UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dept. of Communications

Immersive Teleoperation of a Robotic Arm with Grip Recognition and Haptic Feedback

Master's Thesis

Master of Science in Telecommunication Technologies, Systems and Networks

AUTHOR: Hernandez Gobertti, Fernando Agustin

Tutor: Gómez Barquero, David

External cotutor: Trelis Molina, Pablo

Experimental director: Lozano Teruel, Raúl

ACADEMIC YEAR: 2023/2024

Objectives – The primary objective is to create and evaluate a 5G haptic teleoperation system that allows users to manipulate objects with a robotic arm by mimicking their hand movements and recognizing grip gestures, including force applied and fingers involved. Also, the project seeks to test the configured system on the n78/n40 frequency bands for indoor/outdoor measurements using the 5G Private Standalone Network of iTEAM-UPV.

Methodology – The system is designed around two main nodes: (1) Local Control and Orchestration Node: Situated alongside the robotic arm, this node manages communication with the robot controller and gripper, transmitting and receiving real-time data on robot state and user commands. (2) Remote Teleoperation and Haptic Node: Operable at a distance, this node captures user arm movements through a tracking device and translates them into control signals. It also transmits to the haptic gloves to provide the user with touch feedback based on gripper interactions.

The communication between these nodes using a 5G Private Standalone Network is a critical aspect of the project. The effectiveness of combining two communication protocols (TCP and UDP) on different links is evaluated across diverse environmental conditions (indoor, outdoor) and varying distances through six distinct teleoperation positions within UPV Campus. Additionally, three communication flow configurations are investigated to determine the most efficient and reliable data exchange for optimal system performance.

Theoretical developments – The project leverages existing and new theoretical frameworks for robotic control and human-computer interaction. Machine learning techniques, specifically a custom Convolutional Neural Network, are employed to recognize various grip gestures performed by the user, which allows for a more nuanced translation of user intent into control commands for the robotic arm. Also, several mathematical structures are configured, applied and tested in the overall system, including multiple trajectory mappers for the robot arm, force and width mappers for the gripper, and delay models as well as static, dynamic and adaptive models for specifying complex haptic sensations.

Prototype development and laboratory work – A functional prototype of the teleoperation system has been developed utilizing readily available components. The system integrates a UR5e robotic arm, an OnRobot RG2 gripper, bHaptics TactGloves DK1 for haptic feedback, and a camera interface for remote control. Software libraries such as ROS Noetic, mediapipe, and tensorflow have been employed to facilitate communication, movement tracking, and grip gesture recognition.

The laboratory work focused on defining and configuring the entire study system, together with measuring and analyzing the communication aspects of the system. This involved testing the performance of different protocols under various environmental conditions and flow configurations. Latency, jitter, throughput, RSRP/RSRQ and SINR as well as CPU/RAM requirements were measured to assess the impact of communication protocols and physical location on system responsiveness and data transmission efficiency.

Results – The project successfully demonstrates the feasibility of an immersive teleoperation system with grip recognition and haptic feedback. The communication analysis yielded a comprehensive dataset on the system's performance under various conditions.

(A) Impact of Environment: In indoor environments with minimal signal interference, the average end-to-end system latency using TCP was measured to be around 95 ms. This latency increased to an average of 172 ms in outdoor environments with static (e.g., trees, buildings) and dynamic signal obstacles (e.g., vehicles, individuals). UDP, on the other hand, exhibited slightly lower average latency (80 ms indoors and 148 ms outdoors) but with higher jitter values, indicating less predictable data packet arrival times.

(B) Protocol Suitability: TCP proved more reliable for transmitting critical robot control data due to its in-order delivery guarantees. However, for haptic feedback data, UDP offered a viable alternative with acceptable latency, especially considering its lower overhead and potential for faster transmission rates.

(C) Communication Flow Optimization: The analysis revealed that Option 3 (Parallel flow) achieved the most efficient network usage. In this configuration, the tracking control sends simultaneous packets to both robot and gripper controls, and also both the gripper and tracking controls send independent packets to haptic control with necessary data. This approach resulted in less overall network load compared to Option 1 (Sequential flow) and offered better responsiveness albeit worse program efficiency than Option 2 (Cascaded flow) by reducing the number of independent data packets requiring processing.

Furthermore, subjective user evaluations are also presented, assessing factors such as perceived control latency, haptic feedback fidelity, and overall system engagement.

Future lines – (1) Environmental Adaptation: Develop algorithms that can adaptatively combine 5G transmission modes, communication protocols and flow configurations based on real-time environmental conditions to optimize system performance.

(2) Semantic Optimization: Investigate the relationship between immersion models and their impact on contextual QoS/QoE parameters, identifying optimal configurations.

(3) Real-Time Delay Compensation: Define adaptive compensation techniques that can dynamically adjust to varying network conditions and provide a seamless experience.

Publications – No submitted publications until now, but several proposals are being considered for future articles in collaboration with other European universities.

Abstract – This paper presents the development and communication analysis of an immersive teleoperation system for robotic arm control with a haptic wearable glove. The system utilizes a combination of movement tracking, grip recognition, and haptic feedback to provide users with an intuitive and tactile experience when manipulating objects remotely. The effectiveness of different communication protocols and flow configurations is evaluated across various teleoperation positions and environmental conditions using a 5G Private Standalone Network. The results demonstrate the feasibility of the proposed approach and highlight the importance of communication strategies in optimizing system performance and user experience. Future research directions are outlined to further enhance the capabilities and broaden the potential applications of this teleoperation system.

Author: Fernando Agustin Hernandez Gobertti, email: fahergob@iteam.upv.es
Principal Tutor: David Gomez Barquero, email: dagobar@iteam.upv.es
External Co-tutor: Pablo Trelis Molina, email: pablo.trelis@fivecomm.eu
Experimental Director: Raul Lozano Teruel, email: raulote@iteam.upv.es
Submission date: 03-09-24

LIST OF ACRONYMS –

- 5G NSA/SA: 5G Non-Standalone / Standalone
- AI: Artificial Intelligence (Emerging Concept)
- AR/VR: Augmented Reality / Virtual Reality
- BCI: Brain-Computer Interface (Emerging Concept)
- BR/XR: Blended Reality / Extended Reality
- CAPIF: Common Application Common Interface Framework
- CNN: Convolutional Neural Network (Deep Learning)
- EPON: Ethernet Passive Optical Network
- DT: Digital Twin (IMT-2020, IMT-2030)
- GUI: Graphic User Interface (Software)
- HMI: Human-Machine Interface (Emerging Concept)
- HSV: Hue Saturation Value (Color Theory)
- IC: Immersive Communications (IMT-2020, IMT-2030)
- IMM-Lab: Immersive Communications Laboratory
- iTEAM: Institute of Telecommunications and Multimedia Applications
- LCON: Local Control and Orchestration Node / Local Node
- LRA: Linear Resonant Actuator (Haptic Vibrator)
- LTE: Long-Term Evolution (4G, IMT-2010)
- MCG: Mobile Communications Group
- MIMO: Multiple Input Multiple Output
- MOS: Mean Opinion Score (Subjective Metric)
- NaC: Network as Code (IMT-2020)
- PX: Position X (Measurements)
- QoS/QoE: Quality of Service / Quality of Experience
- RSRP: Reference Signal Received Power

- RSRQ: Reference Signal Received Quality
- RTHN: Remote Teleoperation and Haptic Node / Remote Node
- SDR: Software-Defined Radio (Emerging Concept)
- SMA: SubMiniature Antenna (Routing Device)
- SINR: Signal-to-Interference + Noise Ratio
- UDP: User Datagram Protocol (OSI 4th Layer)
- UE: User Equipment (Mobile Device)
- UML: Unified Modelling Language (Programming)
- UPV: Polytechnic University of Valencia (ES)
- UR: Universal Robots (European Company)
- TCP: Transmission Control Protocol (OSI 4th Layer)
- TI: Tactile Internet (Emerging Concept)
- XML-RPC: Extensible Markup Language Remote Procedure Call

LIST OF FIGURES –

Contents

1 Introduction

The burgeoning field of human-robot interaction is constantly seeking novel methods for intuitive and efficient control of robotic manipulators. Teleoperation systems [49, 50], which allow remote control and skill transfer capabilities of robots, play a crucial role in this domain. However, traditional teleoperation methods [26, 41] often lack the dexterity and sensory feedback necessary for nuanced manipulation tasks.

1.1 Motivation and Applicability

This project is driven by the need for more immersive and user-friendly teleoperation systems. By incorporating features like grip recognition and haptic feedback, the proposed system aims to bridge the gap between human intent and robotic action.

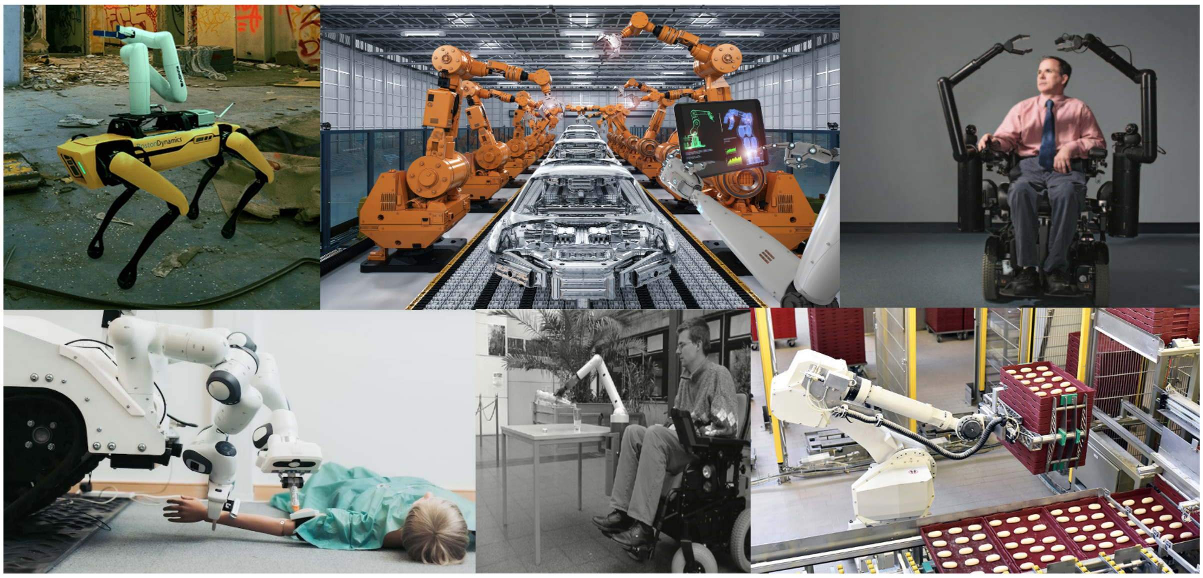


Figure 1: Real-world applications that would benefit from haptic teleoperation systems.

This can be particularly beneficial in scenarios where:

- **Remote manipulation in hazardous environments**: Robots can be employed for delicate tasks in hazardous or hard-to-reach areas, minimizing human risk [64]. For example, transport or inspection robots such as Automatic Guided Vehicles or Quadrupeds equipped with the teleoperation system could be deployed in disaster zones to search for survivors in collapsed buildings or perform other tasks in dangerous environments [24, 67].

- **Precision handling in industrial settings**: The system can facilitate intricate assembly or manipulation tasks in manufacturing processes, improving efficiency and accuracy [25]. For example, the teleoperation system could be employed for remote inspection and handling of delicate products on a production line, ensuring quality control without compromising product integrity [5, 65].

- **Assistive applications**: The teleoperation system can be tailored to assist individuals with limited mobility, empowering them to perform daily tasks with greater autonomy [42]. For example, the teleoperation system could be used to control robotic arms mounted on prosthetic limbs or wheelchairs, enabling users to perform tasks like grasping objects or reaching for items placed out of reach [13, 14].

1.2 Historical Background

The concept of teleoperation has a rich history [6], with early examples dating back to the 19th century. The development of advanced communication technologies and robotics has significantly accelerated the field in recent decades.

- **1870s**: Pioneering inventors like John Louis Lay, John Ericsson, and Victor von Scheliha developed prototypes for remotely controlled torpedoes, laying the groundwork for teleoperated devices.

- **1898**: Nikola Tesla publicly demonstrated a radio-controlled boat, showcasing the potential for wireless teleoperation. While not commercially successful at the time, it foreshadowed the future of wireless robot control.

- **1940s-1950s**: World War II spurred advancements in remote control technologies, with the development of bomb disposal robots and early unmanned aerial vehicles.

- **1950s-1960s**: Raymond Goertz's invention of the first master-slave manipulator with force feedback revolutionized teleoperation. This system allowed operators to handle radioactive materials from a safe distance, demonstrating the practical applications of teleoperation in hazardous environments.

- **1960s-1970s**: The development of computer control systems and advancements in robotics led to more sophisticated telemanipulation systems. The first lunar rovers, like Lunokhod 1 in 1970, were controlled remotely from Earth, showcasing the potential of teleoperation in space exploration.

- **1980s-1990s**: Advancements in microprocessors, communication technologies, and sensor development have fueled the miniaturization and increasing sophistication of teleoperation systems.

- **2000s-2010s**: Evolution in digital mobile communication networks enabled initial long distance remote operations with basic limited grounded haptic feedback.

- **2020s**: The widespread adoption of robots in various industries, coupled with research and innovation in wearable haptics and machine learning, continues to push the boundaries of teleoperation capabilities.

These historical developments [45, 44] highlight the ongoing evolution of teleoperation technology. The proposed system, with its integration of grip recognition and haptic feedback, builds upon this rich legacy and aims to further enhance the user experience and expand the potential applications of teleoperated robotic manipulation.

1.3 Scope and Objectives

This project focuses on the development and evaluation of an immersive teleoperation system for a robotic arm. The system leverages three key aspects:

- **Hand movement tracking**: Capturing the user's hand movements through specialized tracking devices, in particular 3D cameras.
- **Movement-Action correlation**: Mapping user's movements and intent to robot trajectories and actions, using diverse models for an augmented user experience.
- **Grip recognition**: Employing machine learning techniques (specifically, a custom Convolutional Neural Network) to identify different grip configurations performed by user identified by force applied (i.e. Hard, Soft) and fingers used (i.e. 2F to 5F).
- **Haptic feedback**: Providing the user with tactile sensations that simulate the interaction between the gripper and manipulated objects, using haptic gloves.

The primary objective is to create an intuitive and user-friendly system that allows for natural and precise remote manipulation of objects. This is achieved by:

- **Developing a robust communication architecture**: The system utilizes a two-node architecture for efficient data transmission between the local control unit (co-located with the robot) and the remote teleoperation unit.
- **Evaluating the effectiveness of different communication protocols**: The project assesses the performance of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) under various environmental conditions (indoor, outdoor, varying distances) to determine the optimal protocol for reliable and efficient data exchange.
- **Analyzing the impact of communication flow configurations**: Three potential communication flow configurations are investigated to identify the most efficient approach for data routing between the tracking, robot control, gripper control, and haptic control programs.

In addition to the core objective of user-friendly manipulation, the project also has secondary objectives focused on measurement and communication analysis:

- **Measuring system latency and responsiveness**: The project quantifies delays in data transfer across different communication protocols, configurations, and environmental conditions. This is crucial in assessing the system's overall responsiveness and suitability for real-time manipulation tasks.
- **Evaluating network performance**: Throughput (data transfer rate) and Signal-to-Interference + Noise Ratio (SINR) are measured to assess the impact of communication protocols, configurations, and environmental factors on network efficiency.

By achieving these objectives, the project aims to not only create an immersive teleoperation system but also gain valuable insights into the influence of communication strategies on system performance. This data is instrumental in refining the system for real-world deployments, ensuring optimal user experience and reliable remote manipulation capabilities.

1.4 Document Structure

This document is structured to provide a comprehensive understanding of the immersive teleoperation system and its communication analysis. Firstly, Section 2 establishes the context by introducing the project's research environment and the core functionalities of human-to-machine interaction . It then delves into the criteria for immersion quality, emphasizing the role of haptic communication and the specific requirements of the targeted use case. Secondly, Section 3 delves into the technical details of the system, outlining its architecture, describing the utilized devices, and exploring its extended functionalities, such as multiple grip recognition and customization options.

Then, Section 4 details the experimental setup, including the measurement positions, communication protocols and configurations investigated, working models employed, and the data mapping processes used for analysis. Subsequently, Section 5 presents a thorough analysis of the system's performance. It dissects the findings based on teleoperation positions (latency, jitter, throughput, and SINR measurements), communication protocol combinations (network conditions and program execution times), and the chosen communication configuration (network performance and program resource requirements). Additionally, it incorporates subjective evaluations to assess the user experience of the immersive teleoperation system.

Finally, the Section 6 summarizes the project's achievements, identifies potential areas for improvement, and outlines promising avenues for future development in the field of immersive teleoperation systems. Also refer to Annex A for an overview on some article proposals rooted on the research described in this document.

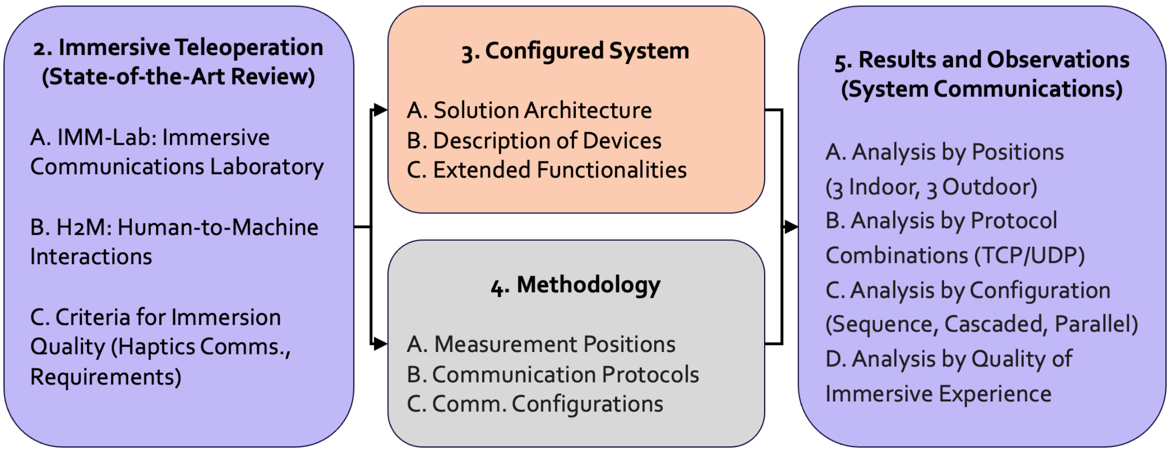


Figure 2: Main sections and subsections outlined in this document.

2 Immersive Teleoperation

This section establishes the foundation for the immersive teleoperation system by introducing the research considerations that facilitated its development and testing, such as the network configuration that was used in the project as well as relevant aspects of human-to-machine interactions and criteria for immersion quality.

2.1 Immersive Communications Laboratory

The IMM-Lab [53] serves as a pivotal platform for research and development efforts focused on evaluating and shaping 5G/6G networks for Immersive Communication (IC) applications. The following subsections delve into the details of this architecture, exploring the specific functionalities of each site and how they contribute to the research conducted at iTEAM.

2.1.1 5G-Advanced Private Network

The 5G-Advanced Private Network managed by iTEAM-UPV [16] serves as the critical foundation for a recently created Testbed on Immersive Communications [52], providing the essential infrastructure for researchers to evaluate the performance of next-generation wireless networks in supporting immersive technologies.

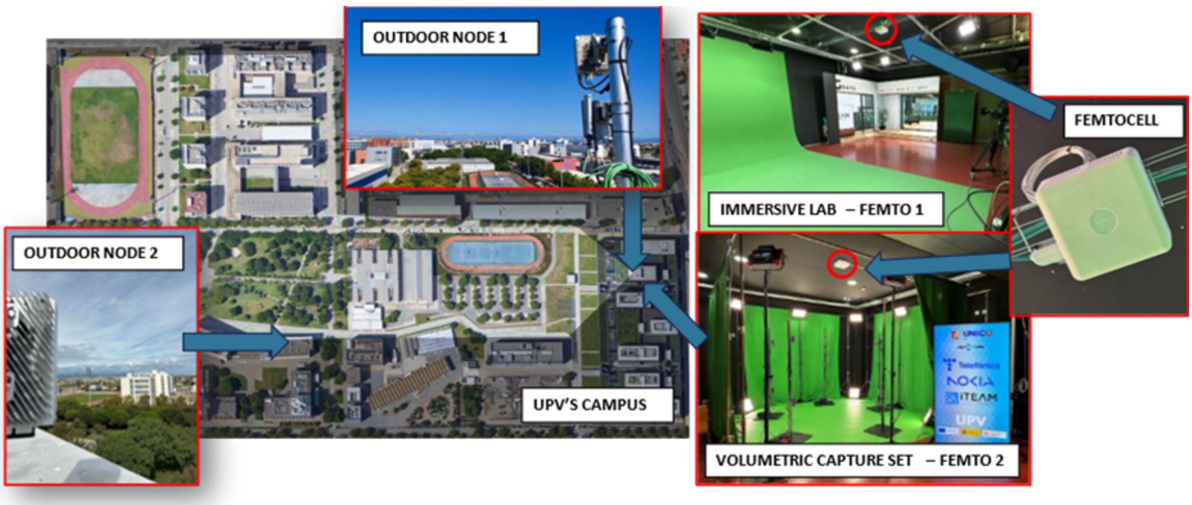


Figure 3: General overview of the distribution of outdoor nodes and indoor femtocells present on the Building 8G of the Vera Campus of UPV.

This network boasts a versatile architecture comprised of three distinct sites, each serving a specific purpose and leveraging different functionalities. This three-site architecture offers UPV researchers a unique advantage. The outdoor nodes provide a realistic testbed environment that reflects real-world network deployments [46], while the dedicated indoor node with its higher throughput and focused coverage allows for in-depth exploration of advanced 5G functionalities [15] specifically tailored for immersive applications. This combination fosters a comprehensive research platform for evaluating the capabilities of 5G networks in supporting the next generation of immersive experiences.

- **Outdoor Nodes, n40 and n258 Bands**:

  The network utilizes two outdoor nodes strategically positioned to provide network coverage across a designated area. These nodes operate in the Non-Standalone (NSA) mode [12], anchoring to an existing LTE network (provided by Telefonica) in the n258 band (estimated bandwidth: 800 MHz) [62]. This configuration allows the network to leverage the established LTE infrastructure for core network functions while utilizing the 5G air interface in the n40 band (owned by UPV, estimated bandwidth: 20 MHz) [23] for enhanced capacity and data transfer speeds.

  These outdoor nodes primarily serve as the backbone for broader network connectivity, offering wider coverage for user equipment (UE) like smartphones or laptops equipped with 5G capabilities [17]. They facilitate initial network access and data exchange between UEs and the core network through existing LTE infrastructure.

- **Indoor Node, n78 Band with Pico-Antennas**:

  Located within the IMM-Lab premises, this dedicated node operates in Standalone (SA) mode [27], offering researchers a high degree of control and flexibility to experiment with core network functionalities without relying on external infrastructure. The node utilizes the n78 band (estimated bandwidth: 100 MHz) [1] specifically allocated to UPV. Additionally, it is equipped with two strategically placed pico-antennas (provided by Nokia) [18] to provide focused and optimized network coverage within the confines of the laboratory.

  This indoor node serves as a dedicated platform for researchers to test and evaluate the performance of 5G core network functionalities in a controlled environment. The SA mode allows them to experiment with network slicing, edge computing, and other advanced features crucial for supporting immersive applications. The pico-antennas ensure high signal strength and minimal interference within the laboratory, creating an ideal testing ground for various immersive technologies.

2.1.2 Testbed on Emerging Immersive Applications

The ICL's Testbed adopts a multifaceted architecture designed to meticulously simulate real-world scenarios relevant to 5G-powered immersive communications. This comprehensive approach allows researchers to evaluate network performance, user experience, and the effectiveness of various technologies in supporting seamless interaction within Blended Reality (BR) [11] environments.

- **Physical Infrastructure**

  The testbed leverages a high-performance core network, responsible for routing and managing data traffic across the entire system. This core network connects to an

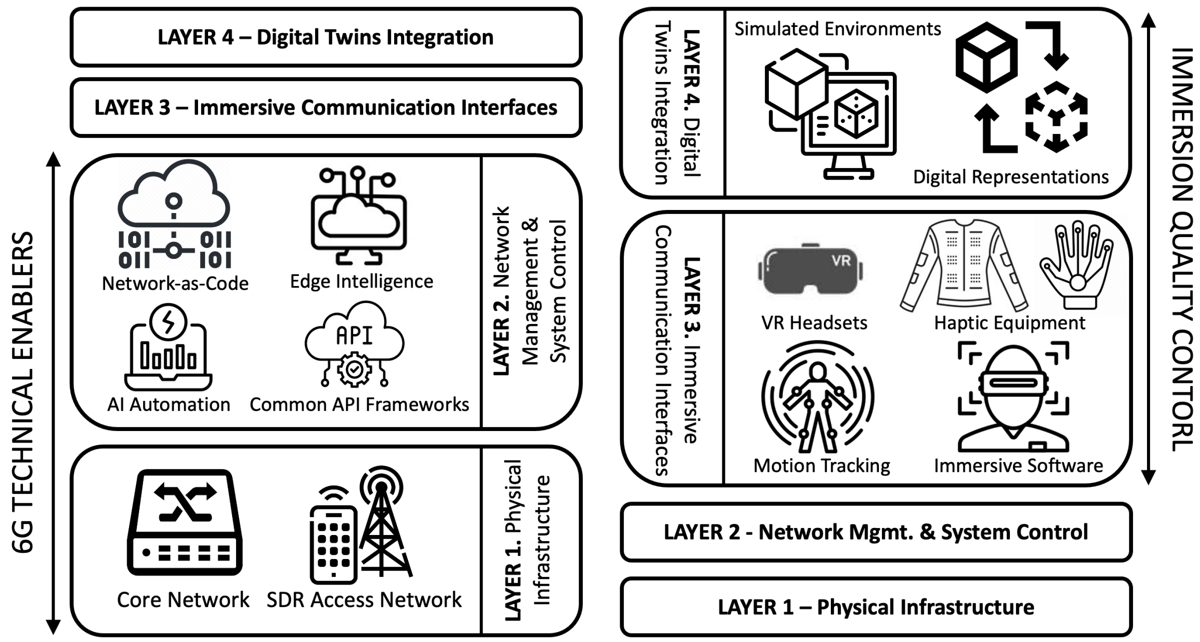


Figure 4: Layered Architecture defined for the Immersive Communications Laboratory Testbed located at the Building 8G of the Vera Campus of UPV.

access network built upon Software-Defined Radio (SDR) technology [8]. SDRs offer unparalleled flexibility, acting as programmable radio transceivers that can be dynamically configured to emulate diverse network conditions. This enables researchers to simulate various network topologies, such as cellular networks, mesh networks, or even satellite communication scenarios. Additionally, SDRs allow for experimentation with different spectrum allocation strategies, optimizing network resource utilization and exploring novel spectrum sharing techniques tailored for the specific demands of IC applications.

- **Network Management and System Control**

  The testbed incorporates advanced network slicing functionalities. Network slicing [37] allows for the logical division of the physical network into multiple virtual network segments. Each slice can be independently configured with dedicated resources (bandwidth, processing power) to cater to the specific requirements of different applications. In the context of ICs, researchers can create dedicated network slices with ultra-low latency and high bandwidth to ensure seamless user experience within VR and AR environments. Furthermore, the testbed integrates edge computing capabilities. Edge computing [35] distributes processing power and storage resources closer to the network edge, where data is generated and consumed. This minimizes latency by reducing the need for data to travel long distances to centralized servers for processing. For latency-sensitive IC applications, real-time processing of data at the network edge is crucial for maintaining a high degree of responsiveness within BR environments.

- **Immersive Communication Interfaces**

  The testbed is equipped with a diverse range of cutting-edge IC interfaces. This allows researchers to create a realistic and comprehensive simulation of user interaction within BR scenarios. Key interfaces utilized within the testbed are VR/AR Headsets [70], Haptic Feedback Devices [55] and Motion Tracking Systems [51].

- **Digital Twin Integration**

  The testbed integrates in this layer the concept of Digital Twins (DT) [19], i.e. real-time virtual replicas of physical entities or processes. Within the context of the IMM-Lab Testbed, researchers can create DTs of the physical testbed environment, including elements such as user locations such as connected courts for remote driving tests, equipment configurations for robotics simulations, and network traffic patterns. This digital representation allows researchers to manipulate various aspects of the simulated environment (e.g., increase user density, introduce network congestion) to assess the performance of the network under diverse conditions relevant to specific immersive scenarios. By leveraging DTs, researchers can efficiently explore a broad range of possibilities without the need for constant reconfiguration of the physical testbed environment.

2.2 Human-to-Machine Interactions

The efficacy of a teleoperation system hinges on the nature of its human-to-machine interactions [69]. This section delves into the core functionalities that underpin the system's operation. By understanding the intricacies of remote manipulation and the potential of learning from demonstration, we establish a foundation for evaluating the system's performance and identifying avenues for improvement.

2.2.1 Remote Manipulation

Remote manipulation technology has witnessed significant advancements in recent years, driven by the convergence of robotics [60], artificial intelligence [56], and advanced communication protocols [66].

Some of the most innovative trends pushing the boundaries of this field:

- **Haptic Feedback and Sensory Richness**: Integrating haptic feedback into teleoperation systems is a growing trend, allowing operators to perceive grasping forces and textures of manipulated objects. Technologies like haptic gloves with microfluidic channels provide increasingly realistic tactile sensations, enhancing dexterity and control [10].

- **Brain-Computer Interfaces (BCIs)**: Emerging research explores the potential of BCIs for remote manipulation. By decoding brain signals, BCIs could enable more intuitive control, bypassing traditional joystick or motion tracking interfaces. However, challenges remain in achieving robust and reliable decoding of complex manipulation tasks [43].

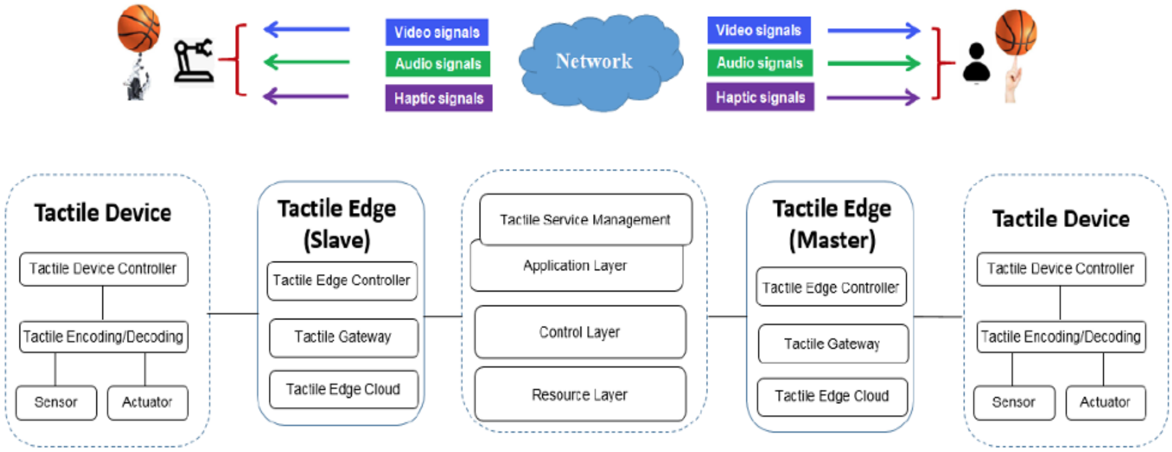


Figure 5: General diagram of Immersive Teleoperation applications. The current work focuses on the management of complex tactile sensations for the Remote Operator.

- **Vision-Based Control and Shared Autonomy**: Vision-based systems that leverage cameras mounted on the robot arm or manipulator are gaining traction. These systems enable real-time visual feedback and object recognition, allowing operators to manipulate objects based on visual cues. Additionally, shared autonomy approaches combine human decision-making with the robot's capabilities, enabling collaborative manipulation tasks [38].

- **Dexterous Robotic Hands and Grippers**: The development of more dexterous robotic hands with multiple articulated fingers is crucial for advanced manipulation tasks. These hands, inspired by the human hand's capabilities, are being equipped with advanced sensors and microfluidic channels to enable grasping and manipulating objects of varying shapes and textures [31].

- **Cloud Robotics and Tactile Internet**: Cloud robotics leverages cloud computing for processing and control tasks, enabling remote manipulation with minimal on-site computational resources. The emerging concept of Tactile Internet, characterized by ultra-low latency and high bandwidth communication, holds promise for revolutionizing remote manipulation by ensuring seamless and reliable data transfer [29].

By integrating these advancements, teleoperation systems are poised to become more intuitive, dexterous, and adaptable, enabling a wider range of applications in various industries, from minimally invasive surgery to hazardous environment manipulation.

2.2.2 Requirements for Teleoperated Applications

The successful implementation of a teleoperation system necessitates a deep understanding of the specific demands imposed by diverse application domains.

Teleoperation systems designed for hazardous environments must prioritize safety, robustness, and reliability. Some key requirements [32, 20] encompass:

- **Latency and jitter tolerance**: Stringent limitations on communication delays and variations to ensure timely operator feedback and prevent accidents. Typical latency thresholds range from 100ms to 200ms, with jitter values below 10ms.
- **Environmental adaptability**: Resilience to harsh environmental conditions such as extreme temperatures, electromagnetic interference, and physical shocks to maintain system functionality.
- **Autonomous capabilities**: Integration of autonomous behaviors for tasks like obstacle avoidance and path planning to reduce operator workload and enhance safety.
- **Telepresence and situation awareness**: Provision of rich sensory feedback, including high-resolution visual and auditory information, to enhance the operator's perception of the remote environment.

Teleoperation systems for industrial applications demand high levels of accuracy, repeatability, and efficiency. Some critical requirements [2, 34] include:

- **Force feedback**: Accurate transmission of forces exerted by the robot's end-effector to provide the operator with tactile feedback for precise manipulation. Force feedback bandwidths typically range from 20 Hz to 1 kHz with force resolution of 0.1 N or better.
- **Position and orientation accuracy**: Precise control over the robot's position and orientation to ensure accurate task execution. Positional accuracy within the range of 0.1-1 mm and orientation accuracy within 0.1-1 degrees are common targets.
- **Human-robot collaboration**: Seamless integration with other automation systems and human workers to optimize workflow and productivity.
- **Human-in-the-loop optimization**: Adaptive systems that can learn from human operators to improve performance over time.

Teleoperation systems designed for assistive purposes must prioritize user-friendliness, intuitiveness, and adaptability. Some task-specific requirements [7, 59] consist of:

- **Intuitive user interface**: Simple and easy-to-use control interfaces tailored to the user's needs and abilities.
- **Adaptability**: Customizable systems that can be adapted to individual user preferences and physical limitations.
- **Safety and reliability**: Robust systems with fail-safe mechanisms to prevent accidents and injuries.
- **Ergonomic design**: Consideration of user comfort and fatigue through ergonomic design of control interfaces and physical devices.

2.3 Criteria for Immersion Quality

Achieving a truly immersive teleoperation experience necessitates a comprehensive evaluation of the factors contributing to the user's sense of presence and engagement [3]. This section examines the critical role of haptic communication in creating a realistic and immersive environment. By defining the specific requirements for different use cases, we establish a framework for assessing the system's ability to deliver a compelling and effective user experience.

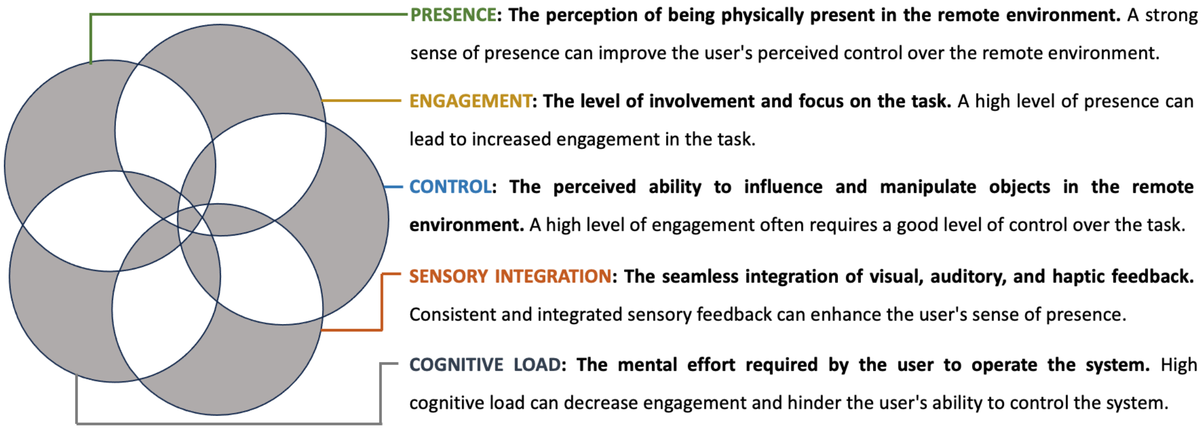


Figure 6: Conceptual Framework that outlines the primary dimensions contributing to Immersion Quality and Quality of Experience, such as feelings of Presence, Engagement, Control, Sensory Integration and Cognitive Load.

2.3.1 Haptic Communications

Haptic communication [4, 36], a rapidly developing field, focuses on transmitting and recreating touch sensations through technological interfaces. This technology plays a crucial role in achieving immersive teleoperation by providing users with a sense of touch that complements visual and auditory feedback.

Some of the explored trends pushing the boundaries of haptic communication are:

- **Rich Tactile Information Transmission**: Beyond simple vibration, advanced haptic devices are incorporating multiple actuation mechanisms to deliver a wider range of tactile sensations. Techniques like microfluidics, shape memory alloys, and ultrasonic actuation enable the creation of pressure variations, texture simulations, and thermal cues, offering a more nuanced and realistic touch experience [30].

- **Biomimetic Haptic Interfaces**: Drawing inspiration from the human somatosensory system, researchers are developing biomimetic interfaces that mimic the intricate structure and functionality of human skin. These interfaces employ arrays of microfluidic channels or pressure sensors to capture and replicate the subtle textures and pressure variations encountered during touch interactions [68].

- **Exploiting Multimodal Feedback**: Integrating haptic feedback with other sensory modalities, such as visual and auditory cues, is a growing trend. By synchronizing haptic feedback with visual representations of grasped objects or the sounds produced during manipulation, researchers aim to create a more cohesive and immersive sensory experience [33].

- **Cloud Haptics and Network Optimization**: The burgeoning concept of cloud haptics leverages cloud computing resources for real-time processing and transmission of haptic data. This approach can overcome limitations of on-site computational power and enable high-fidelity haptic experiences even in resource-constrained environments. However, ensuring ultra-low latency and reliable data transfer through optimized communication protocols remains a challenge [71].

These advancements in haptic communication are fostering the development of more immersive and realistic teleoperation systems. By offering a richer and more nuanced touch experience, haptic technologies are poised to revolutionize various fields, from minimally invasive surgery and remote object manipulation to virtual reality applications and rehabilitation training.

2.3.2 Requirements for Immersive Applications

The successful integration of haptic feedback into a teleoperation system [40] hinges on a deep understanding of the specific use case requirements. By tailoring haptic cues to the unique demands of each application, it is possible to optimize the user experience and enhance task performance.

In hazardous environments, such as nuclear power plants or disaster zones, haptic feedback plays a critical role in enabling safe and effective remote manipulation. Some key requirements [63, 61] include:

- **Force feedback fidelity**: Accurate transmission of forces encountered by the robotic end-effector to the operator's hand is crucial for maintaining control and preventing damage to both the robot and the environment. Force feedback should be capable of reproducing forces within a range of 0-50N with a frequency response of up to 100 Hz.

- **Tactile texture rendering**: Simulating the texture of objects in the environment, such as rough surfaces or slippery materials, can enhance the operator's perception and manipulation capabilities. Haptic devices should be capable of rendering textures with spatial resolutions of at least 100 pixels per inch and a temporal resolution of 100 Hz.

- **Collision detection and impact feedback**: Providing haptic cues to indicate collisions with objects in the environment is essential for preventing damage to the robot and ensuring task completion. Haptic devices should be capable of generating high-frequency (1-2 kHz) impulsive forces to simulate impact sensations.

In industrial settings, haptic feedback can enhance the precision and efficiency of remote manipulation tasks. Some key requirements [28, 39] include:

- **Fine force and torque feedback**: Accurate transmission of forces and torques applied to objects is crucial for tasks requiring delicate manipulation, such as assembly or repair operations. Haptic devices should be capable of reproducing forces and torques with a resolution of 0.1 N and 0.01 Nm, respectively.
- **Kinesthetic feedback**: Providing haptic cues that reflect the object's stiffness and compliance can improve the operator's perception of the object's properties and facilitate precise manipulation. Haptic devices should be capable of rendering stiffness variations within a range of 1-100 N/m.
- **Task-specific haptic cues**: Tailoring haptic feedback to specific industrial tasks, such as tool usage or part insertion, can enhance task performance and reduce operator fatigue. Haptic devices should be capable of generating task-specific haptic patterns, such as vibration patterns for tool activation or force feedback profiles.

3 Configured System

This section delves into the technical architecture and components constituting the proposed teleoperation system. It provides a detailed overview of the system's structure, encompassing both hardware and software elements. Furthermore, it explores the extended functionalities that enhance the system's capabilities beyond basic teleoperation, such as advanced grip recognition and user customization options. An overarching user manual for the installation and utilization of this configured system is detailed in Annex B.

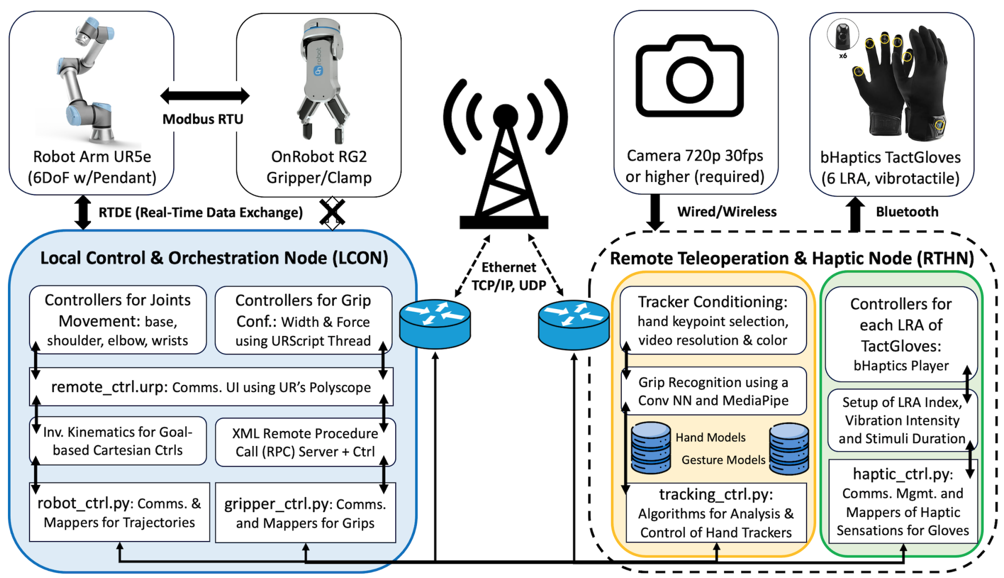


Figure 7: HW/SW components of the proposed immersive teleoperation solution.

3.1 Solution Architecture

The proposed teleoperation system adheres to a distributed architecture comprising two primary nodes: the Local Control and Orchestration Node (LCON) and the Remote Teleoperation and Haptic Node (RTHN). These nodes are interconnected via a robust communication network, facilitating the exchange of essential data for seamless operation.

LCON, situated in close proximity to the robotic arm, serves as the central hub for coordinating robotic actions and environmental interactions. It encapsulates the robotic arm controller, the gripper, and the necessary communication interfaces. This node is responsible for:

- **Low-level robot control**: Executing precise motion commands to the robotic arm and gripper through Universal Robots' Polyscope, multithreaded URScript programs and ROS Noetic packages, based on received instructions.
- **Sensor data acquisition**: Collecting sensory information from the robot's sensors, such as joint positions, end-effector forces, and environmental data using ROS topics provided by Universal Robots and a XML-RPC server for gripper interaction.
- **Communication management**: Handling the exchange of data between the local (*robot_control* and *gripper_control* programs) and remote nodes, ensuring reliable and timely information transfer.
- **Local processing**: Performing real-time computations for tasks such as motion planning, obstacle avoidance, and force feedback generation using the *rospy* library, goal-based cartesian trajectory controllers and a XML-RPC controller.

RTHN, located at the remote user's site, is dedicated to capturing user inputs, processing sensory data, and rendering the haptic feedback of the immersive teleoperation. The key components of this node are:

- **Human-machine interface**: Facilitating user interaction through devices such as a 3D camera as motion capture sensors, bHaptics applications with haptic gloves, and a basic user interface for system status information on visualization displays.
- **Sensory data processing**: Processing data from the human-machine interface using *mediapipe* library for hand detection and gesture models in order to recognize user's intentions, and transforming it into control commands for the robots.
- **Communication management**: Coordinating data exchange (of *tracking_control* and *haptic_control* programs) with the local control node, seeking for low-latency and reliable communications.
- **Haptic feedback generation**: Creating realistic tactile sensations using *tact-python* library based on information received from the remote environment which is translated to actuator selection, vibration intensity and stimuli duration.

The communication network between the two nodes is designed to minimize latency and packet loss, crucial for maintaining a seamless teleoperation experience. Redundancy mechanisms and error correction protocols are implemented to enhance system reliability and fault tolerance. By adopting this distributed architecture, the system seeks to partition functionalities, improve scalability, and enhance system robustness.

3.2 Description of Devices

The system requires a couple of physical devices on LCON and RTHN in order to function adequately. This equipment need not necessarily be the exact same model used in this system, but should work with the same HMIs (e.g. controller topics, messages structure, middle-ware compatibility) as the ones specified below. For a software-specific review of the controllers operation and tracking algorithms, please refer to Annex C.

3.2.1 Local Robot Equipment

LCON needs a robot arm with similar controllers as Universal Robots UR5e and a compatible gripper that can be manipulated through PolyScope's Digital I/O such as OnRobot RG2 with 6-axis force/torque sensor. Using a different e-Series robot arm would require changing the spatial domain limits accordingly, as well as the established safety planes.

	Robot Arm [58] UR5e + Teach Pendant	**Robot Gripper** [54] OnRobot RG2 + Stand
Running Device	Debian OS 8.9 in Atom-E3845 2GB RAM, 4 GB SSD	
Max. Payload / Reach	5 kg / 850 mm	2 kg / 110 mm (max. stroke)
Repeatability	±0.03 mm	±0.02 mm
Degrees of Freedom	6 (base, shoulder, elbow, 3 wrists: yaw, pitch, roll)	2 (fingers)
Power Supply	100-240 V AC	24 V DC
Communication	Ethernet, Modbus TCP, RTDE Fieldbus, EtherNet/IP	Ethernet, Modbus TCP, RTDE, Digital I/O
Weight / Precision	18.4 kg / 5 mm	0.4 kg / 1 mm
Mounting Interface	Standard UR flange	HEX-E QC or other UR-compatible mounting plate
Control Interface	Teach pendant with PolyScope 5.16.0	UR-interface and Digital Outputs
Safety Features	Collision detection, force sensing, safety-rated operation	Force sensing, 3-40 N adjustable grip force
Environmental Rating	IP54 (waterproof, dust protection)	IP40 (basic, not waterproof)

Tabla 1: Technical specifications of LCON equipment: Robot Arm & Robot Gripper.

3.2.2 Remote Equipment

RTHN has been tested with a 720p 30 fps camera for motion tracking which needs to be integrated or connected to the computer node, and vibrotactile gloves that have to interface with bHaptics Player, like bHaptics TactGloves DK1. The camera should have at least the required minimum characteristics for a proper hand detection, including management for adequate brightness and contrast proportions.

	Tracking Device [57] RealSense D415 Camera		**Haptic Gloves** [9] bHaptics TactGlove DK1
Resolution	1920x1080 (RGB), $\leq$ 1280x720 (Depth)	Haptic Actuators	6 LRA motors per glove (Linear Resonant Actuator)
Frame Rate	30 fps (RGB), $\leq$ 90 fps (Depth)	Connectivity	Bluetooth Low Energy (BLE)
Limits (x,y,z)	(X-Y)[0.000, 1.000], Z[-0.050, -0.020]	Vibrotactile Intensity	Range 1-10 m/s^2 Precision 8-12 bits
Depth Accuracy	$\leq$2% at 2 metres	Battery	Lithium-ion rechargeable Bat. Life approx. 3.5 hours
Interface	USB 3.0 Type-C	Charging	2hrs with 5V 0.5A (max) Charge Time approx. 2 hours
Dimensions	99mm x 20mm x 23mm	Compatibility	VR/AR headsets with camera-based hand tracking
Weight	Approximately 30g	Size / Weight	L, XL / Approx 112-115g

Tabla 2: Technical specifications of RTHN equipment: Tracking Device & Haptic Gloves.

3.2.3 Computers Specification

The local and remote equipment communicates with computing processors whose characteristics are specified below in the interest of testings for processing requirements. LCON computer uses Ubuntu 20.04 as it needs to execute ROS, while RTHN computer utilizes Windows 10 as it requires to run bHaptics Player.

	LCON Computer MSI Modern-14 [48] Model C12M-077XES	**RTHN Computer** MSI Cyborg-15 [47] Model A12UCX-657XES
CPU / RAM Available	Intel Core i7-1255U / 16GB (1.70 GHz)	Intel Core i7-12650H / 32GB (2.30 GHz)
Storage / Description	1TB SSD / 14” LLVM 12.0 (256 bits)	512GB SSD / 15.6” RTX 2050 (4GB GDDR6 VRAM)
Operative System	Ubuntu 20.04.6 LTS, 64 bits	Windows 10 Pro 22H2, 64 bits

Tabla 3: Technical specifications of the Computers used in LCON and RTHN.

Additionally, the communication modems used for the testings are a couple of Fivecomm 5G-Broad for SA/NSA with Release-16 and network slicing support [22]. These modems are prototypes built on top of Raspberry Pi 4, and are enabled to function with an inserted SIM card on low and mid frequency bands such as n40 and n78 for outdoor and indoor procedures respectively. Outside the modem box, it contains connectors for 6 SMA external antennas for MIMO as well as Ethernet/USB connectors for external devices.

3.3 Extended Functionalities

In order to tailor the system for more task-specific functionalities, several grip formats and customization options are proposed and implemented. This allows for an more nuanced and smooth experience as well as a more comfortable and easy-to-use testbed for immersive teleoperations. These functionalities have been implemented using the object-oriented code structure with procedural controls specified in Annex D.

3.3.1 Multiple Grip Recognition

Seeking a better experience quality for the user, the system needs to be able to identify the user intentions, i.e. moving the robot, opening/closing the gripper. With that objective, the CNN for gesture recognition, comprised of a feed-forward X-Model structure with 2 hidden layers, is trained to recognize over 90 different grip formats for both strengths (open/soft/medium/hard/close grip) and fingers used (2F/3F/4F/5F).

Figure 8: Examples of hand and gestures detection recognized as grip intentions.

The training is performed by loading many tracking examples of different hand positions, and thus adding new input neurons to the neural network. Then, the testing is done by comparing relative positions of the hand landmarks and comparing it to each possibility of the training set. This entails the definition and optimization of 4 fingers positions and 3 grip strengths for each of the 7 hand position as showed in the picture above, as the CNN is orientation-dependant albeit not size nor location dependent.

Similarly, the hand detection algorithm provided by Google through *mediapipe* library is color-dependant, which requires to modify in real-time the HSV properties of the input image when using the haptic gloves with colors blue, yellow or black. Also, note that the detection algorithm works properly when the fingers are separately distinguished and not overlapped or covered by any obstacle.

3.3.2 Customization Options

Several customized functionalities have been added to both LCON programs (*robot_ctrl* and *gripper_ctrl*) as well as to both RTHN programs (*tracking_ctrl* and *haptic_ctrl*), which can be defined at the nodes initialization step as system arguments through the command line. These options cannot be modified at runtime due to performance and security reasons, given that they change the expected system behaviour and some of them (e.g. communication-related ones) rely on a mutual agreement between the programs.

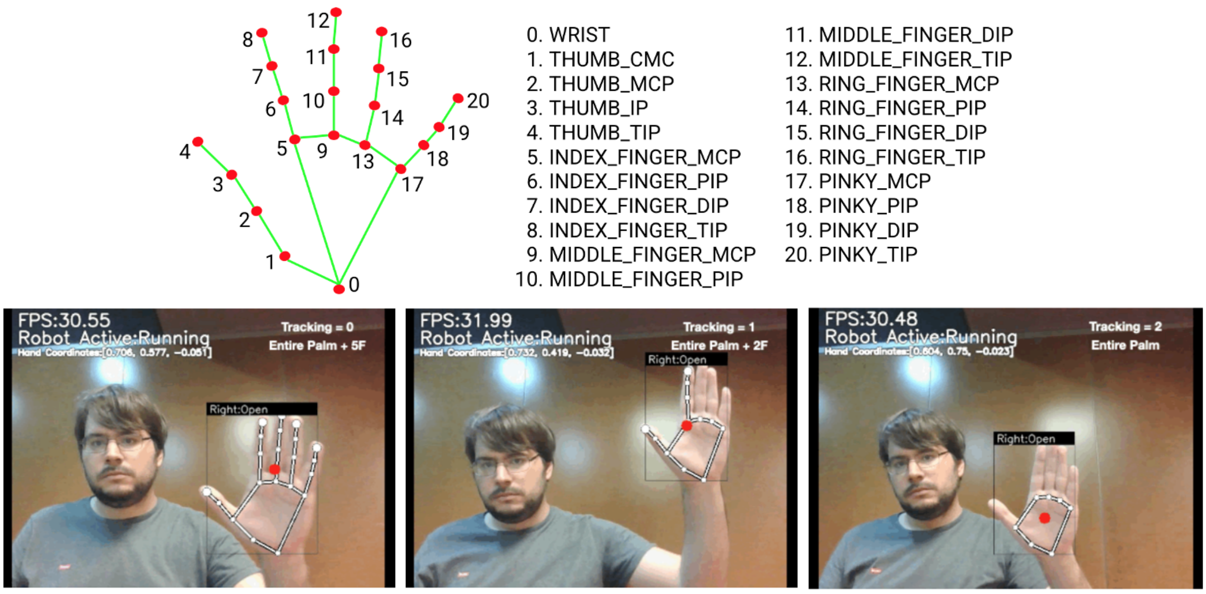


Figure 9: Different landmarks tracking for customized mean point localization.

RTHN's tracking program uses a hand model characterized by 21 landmarks as shown in the above figure, which enables for different configurations to localize a mean point cartesian position to send to LCON node. These positions influence on the precision and quality of experience for gripping teleoperation procedures. Other customization options include the minimum detection and minimum tracking confidence of the gesture model, as well as the selected gloves color, the initial position for starting robot functions, and the switching of the orientation followup procedure (horizontal/vertical).

RTHN's haptic program processes grip output characteristics and sends it to the haptic glove that is being used (left or right), and allows for customization on the intensity range and value shift, maximum fingers allowed, initial and final delay on the vibrations activation, and whether to use several discrete sensations (quantized at 10 ms each) or a unique continuous sensation (of sampled time). In addition to this, the intensity model based on grip width difference and initial force can be selected, which proved to have a great effect on molding the quality experience to each user's sensitiveness.

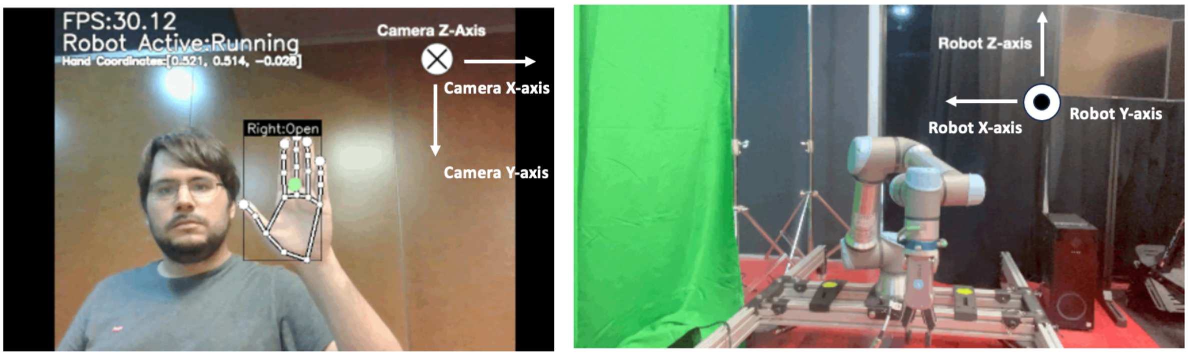


Figure 10: Spatial domains for volumetric Cartesian coordinates in Camera vs Robot.

LCON's robot arm control program allows for the goal-based cartesian trajectory controller specification, as well as the initial robot position and camera-to-robot coordinates mappings in both values and directions, as seen in the above figure. Other customization options include the movement precision, velocity and acceleration indication, which changes the trajectory models used by the robot controllers. Apart from this, the robot arm can be limited for moving in one single direction (X, Y or Z), or in a plane in two dimensions (XY, YZ or XZ), together with specifying whether the movement should mirror or mimic the user tracking result.

LCON's gripper control program provides with options such as maximum width and maximum force indication, their tolerance and gripping models according to tracking strength and sample duration respectively, as explained in Section 4. The program can also limit the maximum amount of considered grip levels, thus optionally filtering tighter closes of the grip in order to avoid increased pressure on the object being manipulated.

Finally, the configured system also allows for a completely local mode, with all tracking management and robot control capabilities enabled while using one single node. This was necessary for cases when both nodes were not available to use simultaneously. However, as the selected haptic drivers are not Unix-compatible, this standalone local mode cannot carry on with tactile functionalities.

For a complete list and description of the customization options, please refer to the *HapticTeleop*'s Github Repository [21], managed by XR iTEAM UPV and iTEAM MCG. Also, refer to Annex E for more detailed information on helper functions used for defining robot trajectories, gripper width and haptic intensity, among others system variables.

4 Methodology

In addition to the system configuration, this project seeks to evaluate and optimize the immersive teleoperation experience by studying different communication procedures and contexts in order further characterize the usage conditions of the prospective service. In this regard, testing aspects such as diverse measurement positions with similar conditions as where the remote manipulation might take place, together with several sets of communications protocols and configurations are implemented and analyzed below.

4.1 Measurement Positions

To systematically assess the teleoperation system's performance under a range of controlled conditions, a comprehensive definition of physical measurement positions was implemented. This involved establishing a series of diverse testing environments encompassing both indoor and outdoor locations. These positions were selected to introduce variations in factors known to influence communication effectiveness, such as signal attenuation, line-of-sight availability, and distance. By meticulously varying these parameters, we aimed to collect data that characterizes the system's robustness and fidelity across a spectrum of potential real-world usage scenarios. The following sections detail the specific indoor and outdoor positions chosen for this evaluation.

4.1.1 Selected Indoor Positions

To comprehensively evaluate the teleoperation system's performance under diverse operational scenarios, a series of controlled indoor testing environments were established.

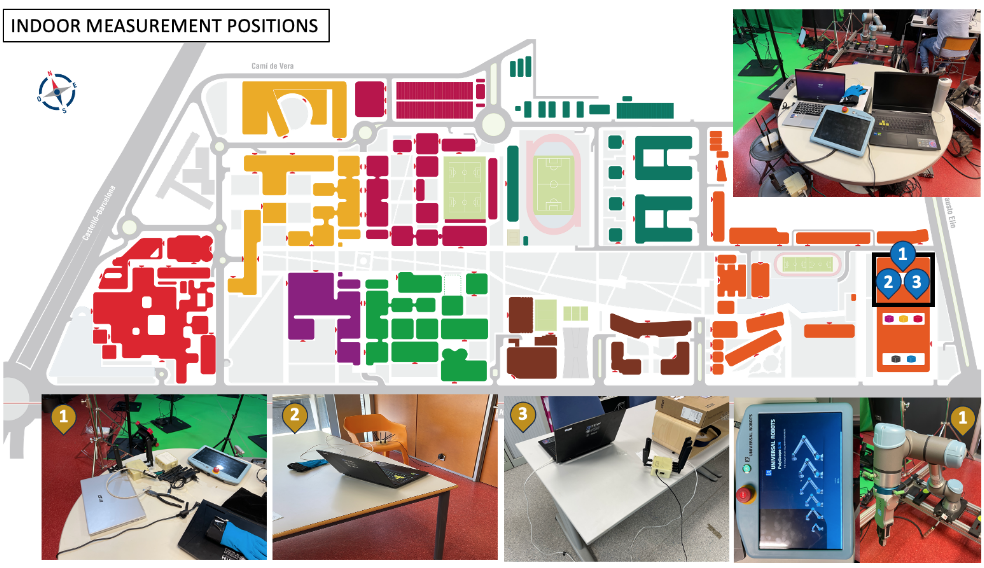


Figure 11: Locations of indoor measurement positions within the UPV Campus, along with pictures of the remote stations where the immersive teleoperation is performed.

These positions aimed to replicate potential real-world usage conditions using the n78 frequency band while systematically varying factors such as:

- **Line-of-Sight**: Position 1 (P1), located within the robot room with no obstacles, represents a best-case scenario with a clear line of sight between operator and robot.
- **Signal Attenuation**: Position 2 (P2) introduces controlled signal attenuation by placing the operator station within the same building but in proximity to the robot room. Walls and other building structures weaken the communication signal, allowing assessment of system robustness under such conditions.
- **Vertical Distance**: Position 3 (P3) investigates the impact of vertical separation by positioning the operator station three floors below the robot room within the same building. This introduces additional signal attenuation and potential multipath propagation effects.

4.1.2 Selected Outdoor Positions

The evaluation extends beyond indoor environments to encompass a set of controlled outdoor testing locations.

Figure 12: Locations of outdoor measurement positions within the UPV Campus, along with pictures of the remote stations where the immersive teleoperation is performed.

These positions were chosen to represent potential real-world scenarios with varying degrees of signal propagation challenges using the n40 frequency band:

- **Line-of-Sight with Minimal Obstructions**: Position 4 (P4) locates the operator station on the terrace of the same floor as the robot room. This scenario offers a relatively clear line of sight with minimal obstructions, allowing assessment of the system's performance in an open-air environment.

- **Long-Range Teleoperation**: Position 5 (P5) delves into a more challenging scenario by placing the operator station in a student dormitory room located approximately 600 meters away from the robot in a different building on campus. This extended distance test the system's capabilities under significant signal attenuation and potential interference from other wireless networks.

- **Moderate Distance with Multiple Obstructions**: Position 6 (P6) situates the operator station outdoors at the sports velodrome, approximately 300 meters away from the robot. This position introduces a moderate distance with the potential for signal attenuation due to intervening foliage or other static and dynamic structures.

4.2 Communication Protocols

Following an initial investigation into communication protocols across various measurement positions, this section delves deeper into optimizing these protocols for real-time teleoperation performance. These protocols are treated as $\{\alpha, \beta, \gamma\}$ combinations where:

- α: Protocol between RTHN's Tracking Control and LCON's Robot Control.

- β: Protocol between RTHN's Tracking Control and LCON's Gripper Control.

- γ: Protocol between LCON's Gripper Control and RTHN's Haptic Control.

The filtered combinations of these protocols, shown in Figure 13, focus on selecting the most effective protocols for each control channel based on factors like real-time requirements, data integrity needs, and the specific control task. This also acknowledges physical limitations and network infrastructure realities that influence communication feasibility.

4.2.1 Filtered Combinations

Building upon the evaluation of various communication protocol combinations over diverse network infrastructures, the subsequent stage involves meticulously filtering out unsuitable configurations. This selection process considers factors such as real-time performance requirements, data integrity demands, and the specific control task at hand. For instance, preliminary testing reveal that UDP is an unacceptable choice for gripper control due to its inherent lack of error correction mechanisms. In such scenarios, the high probability of undetected data loss could lead to a multitude of false-positive gripping commands in contrast to more reliable TCP commands, compromising the safety and precision of manipulation tasks. Conversely, UDP demonstrates suitability for haptic control interactions where the data stream is inherently transient and focused on conveying real-time sensations. In these situations, the potential for occasional data loss is less detrimental as overall haptic experience prioritizes the immediacy of feedback over absolute data fidelity.

This filtering process, guided by the specific demands of each control task, ensures that only the most effective communication protocols are retained for further optimization. By carefully balancing real-time responsiveness with data reliability, we aim to establish

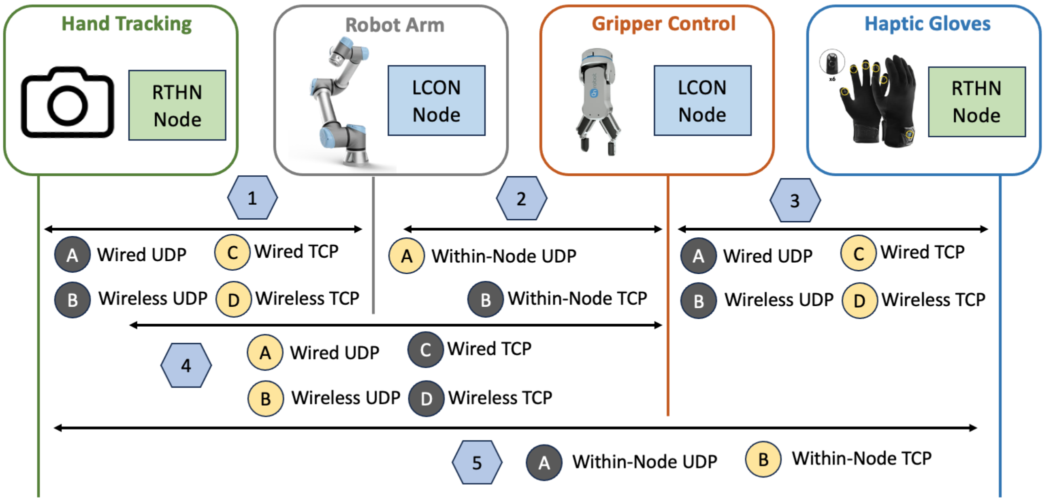


Figure 13: Protocols considered for the interactions between control programs of the study system, emphasizing the filtered combinations in darker colours.

the optimal protocol configuration for each control channel, ultimately contributing to a seamless and immersive teleoperation experience.

Consequently, the selected filtered protocols combinations resulted in:

- Comb. 1 (C1): {TCP, TCP, TCP}
- Comb. 2 (C2): {UDP, TCP, TCP}
- Comb. 3 (C3): {TCP, TCP, UDP}
- Comb. 4 (C4): {UDP, TCP, UDP}

4.2.2 Additional Considerations

While the evaluation primarily focuses on the communication protocols between distinct control programs, it's crucial to acknowledge limitations imposed by the physical infrastructure. Notably, wired EPON connections are only feasible when both communicating nodes reside within the same physical space due to the inherent limitations of cable length. Conversely, intra-node communication, where data exchange occurs entirely within a single physical device, is independent of the communication medium as it doesn't traverse a network. Recognizing these physical constraints alongside the performance characteristics of different communication protocols pave the way for a comprehensive understanding of the system's capabilities and limitations, informing the development of optimization strategies tailored to specific use cases.

4.3 Communication Configurations

This section explores various communication configurations that govern the data flow between the control entities of the study system. These configurations determine the order and dependencies within the communication sequence, ultimately influencing factors like latency, processing overhead, and network traffic. Three primary configurations are investigated: sequential flow, cascaded flow, and parallel flow. Each offers distinct advantages and drawbacks, and the optimal choice hinges on the specific performance requirements and resource constraints of the immersive teleoperation system.

4.3.1 Sequence Flow

This configuration embodies a linear data flow, where the tracking control transmits separate packets directly to both robot arm and gripper controls. Subsequently, the gripper control, upon receiving relevant information, transmits an independent packet to the haptic control unit. This approach offers a relatively straightforward implementation and potentially lower latency for tracking control data. However, it introduces increased network traffic due to the multiple packets and potential delays for haptic control, which relies on feedback from the gripper.

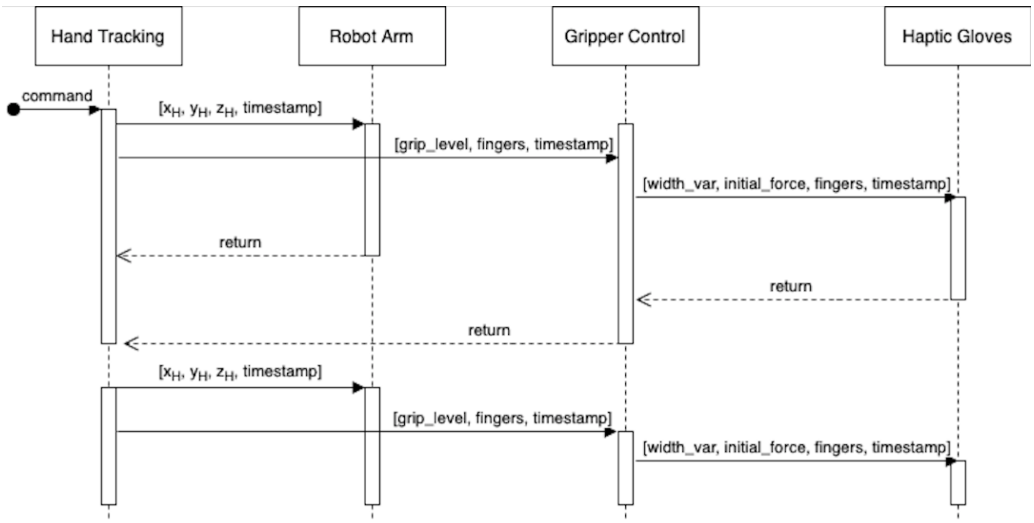


Figure 14: Interaction between control entities for the Sequence Flow Configuration.

4.3.2 Cascaded Flow

In contrast to the linear approach of sequential flow, cascaded flow establishes a chain of communication dependencies. Here, the tracking control transmits a single packet containing control information to the robot arm control unit. The robot control unit then processes this information and relays it, along with additional processing results, to the gripper control unit. Finally, the gripper control unit transmits a separate packet to the haptic control unit, conveying relevant feedback data. This configuration offers the potential benefit of centralized processing and decision-making within the robot control unit. However, it introduces additional processing steps that lead to increased latency for all control entities. Cascaded flow also introduces a single point of failure if the robot control unit malfunctions, and its implementation is fairly more complex compared to sequential flow.

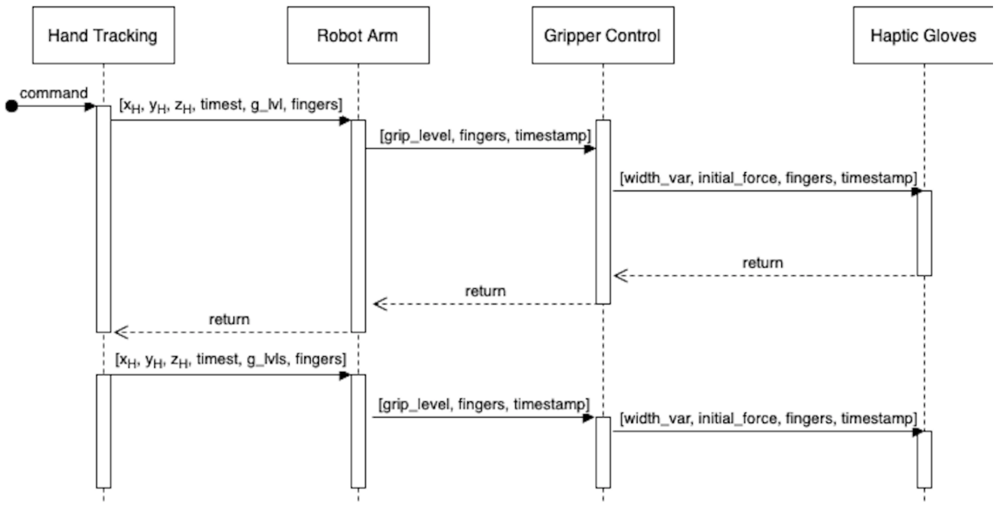


Figure 15: Interaction between control entities for the Cascaded Flow Configuration.

4.3.3 Parallel Flow

Here, the tracking control transmits simultaneous packets to both the robot arm and gripper controls, each containing the necessary information for their respective tasks. Additionally, both the gripper and tracking controls can independently transmit packets to the haptic control unit, including data streams specific to their functions. This approach prioritizes minimizing latency for both tracking and gripper control data. Furthermore, the haptic control unit can receive a richer set of information directly from both sources, leading to a more comprehensive and nuanced haptic feedback experience. However, parallel flow comes at the cost of significantly increased network traffic due to the high number of data packets and potential redundancy within the information streams.

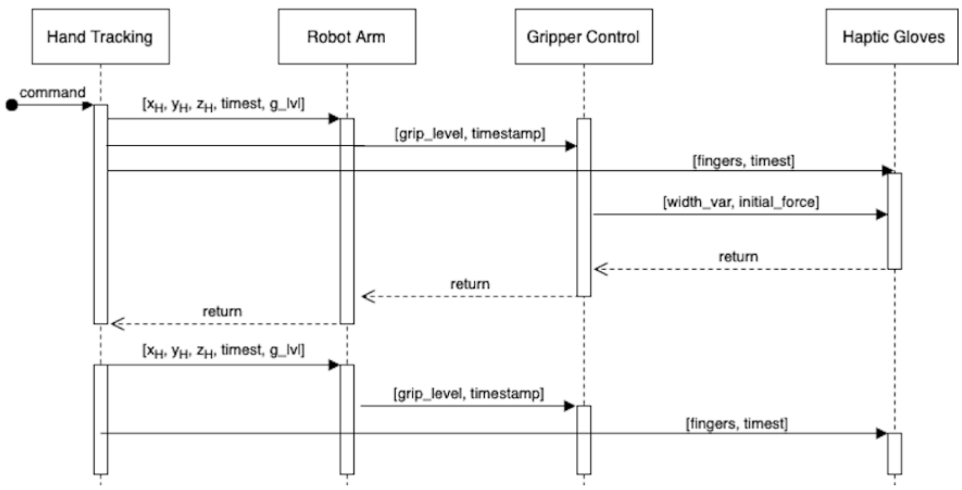


Figure 16: Interaction between control entities for the Parallel Flow Configuration.

Additionally, the haptic control unit faces a greater processing burden as it needs to handle and reconcile multiple data streams simultaneously. Implementing parallel flow also requires careful coordination to avoid conflicting information being received by the haptic control unit, which could negatively impact the overall teleoperation experience.

4.4 Haptic Teleoperation Experience

In addition to the optimization of QoS performance parameters, a reserved survey has been carried out in multiple stages of the project in order to measure immersive QoE aspects. These tests, done by diverse 10 male individuals of 18 to 45 years old, requested the user to successfully grip a small object located within range of the robotic arm, and then rise and lower the arm position with the gripped object.

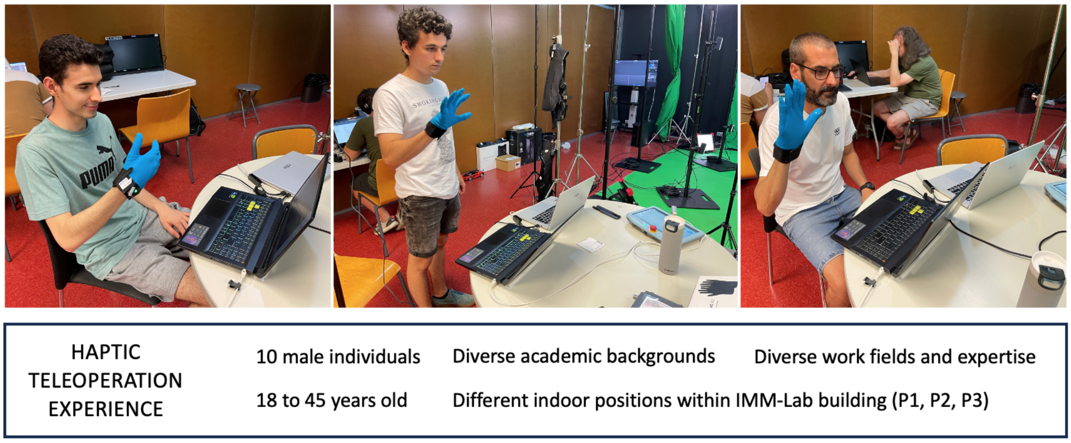


Figure 17: Examples of the Haptic Teleoperation System being tested for User Experience Quality in the Immersive Communications Laboratory of iTEAM-UPV.

The brief survey following the experience focused on quantifying subjective perceptions on MOS scales of 0 to 5 collected by a brief questionnaire completed by the user:

- To what extent did you feel physically immersed in the remote environment?
- How engaged were you in the task at hand during the teleoperation experience?
- How much control did you feel you had over the remote robot during the task?
- How well did the visual and haptic feedback combine to create a cohesive experience?
- How mentally demanding was the task during the teleoperation experience?

Their main objective was to obtain user feedback on immersion quality for feelings of presence, engagement, control, sensory integration and cognitive load across several tries while key system characteristics are being changed. These characteristics involved robot trajectory and grip width mappers as well as haptic intensity models described in Annex E. These experiments also allowed to get outside hands-on commentaries and observations for further improving the system configuration in later stages.

5 Results and Observations

This section delves into the QoS/QoE measurement results from the haptic telemanipulation in various teleoperation positions and using different communication protocols combinations and configuration formats. These measurements have been performed through *AT*, *ping* and *iperf* commands, as well as with in-code registries collected as logs. Clock synchronization between LCON and RTHN is carefully gauged prior to the measurements.

5.1 Analysis by Teleoperation Position

The analysis of system performance across the six designated measurement positions reveals distinct trends in end-to-end latency (from tracking control enabler to haptic control receiver) and signal quality, with significant implications for immersive teleoperation. While indoor environments generally exhibited lower latency and higher signal strength, outdoor conditions introduced challenges due to increased signal attenuation and interference. These findings underscore the importance of robust communication infrastructure in ensuring a seamless and responsive teleoperation experience.

5.1.1 Latency and Measurements Correlation

A comprehensive analysis of latency across the six measurement positions reveals a strong correlation with both distance and environmental conditions. Indoor positions, particularly those within the same building, exhibited average end-to-end latencies ranging from 85 to 125 milliseconds, with minimal fluctuations over time. In contrast, outdoor positions, especially the sports velodrome at 300 meters, demonstrated significantly higher latencies, averaging between 145 and 195 milliseconds, with noticeable spikes during periods of increased environmental interference.

The correlation between latency and signal strength seems evident, with a general trend towards increased latency as signal strength deteriorated. However, the relationship is not strictly linear, as other factors such as multipath propagation, interference, and network congestion influenced the overall latency experience. For instance, while P2 (indoor, nearby robot room) offered generally strong signal levels, the presence of walls and structural elements introduced signal attenuation, resulting in slight latency increases compared to P1 (indoor, within robot room). In contrast, P6 (outdoor, sports velodrome) exhibited both periods of relatively low latency and sudden spikes, likely attributed to fluctuations in signal strength caused by user movement and environmental dynamics.

These aspects are observed in the charts of Figure 18, where Parallel Flow and C4 protocol combination for inter-Node communication has been used. Indeed, the left chart reveals patterns such as consistent low latency in indoor environments and intermittent spikes in outdoor settings, particularly during periods of high user activity or adverse weather conditions. In the case of the right chart, a general trend of increasing latency with decreasing signal strength is detected, although other factors like in-band interference from unrelated applications and multipath propagation from trees and contiguous buildings have influenced the results.

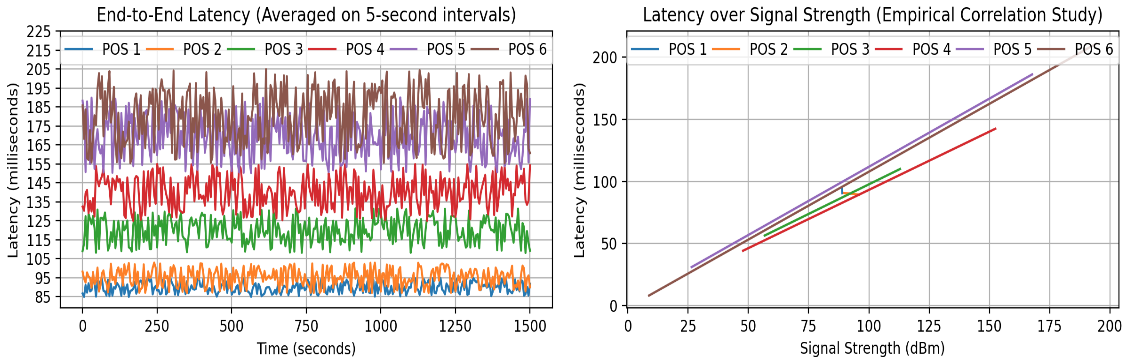


Figure 18: Position-related measurement results for similar gripping operation procedures: End-to-End Latency over Time and Latency over Signal Strength.

5.1.2 Signal Levels and Response Observations

Signal strength measurements, quantified by metrics such as RSRP, RSRQ and SINR, exhibited significant variations across the six measurement positions. Indoor environments generally recorded higher RSRP and RSRQ values, indicating stronger signal reception. Conversely, outdoor environments, especially at the sports velodrome, experienced lower signal levels due to increased path loss and interference.

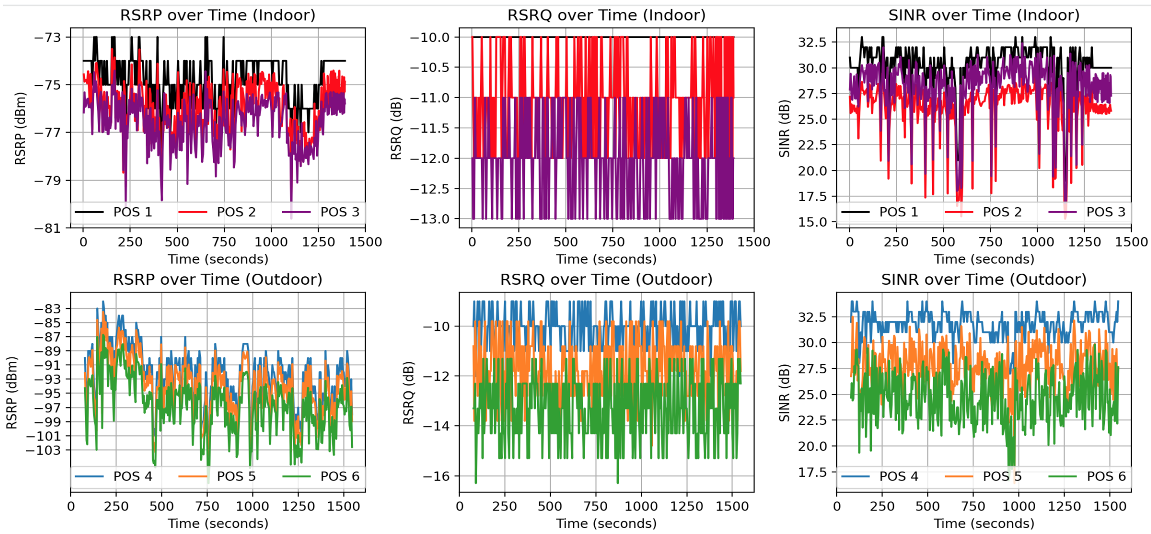


Figure 19: Position-related measurement results: RSRP, RSRQ and SINR over Time.

These considerations can be appreciated in the charts of Figure 19 where RSRP for signal fluctuations, RSRQ for signal quality and SINR for signal interference are analyzed. The left chart shows patterns such as consistent signal strength in indoor environments and intermittent drops in outdoor settings. Then, the middle chart incorporates additional factors like modulation and coding scheme, offering a more comprehensive view of signal conditions. Finally, the right chart indicate higher SINR values for positions (e.g. P2, P3) with better signal quality, while lower values (e.g. P5, P6) suggest increased interference.

5.2 Analysis by Protocol Combinations

The evaluation of various protocol combinations across the control channels yielded distinct performance characteristics with significant implications for system responsiveness, reliability, and overall network utilization. The four configurations tested demonstrated unique strengths and weaknesses in terms of latency, jitter, packet loss, and error rates.

5.2.1 Comparison of Network Performance

A comparative analysis of network performance metrics across the four protocol combinations revealed nuanced trends in latency and jitter. The C1 configuration, while offering robust error correction and reliable data delivery, exhibited higher latency compared to the other combinations, particularly in scenarios with significant network congestion. The introduction of UDP for the tracking control channel in the C2 configuration led to a reduction in latency, but at the cost of increased packet loss rates. Conversely, the C3 combination demonstrated a trade-off between latency and reliability, with improved latency for the tracking and gripper control channels but potential data loss in the haptic control channel. The fully UDP-based C4 configuration achieved the lowest latency but suffered from the highest packet loss rates, necessitating careful consideration of the specific requirements of each control channel.

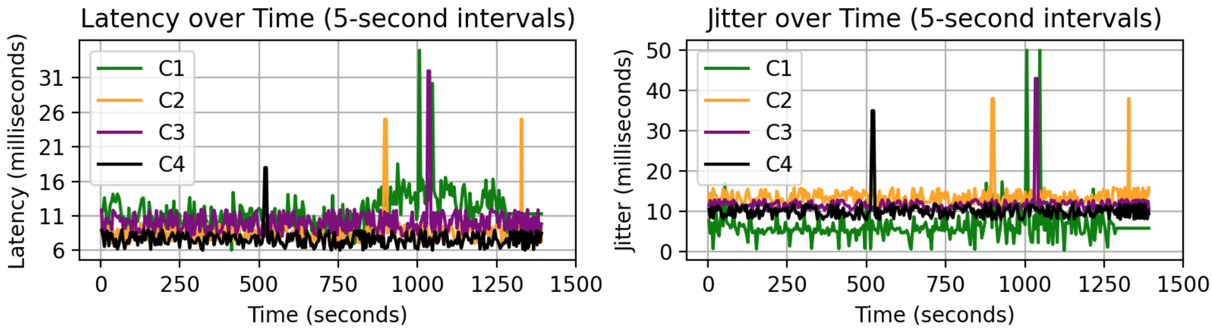


Figure 20: Protocol-related measurement results for similar gripping operation procedures: Network Latency and Jitter over Time.

5.2.2 Packet Loss and Error Rates per Link

An analysis of packet loss and error rates across the different protocol combinations provides crucial information regarding data integrity and reliability. The C1 configuration exhibited the lowest packet loss rates due to its inherent error recovery mechanisms. However, the use of TCP for all channels can introduce additional overhead and latency. The introduction of UDP in the C2 and C3 configurations led to increased packet loss rates, particularly for the UDP-based control channels. The fully UDP-based C4 configuration demonstrated the highest packet loss rates, highlighting the trade-off between low latency and data reliability.

The occurrence of these packet losses and errors generate a dynamic additional delay to the entire system process, which is most noticeable in between consecutive frames of the robot trajectories. This behaviour improves when considering variable-length instead of fixed-length PDUs. Apart from this, note the delay on communications involving the gripper control needs to be added to the frames with the manager of haptic interactions.

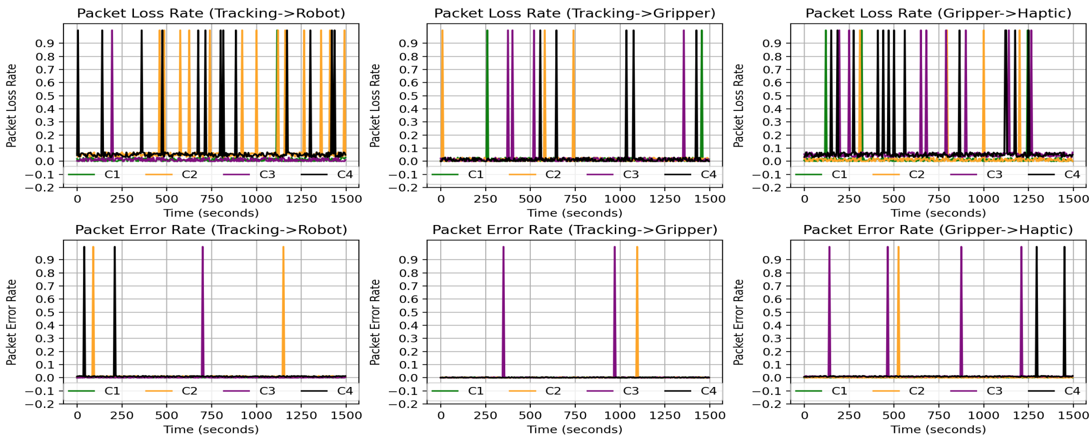


Figure 21: Protocol-related measurement results: Packet Loss Rate and Packet Error Rate over Time for P1 measurement location and Parallel Flow configuration.

5.3 Analysis by Selected Configuration

The evaluation of the three communication configurations (i.e. sequential, cascaded, and parallel) revealed distinct performance characteristics with significant implications for the immersive experience. These configurations exhibited varying degrees of efficiency in terms of control traffic, network utilization, and computational resource demands.

5.3.1 Control Traffic and Network Utilization

A comparative analysis of control traffic and network utilization across the three configurations unveiled notable disparities.

The sequential configuration, characterized by a linear data flow, generally exhibited lower network utilization due to reduced number of data packets exchanged. However, this configuration often resulted in increased latency for the haptic control channel, as it relied on information cascaded from the gripper control.

The cascaded configuration, while introducing additional processing overhead at the robot control node, demonstrated potential benefits in terms of network efficiency by consolidating data transmission. However, the increased latency introduced by the sequential processing steps could impact the overall system responsiveness.

In contrast, the parallel configuration, characterized by simultaneous data transmission, exhibited higher network utilization but offered the potential for lower latency across all control channels. This configuration, however, required careful management to prevent data redundancy and ensure consistent information delivery to the haptic control unit.

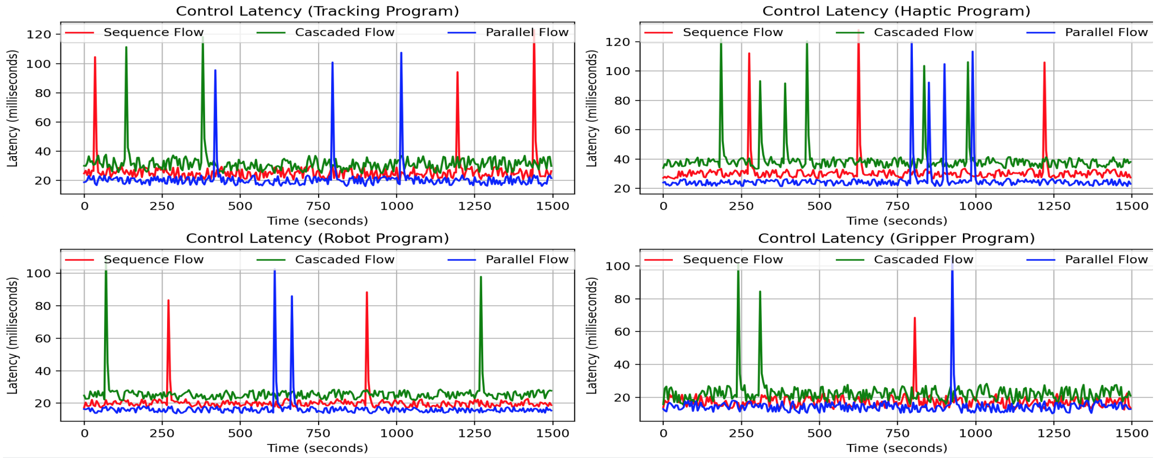


Figure 22: Configuration-related measurement results for similar gripping operation procedures: Latency per Program over Time.

These latency analysis across the three configurations for each control channel (tracking, robot arm, gripper, and haptic gloves) shown in Figure 22 provides valuable insights into the impact of configuration on system responsiveness according to measures carried out in P1 location and with C4 protocol combination for inter-Node communication.

The most relevant perceived bottleneck found in this study is the communication with the Robot Control program, which then causes an overall greater latency for the Cascaded Flow. Both other types of communication flows are not particularly affected by this, as their interactions with the Robot Control program is only circumstantial for modifying the arm trajectory and not a medium for other functionalities.

5.3.2 Program Requirements for each Node

The computational demands placed on the local and remote nodes varied significantly across the three communication configurations. The sequential configuration, with its relatively simple data flow, imposed lower computational requirements on both nodes. The cascaded configuration, particularly at the robot control node, exhibited higher CPU utilization due to the additional processing tasks involved. The parallel configuration, while distributing computational load across multiple nodes, introduced challenges in terms of data synchronization and processing at the haptic control unit.

Regarding RAM demands, the parallel configuration exhibits the highest RAM consumption on both the local and remote nodes, at around 3000MB and 2500MB respectively. This can be attributed to the simultaneous data processing requirements inherent to this configuration. The cascaded configuration demonstrates a moderate increase in RAM usage compared to the sequential configuration, particularly on the local node (2900MB compared to 2650MB in average). This is likely due to additional processing tasks involved in handling cascaded data streams. The sequential configuration exhibits the lowest RAM consumption across both nodes, reflecting reduced processing demands associated with its linear data flow.

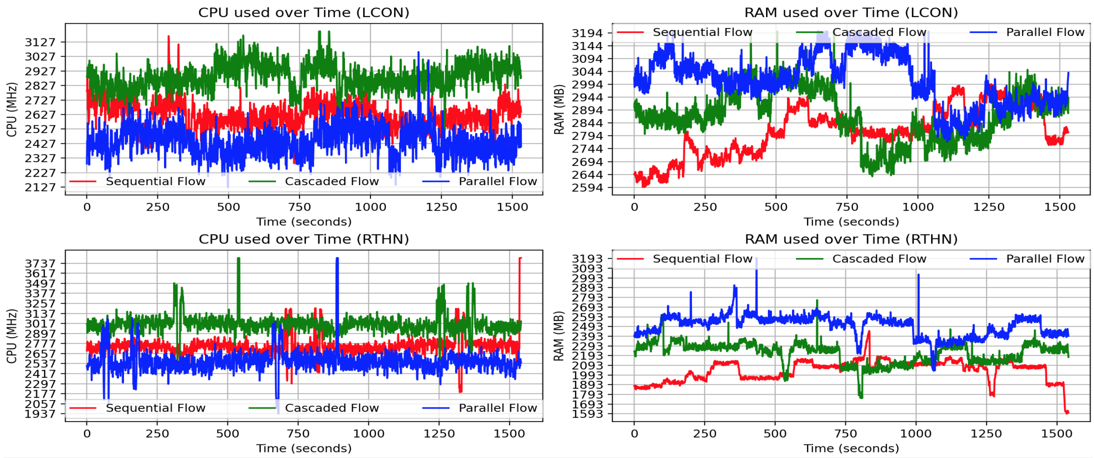


Figure 23: Configuration-based measurements: CPU & RAM usage for LCON & RTHN.

5.4 Empirical Quality of Immersive Experience

A comprehensive evaluation of the immersive teleoperation experience was conducted through a survey involving participants from diverse backgrounds.

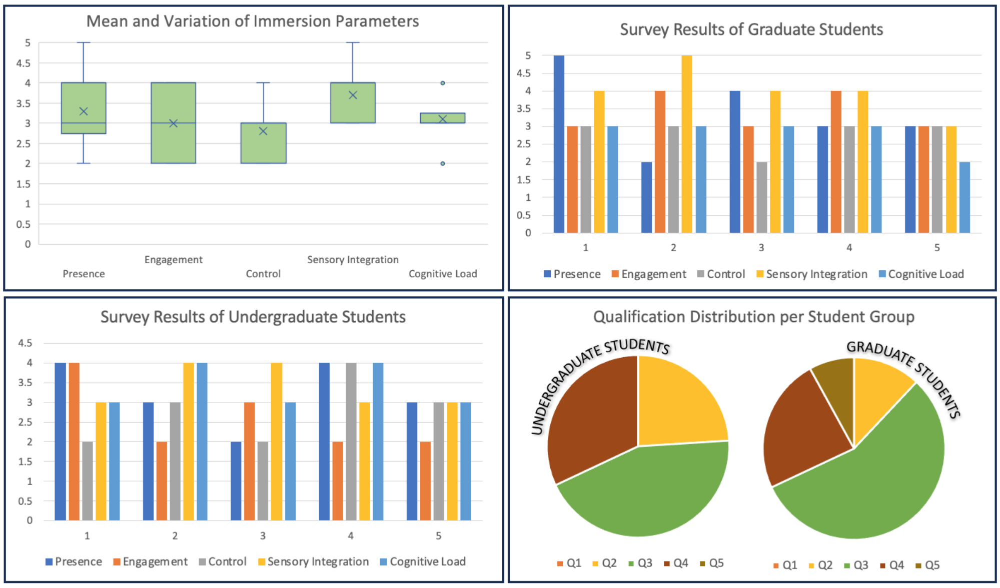


Figure 24: Survey results of the Haptic Teleoperation Experience regarding the 5 parameters or dimensions of the considered Immersion Criteria.

Analysis of the survey data revealed a strong correlation between perceived presence and both haptic feedback quality and system responsiveness. Participants reported a heightened sense of immersion when haptic cues accurately reflected the interaction with the remote environment and system latency was minimal. Conversely, delays in haptic feedback and system sluggishness diminished the feeling of presence, leading to a sense of disconnection from the remote task.

Engagement levels were found to be influenced by task complexity and the level of challenge presented by the teleoperation scenario. Participants reported higher levels of engagement when faced with tasks that required a degree of adaptation. Additionally, the quality of haptic feedback played a role in maintaining engagement, with accurate and informative haptic cues contributing to a sustained focus on the task.

Perceived control was closely linked to system responsiveness and the accuracy of haptic feedback. Participants reported a strong sense of control when the system exhibited low latency and the haptic cues accurately reflected the forces and interactions experienced by the robot through context-adequate haptic intensity models. Conversely, delays in system response and inaccurate haptic feedback diminished the feeling of control, leading to frustration and reduced task performance.

Sensory integration, the seamless integration of visual, auditory, and haptic feedback, emerged as a critical factor in creating a cohesive and immersive experience. Participants reported a more immersive experience when the sensory cues were congruent and complementary, enhancing the overall sense of presence and engagement. Conversely, inconsistencies between sensory modalities led to cognitive overload and diminished the overall quality of the experience.

Cognitive load, as measured by subjective ratings of mental effort, was influenced by factors such as task complexity, system responsiveness, and the quality of haptic feedback. Participants reported higher cognitive load when faced with complex tasks, experiencing system delays, or encountering ambiguous haptic cues. Effective haptic feedback, by providing additional information and reducing reliance on visual and auditory cues, contributed to lower cognitive load and improved overall user experience.

These findings underscore the importance of optimizing haptic feedback, system responsiveness, and task design to create an immersive teleoperation experience intended to maximize user engagement and minimize cognitive load.

6 Conclusions

The preceding analysis delved into the intricate interplay of communication protocols, network infrastructure, and system configuration on the overall performance of the immersive teleoperation system. Next, this section provides with a brief summary and closing arguments to the project related to improvement possibilities and further developments inteded for the configured system.

6.1 General Summary

This project created the baseline configuration of an immersive teleoperation system that functions according to requirements, with some movements lagging still being analyzed. The overarching evaluation of this system working on a 5G SA Private Network across diverse measurement positions, communication protocols, and configurations has yielded valuable insights into the critical factors influencing system performance and user experience. The analysis underscores the intricate interplay between environmental conditions, network infrastructure, and system architecture in shaping the overall efficacy of the teleoperation system. While indoor environments provided a foundation for establishing baseline performance metrics, outdoor conditions exposed the system's vulnerabilities to signal attenuation and interference, necessitating robust communication strategies. The comparative analysis of communication protocols highlighted the trade-offs between latency, reliability, and network utilization, emphasizing the need for careful protocol selection tailored to specific control channel requirements. The investigation of communication configurations demonstrated the impact of data flow patterns on system performance, with each configuration exhibiting distinct strengths and weaknesses in terms of latency, network traffic, and computational resource demands.

6.2 Improvement Possibilities

To further enhance the immersive teleoperation experience, several areas for improvement can be identified. Optimizing communication protocols through adaptive algorithms that dynamically adjust to changing environmental conditions is essential for maintaining consistent performance. Implementing advanced error correction techniques and forward error correction codes can mitigate the impact of packet loss and improve data integrity. Additionally, exploring the integration of edge computing with 5G networks can offload computational tasks, reducing latency and improving overall system responsiveness. To fully exploit the potential of 5G, the development of novel haptic algorithms that can effectively utilize the increased bandwidth and low latency is essential. By optimizing these aspects, the teleoperation system can achieve a higher level of immersion and user satisfaction. Furthermore, the integration of intelligent algorithms for network congestion control and resource allocation can contribute to a more efficient and responsive system.

6.3 Future Developments

Building upon the insights gained from this study, future research can focus on developing innovative teleoperation systems that seamlessly integrate with emerging technologies. The integration of artificial intelligence and machine learning techniques can enable adaptive systems capable of learning user preferences and optimizing performance over time. Furthermore, exploring the semantic correlation between immersion models and QoS/QoE metrics can provide valuable insights for tailoring the system to specific user needs. Additionally, developing dynamic delay recovery procedures can significantly enhance the user experience By combining these advancements, it is envisioned that future teleoperation systems will achieve unprecedented levels of performance, usability, and user satisfaction.

7 Acknowledgments

This work is supported by the European Union's Horizon Europe research and innovation programme (HORIZON-MSCA-2022-DN-01) through the TOAST project under the Marie Skłodowska-Curie grant agreement No. 101073465.

References

[1] Tathababu Addepalli. Compact mimo diversity antenna for 5g sub: 6 ghz (n77/n78 and n79) and wlan (wi-fi 5 and wi-fi 6) band applications. *Wireless Personal Communications*, 132(3):2203–2223, 2023.

[2] Mohammed Al Qaradagi, Fatimah Ameen, Rabab Benotsmane, and Kanstantsin Miatliuk. Application of teleoperation control for a robotic hand using a glove. In *2024 25th International Carpathian Control Conference*, pages 1–6. IEEE, 2024.

[3] Reza Amini Gougeh and Tiago H Falk. Towards instrumental quality assessment of multisensory immersive experiences using a biosensor-equipped head-mounted display. *Quality and User Experience*, 8(1):9, 2023.

[4] Muhammad Awais, Fasih Ullah Khan, Muhammad Zafar, et al. Towards enabling haptic communications over 6g: Issues and challenges. *Electronics*, 12(13):2955, 2023.

[5] Farah Bader and Shahin Rahimifard. A methodology for the selection of industrial robots in food handling. *Innovative Food Science & Emerging Technologies*, 64:102379, 2020.

[6] Luis Basañez, Emmanuel Nuño, and Carlos I Aldana. Teleoperation and level of automation. In *Springer Handbook of Automation*, pages 457–482. Springer, 2023.

[7] Mohamed Behery, Minh Trinh, Christian Brecher, and Gerhard Lakemeyer. Assistive robot teleoperation using behavior trees. *arXiv preprint arXiv:2303.05177*, 2023.

[8] Govinda MG Bezerra, Nicollas R de Oliveira, Tadeu N Ferreira, and Diogo MF Mattos. A comprehensive evaluation of software-defined radio performance in virtualized environments for radio access networks. *Annals of Telecommunications*, pages 1–13, 2024.

[9] bHaptics. Tactile gloves specifications. DK1 Website Datasheet: 2023. Accessed: 2024-07-15.

[10] Pierre-Antoine Cabaret, Thomas Howard, Guillaume Gicquel, Claudio Pacchierotti, Marie Babel, and Maud Marchal. Does multi-actuator vibrotactile feedback within tangible objects enrich vr manipulation? *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[11] Luis Fernando de Souza Cardoso, Bruno Yuji Lino Kimura, and Ezequiel Roberto Zorzal. Towards augmented and mixed reality on future mobile networks. *Multimedia Tools and Applications*, 83(3):9067–9102, 2024.

[12] Giuseppe Caso, Mohammad Rajiullah, Konstantinos Kousias, Usman Ali, et al. An initial look into the performance evolution of 5g non-standalone networks. In *2023 7th Network Traffic Measurement and Analysis Conference*, pages 1–4. IEEE, 2023.

[13] Julio C Cerón, Md Samiul Haque Sunny, Brahim Brahmi, Mendez, et al. A novel multi-modal teleoperation of a humanoid assistive robot with real-time motion mimic. *Micromachines*, 14(2):461, 2023.

[14] Sean Chen, Jensen Gao, Siddharth Reddy, Glen Berseth, Anca D Dragan, and Sergey Levine. Asha: Assistive teleoperation via human-in-the-loop reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7505–7512. IEEE, 2022.

[15] Wanshi Chen, Xingqin Lin, Juho Lee, Antti Toskala, Shu Sun, et al. 5g-advanced toward 6g: Past, present, and future. *IEEE Journal on Selected Areas in Communications*, 41(6):1592–1619, 2023.

[16] Universidad Politecnica de Valencia (UPV). Institute of telecommunications and multimedia applications. https://www.iteam.upv.es. Accessed: 2024-07-11.

[17] Sivaraman Eswaran and Prasad Honnavalli. Private 5g networks: a survey on enabling technologies, deployment models, use cases and research directions. *Telecommunication Systems*, 82(1):3–26, 2023.

[18] Macoumba Fall, Younes Balboul, Mohammed Fattah, Said Mazer, Moulhime El Bekkali, and Ahmed D Kora. Towards sustainable 5g networks: A proposed coordination solution for macro and pico cells to optimize energy efficiency. *IEEE Access*, 11:50794–50804, 2023.

[19] Wen Fan, Xiaoqing Guo, Enyang Feng, Jialin Lin, Yuanyi Wang, et al. Digital twin-driven mixed reality framework for immersive teleoperation with haptic rendering. *IEEE Robotics and Automation Letters*, 2023.

[20] Kai Feng, Qingsong Xu, Seng Fat Wong, and Bin Zi. Design and development of a teleoperated telepresence robot system with high-fidelity haptic feedback assistance. *IEEE Transactions on Automation Science and Engineering*, 2024.

[21] iTEAM Fernando Hernandez. Github repository. https://github.com/xriteamupv/Haptic_Teleop. Accessed: 2024-08-20.

[22] Fivecomm. 5g-broad specifications. Modem Brochure: 2023. Accessed: 2024-07-14.

[23] Srinivasu Garikipati, Gayatri Tangirala, Manikya Krishna Chaitanya Durbhakula, and Virendra Kumar Sharma. Reconfigurable grounds mimo antenna with pin diodes for 5g nr n3/n1/n40/n41/n77/n79 to x-bands. *International Journal of Electronics*, pages 1–23, 2023.

[24] Cosimo Gentile, Giacomo Lunghi, Luca Rosario Buonocore, Francesca Cordella, Mario Di Castro, Alessandro Masi, and Loredana Zollo. Manipulation tasks in hazardous environments using a teleoperated robot: A case study at cern. *Sensors*, 23(4):1979, 2023.

[25] Ayan Ghosh, Daniel Alonso Paredes Soto, Sandor M Veres, and Anthony Rossiter. Human robot interaction for future remote manipulations in industry 4.0. *IFAC-PapersOnLine*, 53(2):10223–10228, 2020.

[26] Basak Gülecyüz, Xiao Xu, Andreas Noll, and Eckehard Steinbach. On the quality-of-learning for haptic teleoperation-based skill transfer over the tactile internet. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[27] Jan Erik Håkegård, Henrik Lundkvist, Ashish Rauniyar, and Peter Morris. Performance evaluation of an open source implementation of a 5g standalone platform. *IEEE Access*, 12:25809–25819, 2024.

[28] Fanghao Huang, Xiao Yang, Tuoyu Yan, and Zheng Chen. Telepresence augmentation for visual and haptic guided immersive teleoperation of industrial manipulator. *ISA transactions*, 2024.

[29] Meipeng Huang, Runhui Feng, Longhao Zou, Rui Li, and Jin Xie. Enhancing telecooperation through haptic twin for internet of robotic things: Implementation and challenges. *IEEE Internet of Things Journal*, 2024.

[30] Ya Huang, Jingkun Zhou, Pingchuan Ke, Xu Guo, et al. A skin-integrated multimodal haptic interface for immersive tactile feedback. *Nature Electronics*, 6(12):1020–1031, 2023.

[31] Suhas Kadalagere Sampath, Ning Wang, Hao Wu, and Chenguang Yang. Review on human-like robot manipulation using dexterous hands. *Cognitive Computation and Systems*, 5(1):14–29, 2023.

[32] Alireza Kamran-Pishhesari, Amin Moniri-Morad, and Javad Sattarvand. Applications of 3d reconstruction in virtual reality-based teleoperation: A review in the mining industry. *Technologies*, 12(3):40, 2024.

[33] Mingyu Kang, Cheol-Gu Gang, Sang-Kyu Ryu, et al. Haptic interface with multimodal tactile sensing and feedback for human–robot interaction. *Micro and Nano Systems Letters*, 12(1):9, 2024.

[34] Sunwook Kim, Ivan Hernandez, Maury A Nussbaum, and Sol Lim. Teleoperator-robot-human interaction in manufacturing: Perspectives from industry, robot manufacturers, and researchers. *IISE Transactions on Occupational Ergonomics and Human Factors*, 12(1-2):28–40, 2024.

[35] Burak Kizilkaya, Olaoluwa Popoola, Guodong Zhao, and Muhammad Ali Imran. 5g-based low-latency teleoperation: Two-way timeout approach. In *Annual Conference Towards Autonomous Robotic Systems*, pages 470–481. Springer, 2023.

[36] Burak Kizilkaya, Changyang She, Guodong Zhao, and Muhammad Ali Imran. Task-oriented prediction and communication co-design for haptic communications. *IEEE Transactions on Vehicular Technology*, 72(7):8987–9001, 2023.

[37] Kai Liang, Wei Guo, Zan Li, Cheng Li, Chunlai Ma, Kai-Kit Wong, and Chan-Byoung Chae. Customizable and robust internet of robots based on network slicing and digital twin. *IEEE Network*, 2024.

[38] Tsung-Chi Lin, Achyuthan Unni Krishnan, and Zhi Li. The impacts of unreliable autonomy in human-robot collaboration on shared and supervisory control for remote manipulation. *IEEE Robotics and Automation Letters*, 8(8):4641–4648, 2023.

[39] Di Liu, Jeonghee Kim, and Youngjib Ham. Multi-user immersive environment for excavator teleoperation in construction. *Automation in Construction*, 156:105143, 2023.

[40] Joe Louca, John Vrublevskis, Kerstin Eder, and Antonia Tzemanaki. Elicitation of trustworthiness requirements for highly dexterous teleoperation systems with signal latency. *Frontiers in Neurorobotics*, 17:1187264, 2023.

[41] Jing Luo, Wei He, and Chenguang Yang. Combined perception, control, and learning for teleoperation: key technologies, applications, and challenges. *Cognitive Computation and Systems*, 2(2):33–43, 2020.

[42] Honghao Lv, Depeng Kong, Gaoyang Pang, Baicun Wang, et al. Gulim: A hybrid motion mapping technique for teleoperation of medical assistive robot in combating the covid-19 pandemic. *IEEE Transactions on Medical Robotics and Bionics*, 4(1):106–117, 2022.

[43] Baraka Maiseli, Abdi T Abdalla, Libe V Massawe, Mercy Mbise, et al. Brain–computer interface: trend, challenges, and threats. *Brain informatics*, 10(1):20, 2023.

[44] Sarmad Mehrdad, Fei Liu, Minh Tu Pham, Arnaud Lelevé, and S Farokh Atashzar. Review of advanced medical telerobots. *Applied Sciences*, 11(1):209, 2020.

[45] MD Moniruzzaman, Alexander Rassau, Douglas Chai, and Syed Mohammed Shamsul Islam. Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey. *Robotics and Autonomous Systems*, 150:103973, 2022.

[46] Raheeb Muzaffar, Mahin Ahmed, Emiliano Sisinni, Thilo Sauter, and Hans-Peter Bernhard. 5g deployment models and configuration choices for industrial cyber-physical systems–a state of art overview. *IEEE Transactions on Industrial Cyber-Physical Systems*, 2023.

[47] MSI Espa na. Msi cyborg-15 specifications. A12UCX-657XES Website Datasheet: 2023. Accessed: 2024-07-18.

[48] MSI Espa na. Msi modern-14 specifications. C12M-077XES Website Datasheet: 2024. Accessed: 2024-07-18.

[49] Syeda Nadiah Fatima Nahri, Shengzhi Du, and Barend Jacobus Van Wyk. A review on haptic bilateral teleoperation systems. *Journal of Intelligent & Robotic Systems*, 104(1):13, 2022.

[50] Jun Nakanishi, Shunki Itadera, Tadayoshi Aoyama, and Yasuhisa Hasegawa. Towards the development of an intuitive teleoperation system for human support robot using a vr device. *Advanced Robotics*, 34(19):1239–1253, 2020.

[51] NASN Nandasena, WAA Vimukthi, HMKKMB Herath, R Wijesinghe, and SLP Yasakethu. Real-time upper body motion tracking using computer vision for improved human-robot interaction and teleoperation. In *2023 Moratuwa Engineering Research Conference (MERCon)*, pages 201–206. IEEE, 2023.

[52] Institute of Telecommunications and Multimedia Applications (iTEAM). Immersive and holographic advanced 5g laboratory. iTEAM Website Project: 2022. Accessed: 2024-07-11.

[53] Institute of Telecommunications and Multimedia Applications (iTEAM). The iteam opens its new laboratory of immersive communications to upv students. iTEAM Website Blog: 2024-03-06. Accessed: 2024-07-11.

[54] OnRobot. Grippers specifications. RG2 Website Datasheet: 2022. Accessed: 2024-07-15.

[55] Claudio Pacchierotti and Domenico Prattichizzo. Cutaneous/tactile haptic feedback in robotic teleoperation: Motivation, survey, and perspectives. *IEEE Transactions on Robotics*, 2023.

[56] Claudia Pérez-D'Arpino, Rebecca P Khurshid, and Julie A Shah. Experimental assessment of human-robot teaming for multi-step remote manipulation with expert operators. *ACM Transactions on Human-Robot Interaction*, 2023.

[57] Realsense. Camera specifications. D415 Website Datasheet: 2023. Accessed: 2024-07-15.

[58] Universal Robots. Ur e-series specifications. UR5e Website Datasheet: 2022. Accessed: 2024-07-14.

[59] Ahmed Eslam Salman and Magdy Raouf Roman. Augmented reality-assisted gesture-based teleoperated system for robot motion planning. *Industrial Robot: the international journal of robotics research and application*, 50(5):765–780, 2023.

[60] Gaayathri Sankar, Soussan Djamasbi, Zhi Li, Jing Xiao, and Norbou Buchler. Systematic literature review on the user evaluation of teleoperation interfaces for professional service robots. In *International Conference on Human-Computer Interaction*, pages 66–85. Springer, 2023.

[61] Daniel Scholl, Jannis Meier, and Michael Reke. Vr-enhanced teleoperation system for a semi-autonomous mulitplatform all-terrain exploration vehicle. In *International Conference on Human-Computer Interaction*, pages 268–285. Springer, 2024.

[62] Aditya Kumar Singh, Ajay Kumar Dwivedi, Chandrabhan Jha, Sweta Singh, et al. A compact mimo antenna for 5g nr frequency bands n257/n258/n261 under millimeter-wave communication. *IETE Journal of Research*, 69(12):8561–8573, 2023.

[63] Harvey Stedman, Basaran Bahadir Kocer, Nejra Van Zalk, Mirko Kovac, and Vijay M Pawar. Evaluating immersive teleoperation interfaces: Coordinating robot radiation monitoring tasks in nuclear facilities. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11972–11978. IEEE, 2023.

[64] Krzysztof Adam Szczurek, Raul Marin Prades, Eloise Matheson, Jose Rodriguez-Nogueira, and Mario Di Castro. Multimodal multi-user mixed reality human–robot interface for remote operations in hazardous environments. *IEEE Access*, 11:17305–17333, 2023.

[65] Houriiyah Tegally, James Emmanuel San, Jennifer Giandhari, and Tulio de Oliveira. Unlocking the efficiency of genomics laboratories with robotic liquid-handling. *BMC genomics*, 21:1–15, 2020.

[66] Allal Tiberkak, Abdelfetah Hentout, and Abdelkader Belkhir. Webrtc-based mosr remote control of mobile manipulators. *International Journal of Intelligent Robotics and Applications*, 7(2):304–320, 2023.

[67] Ozan Tokatli, Pragna Das, Radhika Nath, Luigi Pangione, et al. Robot-assisted glovebox teleoperation for nuclear industry. *Robotics*, 10(3):85, 2021.

[68] Giacomo Valle, Natalija Katic Secerovic, Dominic Eggemann, et al. Biomimetic computer-to-brain communication enhancing naturalistic touch sensations via peripheral nerve stimulation. *Nature Communications*, 15(1):1151, 2024.

[69] Yuhui Wan, Jingcheng Sun, Christopher Peers, Joseph Humphreys, Dimitrios Kanoulas, and Chengxu Zhou. Performance and usability evaluation scheme for mobile manipulator teleoperation. *IEEE Transactions on Human-Machine Systems*, 2023.

[70] Xingchao Wang, Shuqi Guo, Zijian Xu, Zheyuan Zhang, Zhenglong Sun, and Yangsheng Xu. A robotic teleoperation system enhanced by augmented reality for natural human–robot interaction. *Cyborg and Bionic Systems*, 5:0098, 2024.

[71] Xin Wei, Dan Wu, Liang Zhou, and Mohsen Guizani. Cross-modal communication technology: A survey. *Fundamental Research*, 2023.

A Article Proposals

This section outlines three potential research avenues that delve into the intricacies of immersive teleoperation systems. The proposed articles explore distinct facets of the system, ranging from communication protocols and network optimization to the impact of haptic rendering and human-robot interaction on the overall user experience.

A.1 Article 1: Configuration, Evaluation and Optimization of a Haptic Teleoperation Experience

Abstract: This paper delves into the intricate relationship between communication protocols, system configuration, and haptic feedback in shaping the overall immersive teleoperation experience. By meticulously evaluating diverse network topologies, control algorithms, and haptic rendering models, we aim to optimize system performance and enhance user engagement. The study explores the impact of various communication configurations on latency, jitter, and packet loss, identifying optimal combinations for real-time control. Furthermore, the investigation encompasses the role of robot trajectory, grip width, and force models in generating realistic and informative haptic feedback. Through rigorous experimentation and data analysis, this research provides valuable insights into the design and optimization of immersive teleoperation systems, paving the way for future advancements in human-robot interaction.

Keywords: Haptic Teleoperation, Immersive Experience, Robot Trajectory, Grip Control, Tactile Feedback

Note: Intended to be created in collaboration with Technische Universität Munchen (TUM) within the framework of TOAST Doctoral Network.

A.2 Article 2: Advantages of Asynchronous Variable-Length PDUs for Dynamic Delay in Immersive Telemanipulations

Abstract: This paper investigates the efficacy of asynchronous variable-length protocol data units (PDUs) in mitigating the challenges posed by static, arbitrary or cumulative delay in immersive telemanipulation systems. By departing from traditional synchronous communication paradigms, this approach enables adaptive and flexible data exchange with dynamic self-managed delay, enhancing system responsiveness and user experience. Empirical analysis demonstrates the potential of asynchronous variable-length PDUs in reducing latency and improving overall system performance, despite the inherent trade-offs in network efficiency. The findings underscore the importance of tailored communication strategies for achieving optimal results in complex, real-time applications such as long-distance telemanipulation.

Keywords: Asynchronous Communication, Variable-Length Packages, Long-Distance Telemanipulation, Dynamic Delay, Immersive Experience.

Note: Intended to be created in collaboration with Technische Universität Dresden (TUD) within the framework of TOAST Doctoral Network.

A.3 Article 3: Quality of Immersive Services and Experiences for Remote Tactile Applications

Abstract: This paper delves into the intricate relationship between quality of service and experience (QoS/QoE) metrics and the immersive experience afforded by remote tactile applications. By scrutinizing a diverse array of telemanipulation scenarios, encompassing both gripper and robotic hand operations, we explore the critical factors influencing the perception of presence, engagement, and control. The study investigates the interplay of network parameters, haptic rendering techniques, and human factors in shaping the overall user experience. Through rigorous experimentation and analysis, this research aims to establish a comprehensive framework for evaluating and optimizing QoS/QoE in remote tactile applications, paving the way for future advancements in telepresence and human-robot collaboration.

Keywords: Immersive Telemanipulation, Quality of Service, Haptic Feedback, Human-Robot Interaction, User Experience

Note: Intended to be created in collaboration with Universitetet i Oslo (UiO) within the framework of the Eurocluster INGENIOUS and INGENIOUS project.

B User Manual

This section provides a practical description for the requirements and installation procedure of each node, as well as overall system initialization and utilization sequence for intended normal functioning. Each subsection is accompanied by required commands and instructions to better understand the purpose of each stage of the process.

B.1 Local Control and Orchestration Node (LCON)

- Control of the trajectories of robot arm using coordinates and orientation together with gripper movements (width, force/speed) limited to their defined spatial domain.
- Object detection when current width is different than the objective after a configured time. Grip levels configuration for different objective widths and width variations.

B.1.1 LCON Requirements

Hardware Requirements:

- UR e-Series Robot Arm
- UR e-Series Teach Pendant
- OnRobot Gripper RG2/RG6
- OnRobot HEX-E QC Stand

Software Requirements:

- UR PolyScope v.5.16.0
- ROS Noetic v.1.16.0
- Base OS: Ubuntu 20.04
- Python 3.8.3 (Python 3)

B.1.2 LCON Installation and Configuration

Follow these steps to properly install the Software of Local Node, prior to its utilization.

1. Create workspace:

```
mkdir -p ./catkin_ws/src
```

2. Add Universal Robot packages:

```
cd ./catkin_ws/src
git clone https://github.com/fmauch/universal_robot.git
cd universal_robot
git reset --hard 1ffdd69181389b14b7d6342f0c5bad3b45c5e32f
```

3. Clone the repository including the LCON_Local_Node folder:

```
cd <destination_folder>
git clone https://github.com/xriteamupv/Haptic_Teleop.git
```

4. Keep the LCON_Local_Node folder contents and remove the rest:

```
cd <destination_folder>
mv ./Haptic_Teleop-main/LCON_Local_Node/* /catkin_ws/src
rm -r ./Haptic_Teleop-main
```

5. Install Python libraries:

```
pip install numpy
pip install rospy
pip install actionlib
pip install datetime
pip install pymodbus
```

6. Compile packages:

```
cd /catkin_ws
catkin_make
```

B.2 Remote Teleoperation and Haptic Node (RTHN)

- Capture of the hand position and movements of the human arm through a video camera or tracking device, and generate vibro-tactile sensations on a haptic glove.
- Recognize different gripper configurations, including types of grips (soft, medium, hard) and how many fingers (2F, 3F, 4F, 5F) are involved in the grip.

B.2.1 RTHN Requirements

Hardware Requirements:

- bHaptic TactGloves DK1/DK2
- RealSense D415 Camera or similar
- Alternatively: Camera 720p 30fps

Software Requirements:

- bHaptics Player v2.3.5 or newer
- Base OS: Windows 10 v1703 or newer
- Python 3.11.3 (Python 3)

B.2.2 RTHN Installation and Configuration

Follow these steps to properly install the Software of Remote Node, prior to its utilization.

1. Clone the repository including the RTHN_Remote_Node folder:

```
cd <destination_folder>
git clone https://github.com/xriteamupv/Haptic_Teleop.git
```

2. Keep the RTHN_Remote_Node folder contents and remove the rest:

```
move /y ./RTHN_Remote_Node/* ./
rmdir /s ./RTHN_Remote_Node
rmdir /s ./LCON_Local_Node
rmdir /s ./Teloperation_Local_Node
```

3. Modify the following files with your corresponding directories:

```
cd ./haptic_ur5e/src
nano ./utils/Classifier.py
nano ./model/keypoint_classifier/keypoint_classifier.py
nano ./model/point_history_classifier/point_history_classifier.py
```

   If you decide to use Windows Subsystem for Linux (WSL), you have to modify the directories of:

```
./init_remote_node.sh
./initialize_remote_node.sh
```

4. Install Python libraries:

```
pip install numpy
pip install datetime
pip install pymodbus
pip install websocket
pip install websocket-client
pip install opencv-python
pip install mediapipe
pip install tensorflow
```

B.3 System Initialization and Utilization

The system allows a user to manipulate remote objects imitating the movements of their own arm through movement capture sensors. The experience is enriched by haptic feedback, providing the user with a real contact sensation with the manipulated object.

Follow these steps for initializing the entire system, i.e. both LCON and RTHN. Make sure to have access and control to all system devices and terminals. Respecting the steps sequence is recommended.

(A) **Turn on and connect all devices from LCON and RTHN**

(B) **RTHN - Put your TactGloves on and link them to bHaptics Player**

(C) **RTHN - Open a terminal and run the *haptic_control.py* file**

This operation will open the receiving end of the communication between nodes, and connect to bHaptics SDK.

```
cd <folder_location>/haptic_ur5e
python haptic_control.py <param1> <value1> <param2> <value2> ...
```

Success Indicator:

```
LINK ESTABLISHED. Waiting for commands from Local Node ...
```

Note: Change *paramX* and *valueX* as needed to modify default application.

(D) **LCON - Launch ROS package for URXe and XML-RPC for RG2 communication**

If successful, this operation will set the communication between nodes, and connect with Robots through Polyscope.

```
cd <ros_workspace>
. init_controllers.sh --model <ur_model> --robot_ip <robot_ip> //
--calibration <calibration_file.yaml>
```

Success Indicator:

3 tabs will open in the same window, successfully running UR Robot Arm bringup, XML-RPC server and Gripper Controller.

Note: Change *ur_model*, *robot_ip* and *calibration_file.yaml* as needed if you want to modify default application.

(E) **Teach Pendant - Load and start program *rg2_remote_control.urp* in Polyscope**

This URScript program contains the Variables Setup, the Robot Program with instructions to connect to host, and the RGX Gripper initializer. It should be previously loaded via USB to the URXe Teach Pendant containing Polyscope.

Success Indicator:

```
Robot requested program.
Sent program to robot.
Robot ready to receive commands.
```

(F) **LCON - Run the *robot_control.py* and *gripper_control.py* programs**

This operation will established the communication between the nodes and connect to the local Robots Movement Controllers.

```
cd <ros_workspace>
. init_local_node.sh <param1> <value1> <param2> <value2> ...
```

Success Indicator:

2 tabs will open in the same window, one for each program.

Output on each console:

```
LINK ESTABLISHED. Waiting for commands from Tracking Program ...
```

Note: Change *paramX* and *valueX* as needed to modify default application.

(G) **RTHN - Open another terminal and run the *tracking_control.py* program**

This operation will activate the whole system. By default, a camera window should be displayed, which will start tracking the user's hand.

```
cd <folder_location>/haptic_ur5e
python tracking_control.py <param1> <value1> <param2> <value2> ...
```

Success Indicator:

A camera window appears and starts the hand movements tracking.

Output on Tracking console:

```
System started, currently tracking your hand ...
```

Note: Change *paramX* and *valueX* as needed to modify default application.

(H) **Enjoy the experience!**

(a) Ensure the tracking system correctly detects your hand landmarks, successfully identifies its coordinates and properly follows your movements:
Robot State: Inactive.

(b) Observe a red dot that should appear in the landmarks baricenter, thus indicating the hand is NOT within the tolerance of the initial reference position.

(c) Activate the robot and gripper control by making a Pointer gesture with index finger or OK gesture by touching thumb and index in initial reference region:
Robot State: Active.

(d) Observe a green dot that should appear in the landmarks baricenter, thus indicating the hand is within the tolerance of the initial reference position.

(e) Telemanipulate the robot arm by moving your hand within camera range and making appropriate hand gestures for operating on the gripper. When an object is gripped, you will feel an adequate feedback on the tactile glove.

(f) Deactivate the robot and gripper control by repeating procedure on Step (c).

(g) Press ESC key to stop the tracking system and close the entire system.

C Software Examination

This section provides a comprehensive overview of the underlying software architecture that drives the system's functionality.

C.1 Controllers Operation

First, we delve into the core functionalities that govern the system's physical components.

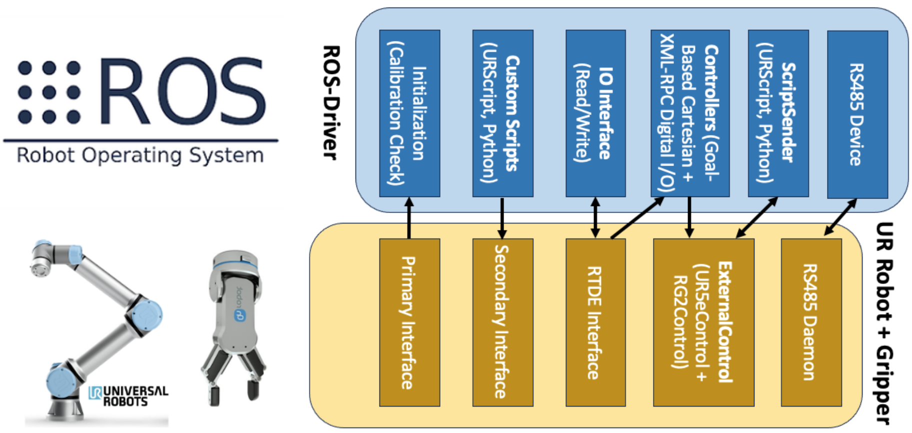


Figure 25: Overview of the ROS-Driver architecture for Robot Arm and Gripper.

C.1.1 Robot Arm: Trajectory Controllers

Universal Robots offers a suite of goal-based Cartesian trajectory controllers accessible through the Teach Pendant with PolyScope interface, and specific topics and messages types with ROS on Ubuntu OS. These recommended controllers, available to use in system, leverage robot kinematics and dynamics to achieve precise and flexible motion planning.

1. **Controller ID**: *pose_based_cartesian_traj_controller*
   - **Type**: *pose_controllers/CartesianTrajectoryController*
   - **Function**: Accepts desired end-effector poses as input.
   - **Processing**: ROS continuously sends these poses to the robot controller.
   - **Kinematics**: The robot's internal inverse kinematics engine calculates the corresponding joint commands for each pose. This approach ensures smooth and accurate execution based on the robot's actual configuration.
   - **Benefits**: Precise control over end-effector position and orientation.
2. **Controller ID**: *joint_based_cartesian_traj_controller*
   - **Type**: *position_controllers/CartesianTrajectoryController*

- **Function**: Takes desired Cartesian waypoints as input.
- **Processing**: ROS performs inverse kinematics calculations to convert these waypoints into joint space commands.
- **Kinematics**: Unlike pose_based, the ROS side handles the inverse kinematics, offering more flexibility for customization.
- **Benefits**: Balance between Cartesian intuitiveness and computational efficiency, suitable for various applications.

3. **Controller ID**: *forward_cartesian_traj_controller*
    - **Type**: *pass_through_controllers/JointTrajectoryController*
    - **Function**: Directly receives the complete desired Cartesian trajectory.
    - **Processing**: ROS forwards the entire trajectory to the robot controller without any interpolation.
    - **Kinematics**: The robot's internal forward and inverse kinematics handle trajectory execution. This method offers real-time performance and leverages the robot's optimized control mechanisms.
    - **Benefits**: Precise control over the Cartesian path, similar to motions programmed on the Teach Pendant. However, ROS doesn't check for configuration changes, potentially leading to safety stops.

While ROS interacts with the robot and potentially performs some calculations, the core kinematics processing (both forward and inverse) primarily happens within the Universal Robots controller. This ensures real-time performance and efficient interaction with the robot's low-level control mechanisms. The choice of controller depends on the desired balance between user control, computational efficiency, and motion fidelity.

C.1.2 Gripper: Multithreading and XML-RPC

To ensure synchronized and efficient operation between the robot arm and the gripper, a multithreading approach is employed. This strategy enables concurrent execution of robot arm movements and gripper control within separate threads.

The robot arm thread handles the overall robot motion, utilizing the robot's built-in control system. A brief delay between instructions is incorporated to allow for potential synchronization adjustments but does not considerably affect user experience.

The gripper thread employs XML-RPC communication to interact with the OnRobot RG2 gripper. XML-RPC is a protocol that allows for communication between different systems and languages by executing functions on a remote system as if they were local. On the server side, the gripper's control unit acts as an XML-RPC server, exposing functions like *open*, *close*, *get_position*, etc. On the client side, the main application acts as an XML-RPC client, making calls to the server's functions.

The system establishes a connection to the gripper's XML-RPC server and continuously monitors for incoming commands. Upon receiving a *MOVE COMMAND*, the thread retrieves the desired gripper position and force from the XML-RPC server and executes the grip operation using the *rg_grip* function. Subsequently, the thread updates the gripper's actual position through the XML-RPC interface.

The control script provides the necessary framework for both threads, including variable initialization, robot program setup, and XML-RPC communication parameters. It also incorporates functions for gripper control, data acquisition, and TCP/IP communication with gripper. This multithreading architecture offers several advantages:

- **Improved responsiveness**: By separating robot arm and gripper control into independent threads, the system can handle concurrent tasks more efficiently.
- **Flexibility**: The XML-RPC interface allows for easy integration of different gripper models and communication protocols.
- **Modularity**: The code structure promotes code reusability and maintainability.

C.2 Tracking Algorithms

Next, we delve into the core algorithms responsible for capturing and interpreting human hand movements.

C.2.1 Hand Landmarks Detection

MediaPipe's Hand Landmarker employs a deep neural network architecture specifically tailored for hand detection and landmark localization. The model is trained on a massive dataset of hand images, encompassing diverse hand poses, skin tones, lighting conditions, and occlusions. This robust training regimen enables the model to generalize well to real-world scenarios.

The network architecture likely incorporates a combination of convolutional layers for feature extraction, followed by recurrent or attention-based layers for capturing spatial and temporal dependencies. This design allows the model to effectively handle complex hand shapes and dynamic movements.

The initial stage of hand detection involves a lightweight palm detection model that scans the entire image for potential hand regions. This model is optimized for speed and accuracy, ensuring efficient processing. Once a palm is detected, a bounding box is generated around it to define the region of interest for the landmark detection stage.

To improve the accuracy of the hand localization, the system often employs a refinement process. This involves iteratively adjusting the bounding box based on the detected hand landmarks. By refining the bounding box, the subsequent landmark detection stage can focus on a more precise region, leading to enhanced accuracy.

The core of the detection algorithm is the landmark localization model. This model takes the cropped hand image as input and predicts the 21 key points' coordinates. The model is trained to accurately localize these points, even in challenging conditions such as occlusion or low-light environments.

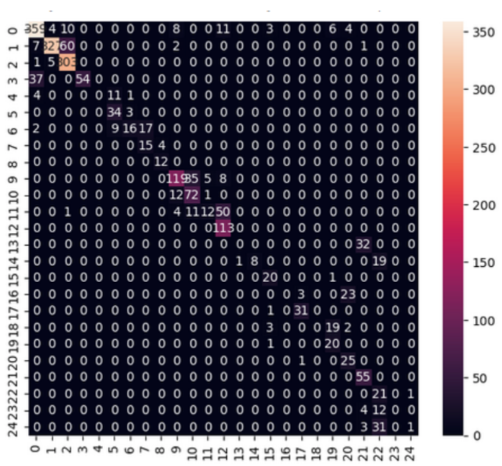
To further improve the accuracy of the landmark positions, the system incorporates refinement techniques. These techniques involve iterative refinement of the landmark positions based on local image features or contextual information. Additionally, post-processing steps, such as smoothing or filtering, are applied to reduce noise and improve the overall landmark quality.

To achieve real-time performance, MediaPipe Hand Landmarker incorporates several optimization strategies. These include model quantization, hardware acceleration, and efficient algorithm design. By optimizing the model for target platforms and using hardware accelerators, the system process video frames at high frame rates.

C.2.2 Grip Gestures Recognition

From the detected hand and its estimated pose, a set of informative features is extracted. These features encompass the relative positions of various key points, the hand's overall shape, and its orientation in the image or video frame. Feature extraction plays a critical role in gesture recognition, as it condenses the hand's characteristics into a compact representation that is readily processed by the subsequent classification stage.

Intermediate-Stage Classification Model with the Training of 24 New Gestures



Classification Report

	precision	recall	f1-score	support
0	0.88	0.89	0.88	405
1	0.97	0.82	0.89	397
2	0.81	0.98	0.89	309
3	1.00	0.59	0.74	91
4	0.00	0.00	0.00	16
5	0.63	0.92	0.75	37
6	0.80	0.36	0.50	44
7	0.47	0.79	0.59	19
8	0.75	1.00	0.86	12
9	0.82	0.71	0.76	167
10	0.61	0.85	0.71	85
11	0.67	0.15	0.25	78
12	0.62	1.00	0.77	113
13	0.00	0.00	0.00	32
14	1.00	0.29	0.44	28
15	0.71	0.95	0.82	21
16	0.00	0.00	0.00	26
17	0.89	0.97	0.93	32
18	0.00	0.00	0.00	24
19	0.43	0.95	0.60	21
20	0.46	0.96	0.62	26
21	0.58	1.00	0.73	55
22	0.25	0.95	0.40	22
23	0.00	0.00	0.00	16
24	0.50	0.03	0.05	35
accuracy			0.77	2111
macro avg	0.55	0.61	0.53	2111
weighted avg	0.77	0.77	0.74	2111

Figure 26: Classification output for the Training of 24 gestures related to grip recognition. The Final-Stage Classification Model includes a greater amount of gestures states which complicates its visual representation.

The extracted features are then fed into a machine learning model, specifically a Multi-Layer Perceptron (MLP) in this case in particular. The MLP is tasked with classifying the hand gesture based on the provided features. Through a training process on a curated dataset of labeled hand gestures, the MLP learns to associate specific feature patterns with corresponding hand gestures.

Upon processing the features through the MLP, the model generates an output that signifies the recognized hand gesture. This output contains a class label corresponding to a predefined gesture and a more nuanced representation of the hand pose.

D Code Structure

This section delves into the code and program specifications defined for the configured system, which comprises both object-oriented and procedural programming. In particular, the next subsections focus on the non-static classes structure for the Local (LCON) and Remote (RTHN) Nodes, independently used in the four main programs of the haptic teleoperation system: *robot_control.py*, *gripper_control.py*, *tracking_control.py* and *haptic_control.py*. Each program interacts with their classes structure through one main access point for simpler and quicker debugging and error detection, remarked with thicker lines in the following UML Class Diagrams.

D.1 LCON Classes Description

The classes used in the programs *robot_control.py* and *gripper_control.py* are subdivided into 2 subgroups focused in the robot trajectory management and the gripper movement control respectively. This modular approach allows for further customizations and future scalability and ease-of-configuration.

D.1.1 Robot Control

This subgroup of classes is responsible for the robot trajectory management. This involves the robot controller properties specification for initialization and execution, the human arm positions received from the Tracking Intelligence, the equivalent robot arm positions, and the possible transmission of grip information to the Gripper Movement Control.

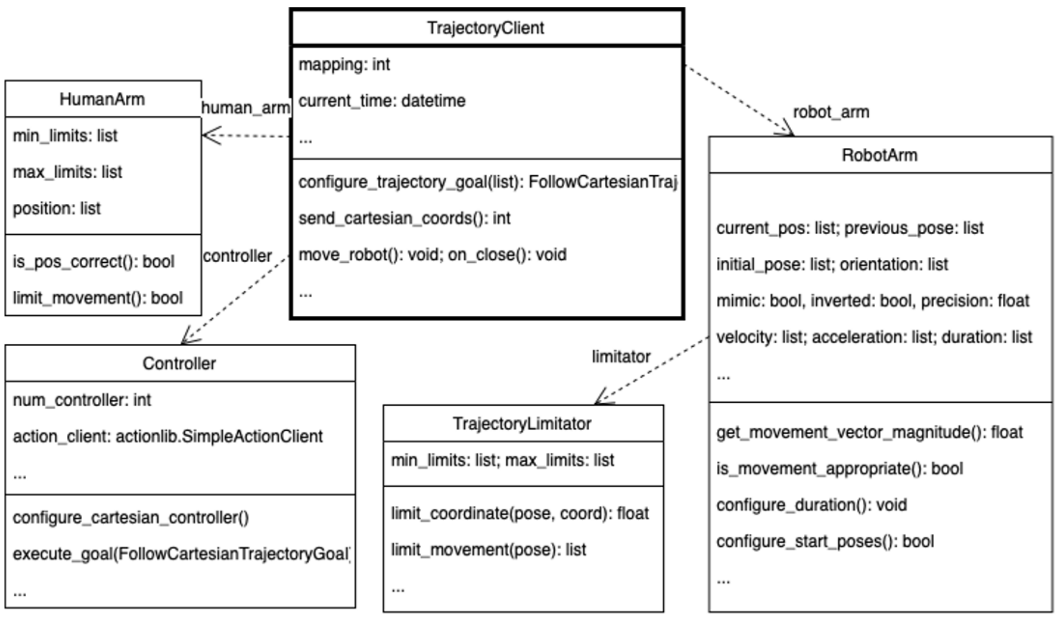


Figure 27: Class diagram of the subgroup utilized for robot control procedures.

As shown in Figure 27, the main class is called *TrajectoryClient* and is responsible for the general trajectory management helped by instances of auxiliary classes *Controller*, *HumanArm* and *RobotArm*. The first class manages all operations related to the *ActionClient* Service that interacts with the selected Goal-based Cartesian Trajectory Controllers provided by the system. The other auxiliary classes are needed for processing input tracking coordinates and output robot movement coordinates respectively, including the application of customization options.

The class *RobotArm* depends on *TrajectoryLimitator* that controls the coordinates limits for the robot movements. The default attribute values of this last class needs to be modified when changing the robot arm model and subsequently its size and dimensions.

D.1.2 Gripper Control

This subgroup of classes is required for the gripper actions derivations. This includes the gripper XML-RPC client specification and its synchronization with the PolyScope Thread, the gripper levels received from the Tracking Intelligence and the grip characteristics transmitted to the Haptic Enablers Conditioning.

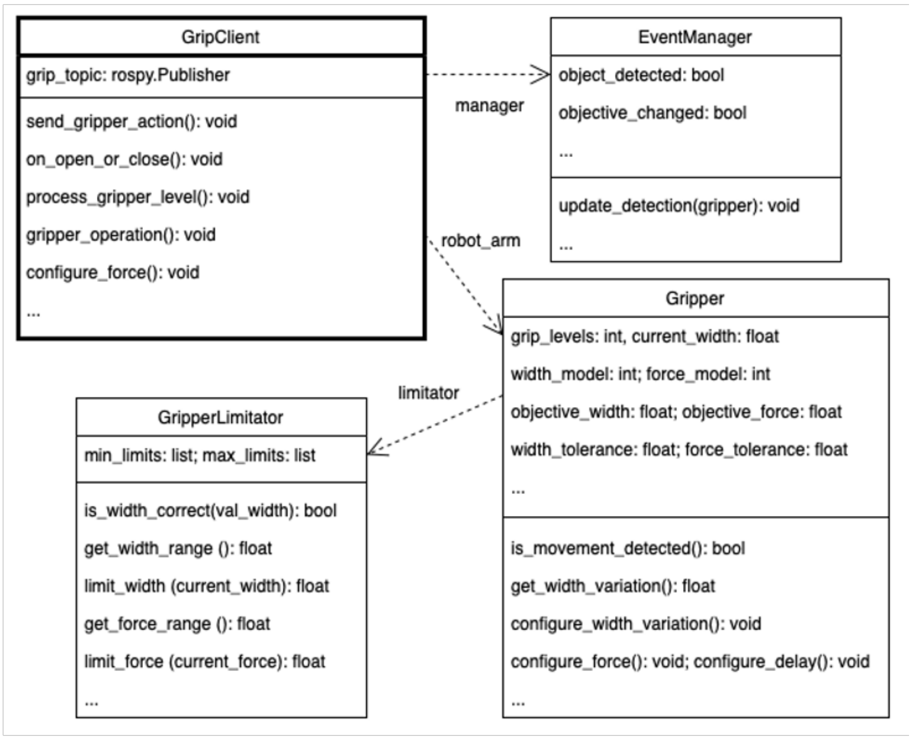


Figure 28: Class diagram of the subgroup utilized for gripper control procedures.

Figure 28 depicts the general structure used for this program whose main class is called *GripClient* and depends on auxiliary classes *EventManager* and *Gripper*. The former class has the state information result of the gripper actions, such as if an object has been gripped, or the objective width has changed. The latter class entails the device core functionality characteristics that are modified in accordance to the tracking data.

Similarly to the *RobotArm* class, the *Gripper* class depends on a limitator class which needs to be modified in case the gripper model is changed.

D.2 RTHN Classes Description

The classes used in the programs *tracking_control.py* and *haptic_control.py* are subdivided into 2 subgroups focused in the tracking intelligence and the haptic enablers conditioning respectively. This modular approach allows for further customizations and future scalability and ease-of-configuration.

D.2.1 Tracking Control

This subgroup of classes is responsible for the human movement tracking and hand gestures recognition. This involves the tracking algorithms specification for model configuration, the hand landmarks visualization for user feedback, the system state for action registries, and the pre-processed keypoints history for movement predictions.

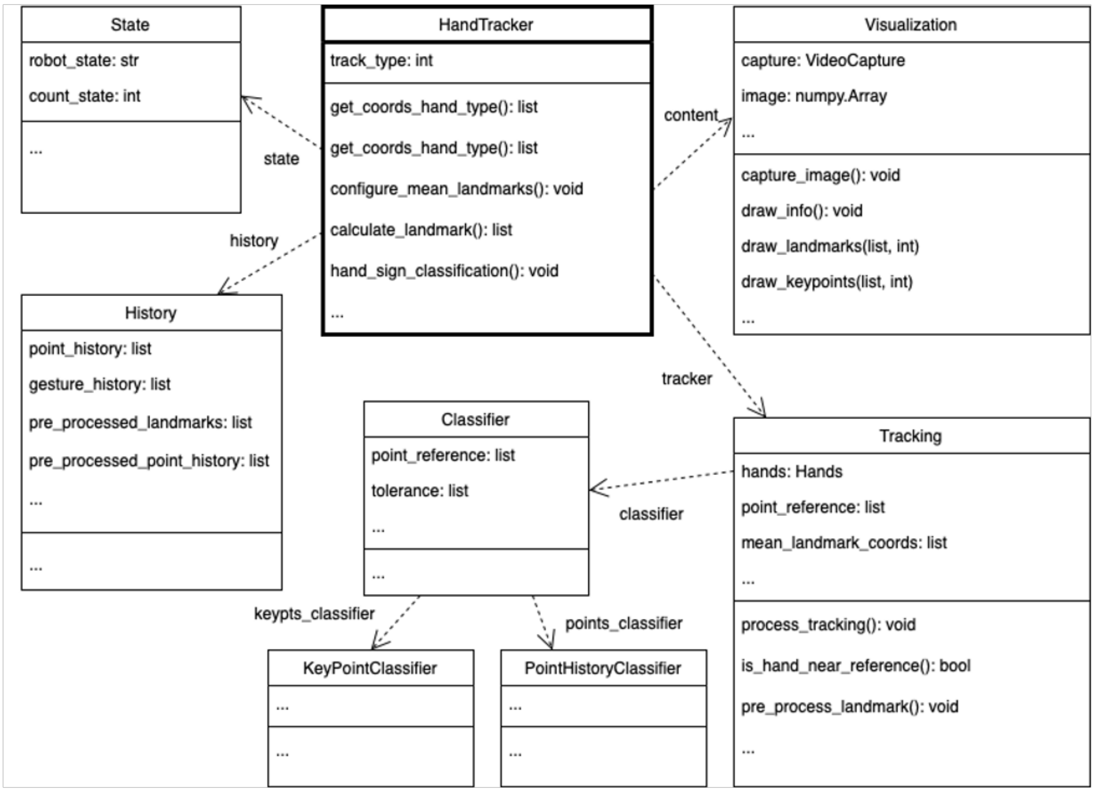


Figure 29: Class diagram of the subgroup utilized for tracking control procedures.

As depicted in Figure 29, the main class for tracking control is HandTracker, which encompass four auxiliary classes: *History* for preprocessing and gesture prediction, *State* for system management similar to *EventManager* for Gripper Control, *Visualization* for GUI implementation, and *Tracking* for hand detection logic definition. The latter class depends on Classifiers, either *KeyPointClassifier* or *PointHistoryClassifier*, required for tracking both present and past recollection of hand landmarks. Note that this subgroup of classes do not require any explicit limitator, as the video processing will always keep the same coordinate ranges independent of the camera model or quality.

D.2.2 Haptic Control

This subgroup of classes is responsible for the conditioning of haptic enablers received from the Tracking Intelligence and the Trajectory Management, intended to provide the Haptic Feedback node with pre-analyzed characteristics (i.e. vibration intensity and duration) and configurations (i.e. actuator/s to activate) for the tactile gloves.

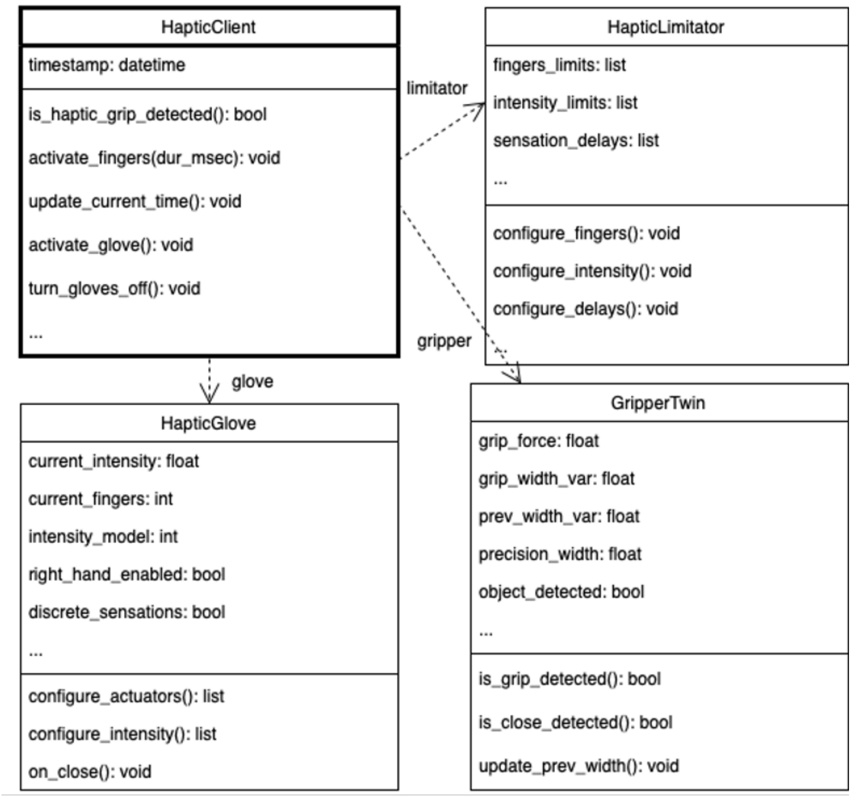


Figure 30: Class diagram of the subgroup utilized for haptic control procedures.

Figure 30 shows the main class for haptic control called *HapticClient*, and its three auxiliary classes. *HapticGlove* contains the structure with the properties needed to manage the haptic device, such as vibration intensity, fingers or amount of actuators to activate and which hand is being utilized. *GripperTwin* is needed in order to facilitate keeping track of the gripper state and characteristics while retaining a brief history of grip width variations so to avoid false negatives in the object detection. Finally, *HapticLimitator* provides the minimum and maximum limits for the interested parameters, which can be modified using customization options or changing the code in case of switching models of haptic gloves, or if it is demanded due to user sensitivity issues.

E Helper Mappers and Procedures

This section presents an overview of the core mathematical models and algorithms employed to customize various aspects of the teleoperation system. These models encompass the mapping of human inputs to robot outputs, the management of system delays, and the generation of haptic feedback.

E.1 Models for Robot Trajectory

Four distinct mapping models are explored: linear mapping, focused linear mapping, multi-focused linear mapping, and non-linear mapping. Each model employs varying degrees of complexity and adaptability to approximate the desired relationship between human hand coordinates tracked by a camera device, and the robot arm motion whose coordinates are controlled by goal-based cartesian trajectory controllers.

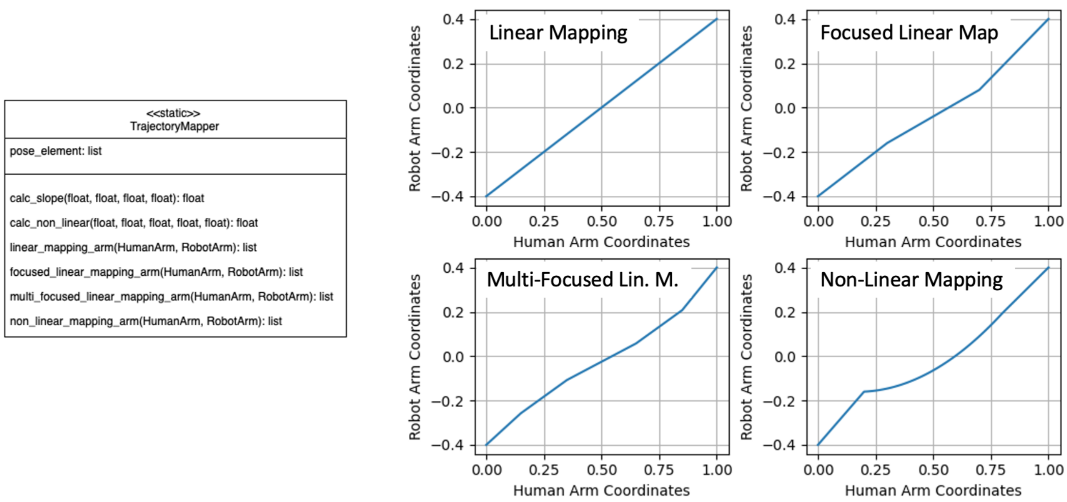


Figure 31: Different models and mappers used for Robot Trajectory computations.

The **Linear Mapping** implements a straightforward linear mapping between human and robot arm coordinates. This model assumes a proportional relationship between the two systems, with scaling factors applied to account for differences in workspace dimensions and kinematic characteristics. While computationally efficient, often results in a simplified and less intuitive motion mapping, leading to a reduced sense of control.

The **Focused Linear Mapping** introduces a more sophisticated approach by dividing the human arm workspace into focused regions. Within each region, a distinct linear mapping is applied, allowing for greater flexibility and adaptability to non-linear relationships between human and robot motion. This model aims to enhance the user's perception of system responsiveness and accuracy.

The **Multi-Focused Linear Mapping** extends the focused linear mapping concept by incorporating multiple focused regions, further refining the mapping accuracy. This model offers increased adaptability to complex human arm movements and tries to approximate non-linear robot kinematics with simpler processing requirements.

Finally, the **Non-Linear Mapping** employs a parabolic transformation to capture intricate relationships between human and robot arm motion. This model utilizes polynomial functions to approximate the mapping, providing a higher degree of freedom for capturing complex kinematic patterns. However, it requires careful parameter tuning and computational resources to achieve optimal performance.

E.2 Models for Gripper Width

This subsection explores various techniques for mapping a desired gripper width (objective width) based on a user-specified grip level (0-4). These models offer varying levels of complexity and adaptability, catering to different requirements and gripper characteristics.

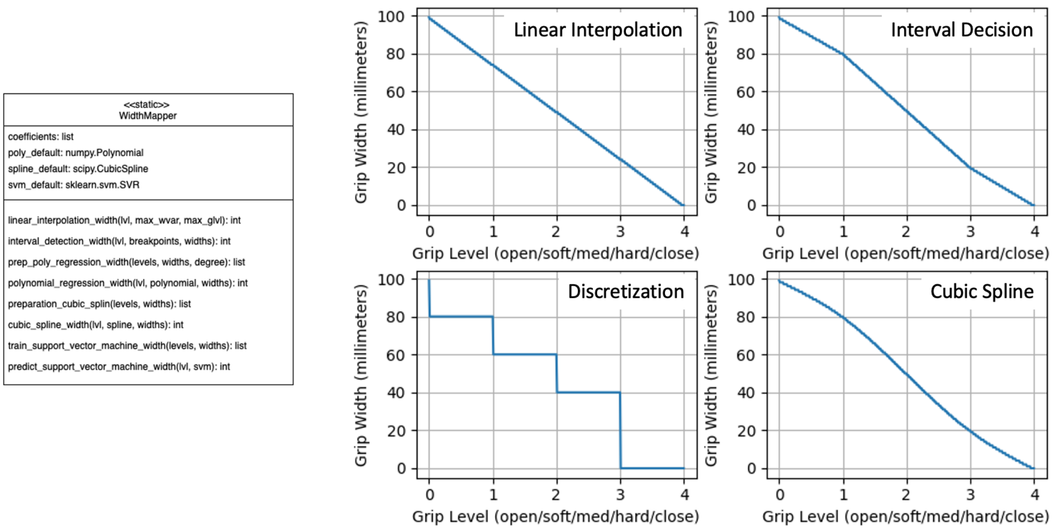


Figure 32: Different models and mappers used for Gripper Width computations.

The **Linear Interpolation** approach assumes a linear relationship between grip level and objective width. This approach is computationally efficient but exhibits limitations in accurately capturing non-linear variations in gripper requirements. Its simplicity makes it suitable for applications where approximate width control is sufficient.

This **Interval Detection** approach divides the grip level range into intervals and applies linear interpolation within each interval. By segmenting the mapping function, it offers greater flexibility in accommodating non-linear relationships between grip level and objective width. However, the accuracy of the model is dependent on the appropriate selection of interval boundaries.

The **Polynomial Regression** approach employs a higher-order polynomial function to approximate the mapping between grip level and objective width. This approach can capture complex non-linear relationships, providing a more accurate representation of the desired width variation. However, it is susceptible to overfitting if the polynomial degree is excessively high.

The **Cubic Spline** approach utilizes piecewise cubic polynomials to interpolate the data points, resulting in a smooth and continuous function. This approach offers a balance between flexibility and computational efficiency. Cubic splines can effectively capture complex patterns in the data while maintaining smoothness and avoiding overfitting.

The **Support Vector Machine (SVM) Regression** approach employs a kernel-based approach to map grip level to objective width. This model is particularly effective in handling non-linear relationships and can achieve high accuracy. However, the computational cost of training and applying the SVM model is generally higher compared to the other methods.

E.3 Models for Gripper Force

Four primary models are explored: linear mapping, exponential decay, piecewise quadratic mapping, and cubic spline interpolation. Each model offers distinct characteristics in terms of force profile, computational efficiency, and adaptability to different scenarios.

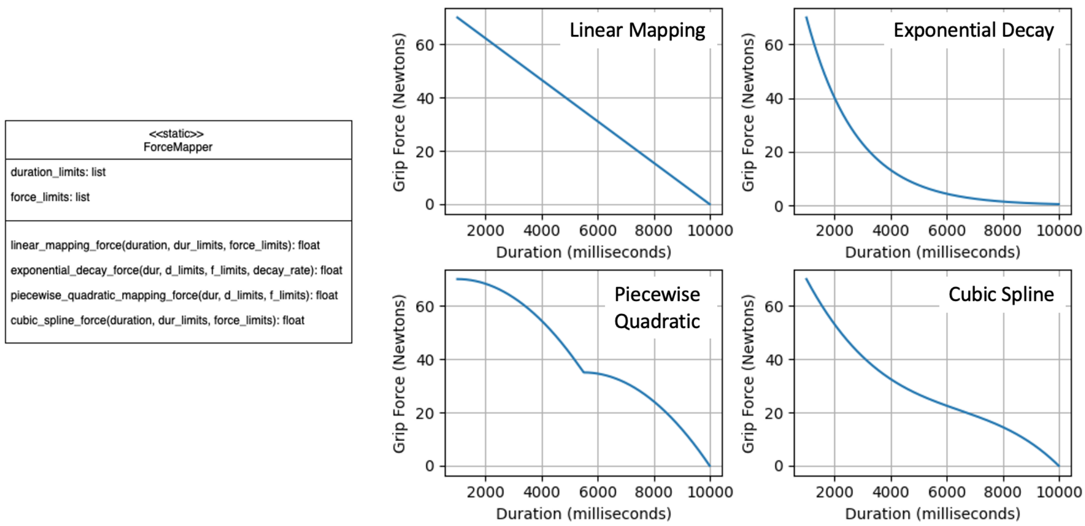


Figure 33: Different models and mappers used for Gripper Force computations.

The **Linear Mapping** approach establishes a direct proportional relationship between interaction duration and grip force. While computationally efficient, this model often oversimplifies the complex dynamics of grip force generation. Its applicability is limited to scenarios where a linear force profile is sufficient and precise force control is not critical.

The **Exponential Decay** approach introduces a more nuanced approach by simulating a gradual decrease in grip force over time. This model can be used to mimic the natural decay of human grip strength and is suitable for applications requiring a controlled release of grasped objects. However, the decay rate parameter needs to be carefully tuned to achieve the desired force profile.

To address the limitations of previous models, the **Piecewise Quadratic** approach divides the interaction duration into segments and applies quadratic functions within each segment. This approach offers greater flexibility in shaping the force profile, allowing for more complex force patterns. However, it requires careful determination of segment boundaries and polynomial coefficients to achieve desired force characteristics.

Lastly, the **Cubic Spline** approach provides a smooth and continuous representation of the grip force profile. By fitting cubic polynomials to the data points, this model can capture intricate force variations while maintaining computational efficiency. Cubic splines offer a balance between flexibility and smoothness, making them suitable for applications requiring precise force control.

E.4 Models for Delay Recovery

This section outlines the various strategies employed to compensate for potential delays in the teleoperation system. These models aim to synchronize robot movements with user inputs, mitigating the negative impact of increased latency on the overall user experience.

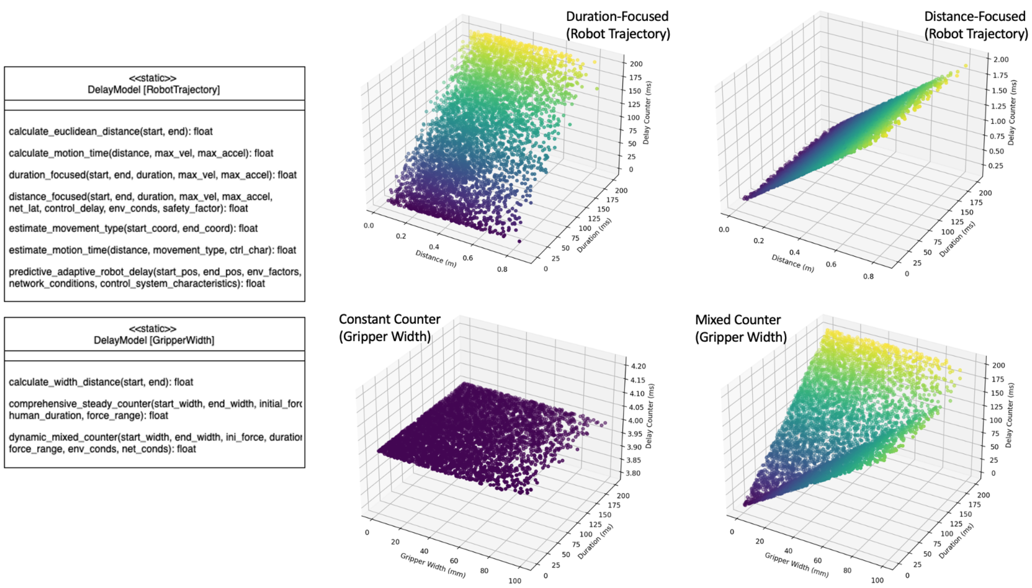


Figure 34: Different models and mappers used for Delay Counter computations.

The **Duration-Focused Model** calculates the robot's motion time based solely on the desired human movement duration. This approach ensures synchronization between human and robot actions but leads to suboptimal performance in scenarios with varying distances or environmental conditions. Its simplicity makes it suitable for applications with predictable and consistent operating conditions.

The **Distance-Focused Model** incorporates additional factors such as distance between start and end positions, maximum velocity, and acceleration to estimate the robot's motion time. By considering these parameters, this model offers a more accurate prediction of the required delay. Furthermore, it incorporates environmental conditions and network latency to enhance adaptability and robustness.

The **Predictive Adaptive Model** introduces a more sophisticated approach by estimating motion time based on movement type, environmental factors, and network conditions. This model employs predictive techniques to anticipate potential delays and adjust the robot's motion accordingly. While computationally more intensive, this approach has the potential to significantly improve system responsiveness and user experience.

Focusing on the gripper, the **Constant Counter Model** establishes a direct relationship between the desired width change and the required motion time. This model assumes a constant gripper velocity and disregards external factors such as force or environmental conditions. Its simplicity makes it computationally efficient but results in suboptimal performance in dynamic scenarios.

Finally, the **Mixed Counter Model** introduces a more complex approach by considering both width change and initial grip force to estimate the motion time. This model offers greater flexibility in adapting to varying gripping conditions. Additionally, it incorporates environmental and network factors to enhance robustness and adaptability.

E.5 Models for Haptic Intensity

This section outlines various basic and advanced models for mapping input parameters (width and force) to haptic intensity. Each model offers distinct characteristics in terms of computational complexity, accuracy, and adaptability to different application scenarios.

The **Linear Interpolation** model establishes a direct linear relationship between input parameters (width and force) and haptic intensity. While computationally efficient, it often falls short in capturing the nuanced complexities of human perception and leads to suboptimal haptic feedback.

The **Weighted Interpolation** Model introduces weighting factors to balance the contributions of width and force to the final haptic intensity. By adjusting the weights, it is possible to prioritize either width or force-based cues, providing greater flexibility in tailoring the haptic feedback to specific applications. However, the choice of appropriate weights remains empirical and requires fine-tuning.

The **Biased Interpolation** Model extends the weighted interpolation approach by incorporating non-linear transformations and dynamic weight adjustments. By applying exponential scaling and threshold-based weight modifications, this model can better capture the perceptual non-linearities of human haptic perception. However, it requires careful parameter tuning and introduces additional computational overhead.

The **Bivariate Interpolation** Model leverages grid-based interpolation techniques to create a smooth mapping surface between width, force, and haptic intensity. By generating a grid of precomputed values and interpolating within this space, this approach can capture complex relationships between the input parameters and output intensity. However, the computational cost and memory requirements of this model is higher compared to simpler methods.

The **Cubic Bezier Interpolation** Model utilizes a cubic Bezier curve to define the mapping function between input parameters and haptic intensity. By carefully selecting

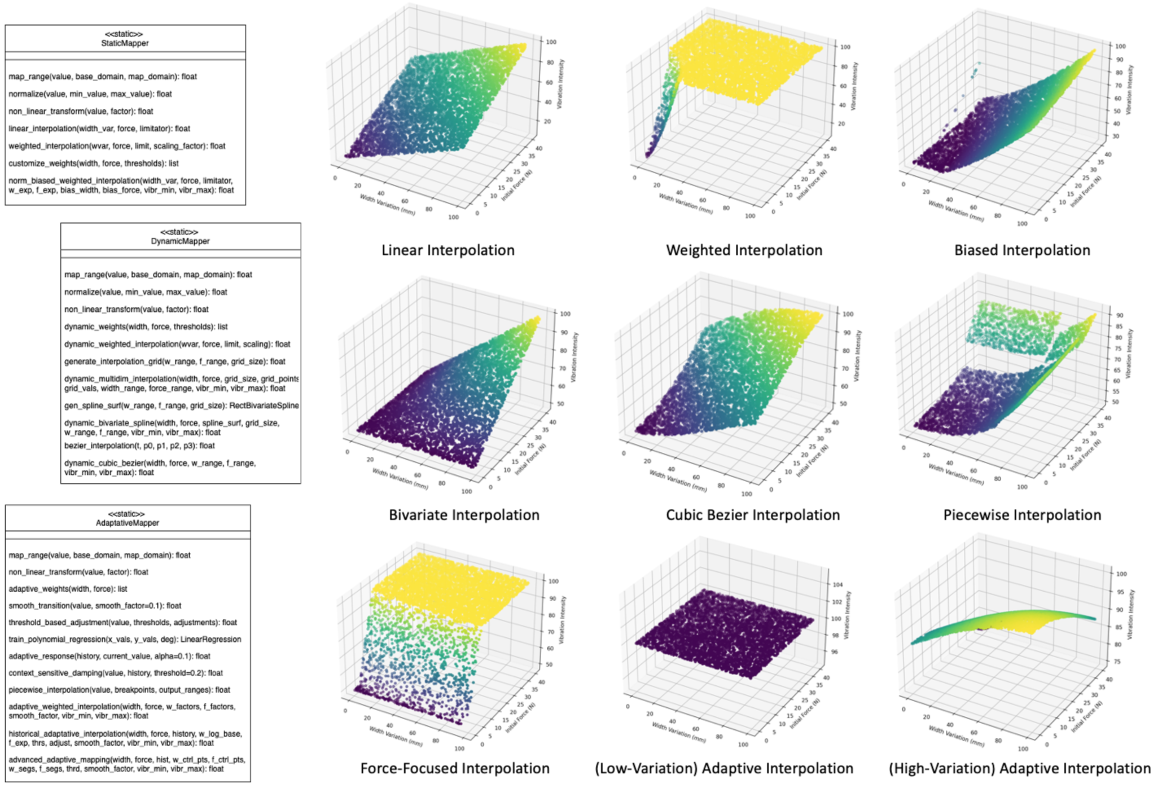


Figure 35: Different models and mappers used for Haptic Intensity computations.

control points, it is possible to create smooth and customizable haptic feedback profiles. However, the effectiveness of this model depends on the appropriate choice of control points, which requires iterative refinement.

The **Piecewise Interpolation** Model divides the input space into segments and applies linear interpolation within each segment. By allowing for different slopes and intercepts in each segment, this approach can capture non-linear relationships between input parameters and haptic intensity. However, careful selection of segment boundaries is crucial to achieve accurate mapping.

The **Force-Focused Interpolation** Model prioritizes force as the primary determinant of haptic intensity, with width playing a secondary role. By applying non-linear transformations and weightings to the force input, this model can emphasize force-related cues in the haptic feedback. However, it limits the ability to convey information related to object size or shape through haptic variations.

The **Adaptive Interpolation** Model incorporates elements of previous models, such as piecewise interpolation, non-linear transformations, and dynamic weight adjustments, to create a more adaptive and flexible mapping function. By combining these techniques, it is possible to achieve a higher degree of customization and responsiveness to changing input conditions. However, the complexity of this model requires careful implementation and parameter tuning.