

FEDERAL UNIVERSITY OF SANTA CATARINA
CAMPUS JOINVILLE
POSTGRADUATE PROGRAM IN MECHANICAL ENGINEERING AND MECHANICAL SCIENCES

MATHEUS NICOLÁS HERNANDEZ

DEVELOPMENT, INTEGRATION, AND COMPARATIVE EVALUATION OF MIXED REALITY AND CONVENTIONAL ROBOTIC TELEOPERATION SYSTEMS: AN INTEGRATED FRAMEWORK FOR PERFORMANCE, WORKLOAD, USABILITY, AND USER EXPERIENCE

JOINVILLE
2026

MATHEUS NICOLÁS HERNANDEZ

DEVELOPMENT, INTEGRATION, AND COMPARATIVE EVALUATION OF MIXED REALITY AND CONVENTIONAL ROBOTIC TELEOPERATION SYSTEMS: AN INTEGRATED FRAMEWORK FOR PERFORMANCE, WORKLOAD, USABILITY, AND USER EXPERIENCE

Dissertation submitted to the Postgraduate Program in Mechanical Engineering and Mechanical Sciences at Federal University of Santa Catarina as part of the requirements for the degree of Master of Science in Mechanical Engineering.
Supervisor: Prof. Roberto Simoni, Ph.D.
Co-supervisor: Douglas Wildgrube Bertol, Ph.D.

JOINVILLE
2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Hernandez, Matheus Nicolás
DEVELOPMENT, INTEGRATION, AND COMPARATIVE EVALUATION OF MIXED REALITY AND CONVENTIONAL ROBOTIC TELEOPERATION SYSTEMS: AN INTEGRATED FRAMEWORK FOR PERFORMANCE, WORKLOAD, USABILITY, AND USER EXPERIEN / Matheus Nicolás Hernandez ; orientador, Roberto Simoni, coorientador, Douglas Wildgrube Bertol, 2026.
158 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Campus Joinville, Programa de Pós-Graduação em Engenharia e Ciências Mecânicas, Joinville, 2026.

Inclui referências.

1. Engenharia e Ciências Mecânicas. 2. teleoperation. 3. mixed reality. 4. human-robot interaction. 5. telerobotics. I. Simoni, Roberto . II. Bertol, Douglas Wildgrube. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia e Ciências Mecânicas. IV. Título.

MATHEUS NICOLÁS HERNANDEZ

DEVELOPMENT, INTEGRATION, AND COMPARATIVE EVALUATION OF MIXED REALITY AND CONVENTIONAL ROBOTIC TELEOPERATION SYSTEMS: AN INTEGRATED FRAMEWORK FOR PERFORMANCE, WORKLOAD, USABILITY, AND USER EXPERIENCE

The present work at the Master's level was evaluated and approved, on August 7, 2026, by an examining committee composed of the following members:

Prof. Med Amine Laribi, Ph.D.
University of Poitiers

Prof. Henrique Simas, Ph.D.
Federal University of Santa Catarina

We certify that this is the **original and final version** of the dissertation approved for the award of the degree of Master of Science in Mechanical Engineering.



Documento assinado digitalmente
Roberto Simoni
Data: 14/08/2026 07:59:57-0300
CPF: ***.398.969-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Roberto Simoni, Ph.D.
Program Coordinator



Documento assinado digitalmente
Roberto Simoni
Data: 14/08/2026 08:00:27-0300
CPF: ***.398.969-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Roberto Simoni, Ph.D.
Supervisor

JOINVILLE, 2026.

I dedicate this work to my wife, Susana,
and to my children, Manuel and the
one yet to come, for always supporting
me in every achievement. I also
dedicate it to my parents, for investing
in my education from an early age.

ACKNOWLEDGEMENTS

I would first like to thank God for giving me health, creativity, and the opportunities necessary to reach each stage of this work. I also thank my wife Susana for encouraging me every day in my studies and making this achievement possible. I also thank my son Manuel for sharing his deserved attention with his father's studies and motivating me to achieve a better future for our family. I also thank the baby who is on the way for bringing much joy in the final stage of this work.

I also thank my parents, Daniel and Simone, for always being my support and for providing me, from an early age, with access to quality schools and all the support necessary for my learning, guiding me in every choice in life.

I thank my advisor Roberto Simoni for providing directions and improvements to this development, and for the invitation to be part of the Graduate Program in Mechanical Engineering and Sciences. I also extend my thanks to my co-advisor Douglas Bertol for the review and suggestions for improvements to the developed platform.

I also thank the SENAI Institute of Innovation in Manufacturing Systems, especially Douglas Negri and Saulo Melloti, for seeing potential and investing in my work, as well as for providing the equipment that made this research possible.

“Happy is the man that findeth wisdom,
and the man that getteth understanding.”
(Proverbs 3:13)

RESUMO EXPANDIDO

INTRODUÇÃO

A crescente utilização de robôs em ambientes industriais tem contribuído para reduzir a exposição de trabalhadores a atividades perigosas, repetitivas e insalubres. Nesse contexto, a teleoperação robótica destaca-se como uma alternativa capaz de permitir que operadores executem tarefas remotamente, mantendo o ser humano afastado de ambientes de risco. Entretanto, os sistemas convencionais de teleoperação normalmente utilizam monitores bidimensionais e dispositivos como joysticks e teclados para controlar manipuladores robóticos, exigindo que o operador interprete um ambiente tridimensional por meio de uma interface bidimensional, o que pode dificultar tarefas de manipulação mais complexas.

Com os avanços das tecnologias imersivas, a Realidade Mista (MR) passou a ser considerada uma alternativa para tornar a interação entre operador e robô mais intuitiva, oferecendo maior percepção espacial nas teleoperações robóticas. Apesar do crescente número de pesquisas, os estudos comparativos entre interfaces convencionais e imersivas para a teleoperação de braços robóticos ainda apresentam pouca convergência metodológica, empregando diferentes métricas, questionários e estratégias de avaliação. Essa diversidade dificulta a comparação entre os estudos e a identificação das reais vantagens e limitações das interfaces imersivas.

Diante desse cenário, esta dissertação apresenta o desenvolvimento, a integração e a avaliação comparativa de dois sistemas de teleoperação robótica aplicados ao manipulador industrial FANUC LR Mate 200iD/7L: uma interface convencional e uma interface imersiva baseada em MR. Além do desenvolvimento dos sistemas, a dissertação propõe uma metodologia integrada de avaliação, que combina métricas objetivas de desempenho com instrumentos padronizados para avaliação da carga de trabalho, da usabilidade e da experiência do usuário.

OBJETIVOS

O objetivo geral desta dissertação foi avaliar comparativamente interfaces de teleoperação robótica baseadas em MR e convencionais quanto ao desempenho operacional, à carga cognitiva, à usabilidade e à experiência do usuário no contexto de ambientes de risco.

Para atingir esse objetivo, foram definidos como objetivos específicos: desenvolver e implementar uma interface de teleoperação baseada em MR e uma interface convencional para o controle de um manipulador robótico industrial; comparar o desempenho operacional das duas abordagens por meio de métricas quantitativas de execução das tarefas; avaliar a carga cognitiva, a usabilidade e a experiência do usuário utilizando métodos qualitativos de avaliação; e analisar o desempenho dos operadores durante a realização de tarefas repetidas de teleoperação utilizando ambas as interfaces.

METODOLOGIA

A pesquisa foi conduzida por meio do desenvolvimento de dois sistemas de teleoperação robótica baseados em uma arquitetura mestre-escravo. Ambos os sistemas foram

desenvolvidos utilizando o Robot Operating System (ROS) como plataforma de comunicação e controle do manipulador industrial FANUC LR Mate 200iD/7L. A interface convencional foi implementada por meio do MoveIt/RViz, enquanto a interface imersiva foi desenvolvida em Unity e utilizada com o dispositivo Meta Quest 3. O sistema experimental também contou com uma garra robótica, câmeras de monitoramento e um sensor de distância a laser.
Foram implementadas funcionalidades para ambos os sistemas, incluindo modos de controle manual e automático, integração da garra, mecanismos de segurança e prevenção de colisões, sincronização dos modelos robóticos, além de recursos de percepção e monitoramento por meio de câmeras e sensor de distância.
Para validar os sistemas foi realizado um estudo experimental envolvendo 14 participantes submetidos à execução da mesma tarefa de telemanipulação utilizando ambas as interfaces. Durante os experimentos foram coletadas métricas objetivas, incluindo tempo de execução, número de tentativas de preensão, colisões ao longo das repetições. Paralelamente, foram aplicados os questionários NASA-TLX para avaliação da carga cognitiva, SUS para avaliação da usabilidade e UEQ-S para avaliação da experiência do usuário. Os dados foram analisados estatisticamente e permitiram comparar quantitativa e qualitativamente as duas abordagens de teleoperação.

RESULTADOS OBTIDOS

Os resultados experimentais demonstraram que a interface de teleoperação baseada em MR possibilitou tempos de execução significativamente menores em comparação com a interface convencional. Em contrapartida, a interface imersiva apresentou maior número de tentativas de preensão, enquanto não foram observadas diferenças estatisticamente significativas na ocorrência de colisões entre as duas abordagens. Em relação ao comportamento de aprendizagem, também não foram identificadas diferenças estatisticamente significativas entre as interfaces, embora eventos operacionais, como liberações involuntárias do objeto e dificuldades de alinhamento, tenham influenciado parte da variabilidade observada durante as repetições.
Na avaliação qualitativa, os questionários NASA-TLX e SUS indicaram que as interfaces apresentaram percepções semelhantes quanto à carga de trabalho e à usabilidade, sem diferenças estatisticamente significativas. Por outro lado, o UEQ-S revelou diferenças expressivas nos aspectos hedônicos da experiência do usuário, com a interface em MR apresentando pontuações significativamente superiores nessa dimensão, indicando maior envolvimento, estímulo e qualidade experiencial durante a interação. Esses resultados sugerem que a principal vantagem subjetiva da interface imersiva está relacionada à experiência proporcionada pela interação, e não necessariamente à redução da carga cognitiva ou ao aumento da usabilidade.
Os resultados também evidenciaram que a interface em MR foi percebida pela maioria dos participantes como mais intuitiva para movimentos amplos do manipulador e para a visualização tridimensional do ambiente remoto. Entretanto, durante tarefas que exigiam posicionamento fino, alguns participantes preferiram a interface convencional, devido à possibilidade de realizar ajustes segmentados por eixo, proporcionando maior precisão em movimentos delicados.
Além da comparação entre as interfaces, esta dissertação contribuiu metodologicamente ao propor uma abordagem integrada de avaliação para estudos comparativos em teleoperação de braços robóticos, combinando métricas objetivas de desempenho com instrumentos padronizados para avaliação da carga cognitiva, da usabilidade e

da experiência do usuário. Os resultados mostraram que essas dimensões são complementares e não intercambiáveis, uma vez que diferenças na experiência do usuário não foram necessariamente acompanhadas por diferenças na carga cognitiva, na usabilidade ou no desempenho operacional. Assim, o estudo reforça a importância de estratégias de avaliação multidimensionais para uma compreensão mais abrangente das interfaces de teleoperação robótica.

Palavras-chave: teleoperação; telerrobótica; realidade mista; interação humano-robô; usabilidade; experiência do usuário.

ABSTRACT

This dissertation presents the development, integration, and comparative evaluation of two robotic teleoperation systems applied to the control of a FANUC LR Mate 200iD/7L industrial manipulator: a conventional system based on the 2D MoveIt/RViz interface and an immersive system built in Mixed Reality (MR) using Unity, Meta Quest 3, and ROS. The motivation for this study is associated with the need to reduce human exposure to high-risk tasks through robotic teleoperation, seeking approaches that provide greater safety, efficiency, and spatial perception during teleoperation. Although the literature presents recent advances in immersive teleoperation, there is still a gap in experimental studies that comparatively evaluate operational performance, usability, cognitive workload, and user experience between MR approaches and conventional methods in an integrated manner. The developed systems were experimentally evaluated through a telemanipulation task using quantitative performance metrics and qualitative assessment instruments. The results indicated that the immersive interface presented advantages related to execution speed and participants' subjective experience, especially regarding hedonic and experiential aspects of the interaction, although limitations related to fine manipulation precision were also identified. As a contribution, this dissertation proposes an integrated comparative evaluation approach applied to the analysis of immersive and conventional robotic teleoperation interfaces.

Keywords: teleoperation; telerobotics; mixed reality; human-robot interaction; usability; user experience.

LIST OF FIGURES

Figure 1 – Worldwide operational stock of industrial robots 23
Figure 2 – Overview of a Master-Slave telerobotic system 28
Figure 3 – Main historical milestones of telerobotics 29
Figure 4 – Control architectures: Direct, Shared, and Supervisory 30
Figure 5 – Adapted representation of the continuum concept 33
Figure 6 – Number of comparative studies found in the Scopus database using the search string from Table 1 39
Figure 7 – Number of comparative studies identified in the Scopus database using the search string from Table 2 41
Figure 8 – Schematic overview of the refinement of the identified studies 41
Figure 9 – FANUC LR Mate 200iD/7L robot and R-30iB Mate controller 49
Figure 10 – MyGripper F100 50
Figure 11 – Camera attached to the gripper 51
Figure 12 – Fixed camera positioned for workspace visualization 52
Figure 13 – Arduino Uno and TOF10120 sensor installed on the robot 53
Figure 14 – Meta Quest 3 HMD 54
Figure 15 – Unity logo 55
Figure 16 – Meta Quest Link logo 56
Figure 17 – ROS and Noetic logos 56
Figure 18 – ROS-Industrial logo 57
Figure 19 – Master-slave communication and command architecture of the immersive teleoperation system 59
Figure 20 – Physical setup of the immersive teleoperation system 60
Figure 21 – Master-slave communication and command architecture of the conventional teleoperation system 61
Figure 22 – Physical setup of the conventional teleoperation system 62
Figure 23 – Comparison between the original and modified gripper system architectures 65
Figure 24 – Comparison between the original and updated robot and gripper models 68
Figure 25 – Comparison between the robotic representations in the ROS and Unity environments 69
Figure 26 – Configuration parameters of the `RevoluteRobot` component in Unity, used in the inverse kinematics calculation of the target robot 71
Figure 27 – Standard MoveIt planning and execution buttons 73
Figure 28 – Color change of the target robot in the Unity environment during collision detection 75
Figure 29 – Example of collision detection by MoveIt in the RViz interface 76

Figure 30 – Comparison between the camera visualizations in the conventional and immersive modes 78
Figure 31 – Comparison between the laser distance sensor representations in the ROS and MR environments 79
Figure 32 – Conventional environment interface 81
Figure 33 – Immersive environment interface 83
Figure 34 – Cable routing and peripheral integration 85
Figure 35 – Telemanipulation Task 90
Figure 36 – Summary of the participants' profile in the study 97
Figure 37 – Operational performance metrics 99
Figure 38 – Best execution time achieved by each participant 100
Figure 39 – Individual execution-time learning slopes 101
Figure 40 – Individual grasp-action learning slopes 102
Figure 41 – NASA-TLX dimension scores. 104
Figure 42 – SUS scores 105
Figure 43 – UEQ-S scores 106
Figure A.1 – Technical Specifications of the FANUC LR Mate 200iD/7L 136
Figure B.1 – Technical Specifications of the MyGripper F100 137
Figure C.1 – Packages installed in Unity 139
Figure C.2 – ROS–TCP Connector configuration 140
Figure C.3 – Workcell configuration in Roboguide 142
Figure C.4 – Files imported and compiled in Roboguide 143
Figure C.5 – List of installed programs on the controller verified in the *Teach Pendant* 143
Figure C.6 – Example of a *Server Tag* configuration in the controller 144
Figure C.7 – Default configuration of the *ros_relay* program 145
Figure C.8 – Default configuration of the *ros_state* program 145
Figure C.9 – Configured *ros_movesm* program in the *Teach Pendant* 146
Figure D.1 – Teach Pendant screen with access to the Host Comm menu 147
Figure D.2 – Server configuration on the Teach Pendant 148
Figure D.3 – Definition and startup of tag "S5" on the Teach Pendant 148
Figure D.4 – Verification of tag "S5" on the Teach Pendant 149
Figure D.5 – Selection and execution of the "ROS" program on the Teach Pendant 149
Figure D.6 – Green controller button indicating active operation 150
Figure D.7 – Screen indicating ROS program operation 150
Figure F.1 – Camera configuration in the Unity "Stream Receiver" script 157

LIST OF TABLES

Table 1 – Search string used in the general review 39
Table 2 – Search string for comparative studies 40
Table 3 – Synthesis of comparative studies between conventional and immersive interfaces 42
Table 4 – Predominant user experience dimensions addressed by the subjective instruments identified in the reviewed studies 44
Table 5 – Search string used to identify studies that simultaneously apply SUS, NASA-TLX, and UEQ 46
Table 6 – Comparison between interaction mechanisms in the conventional and immersive modes 84
Table 7 – Joint limits of the FANUC LR Mate 200iD/7L 86
Table 8 – Definitions of NASA-RTLX dimensions 93
Table 9 – System Usability Scale (SUS) statements 94
Table 10 – Dimensions and items of the UEQ-S questionnaire 95
Table 11 – Summary of quantitative performance results. 99
Table 12 – Execution time statistics across repeated trials (s). 101
Table 13 – Summary of individual execution-time learning slope analysis. 101
Table 14 – Summary of individual grasp-action learning slope analysis. 102
Table 15 – Summary of NASA-TLX results. 104
Table 16 – Summary of SUS results. 105
Table 17 – Summary of UEQ-S results. 106
Table 18 – Main findings and limitations identified in the reviewed studies 113
Table 19 – Methodological patterns and recurring limitations identified in the reviewed studies 114

LIST OF ABBREVIATIONS AND ACRONYMS

AR	Augmented Reality
AV	Augmented Virtuality
CEP	*Comitê de Ética em Pesquisa* (Research Ethics Committee)
CNS	*Conselho Nacional de Saúde* (National Health Council)
EoU	Ease-of-Use
GTQ	Gas Tank Questionnaire
GUI	Graphical User Interface
HMDs	Head-mounted displays
HRE	Human Risk Exposure
IFR	International Federation of Robotics
ILO	International Labour Organization
ISI-SM	SENAI Institute for Innovation in Manufacturing Systems
LTS	Long Term Support
MEC-SPQ	MEC Spatial Presence Questionnaire
MPT	*Ministério Público do Trabalho* (Brazilian Labor Prosecution Office)
MR	Mixed Reality
NASA	National Aeronautics and Space Administration
NASA-TLX	NASA Task Load Index
RIA	Robot Institute of America
ROS	Robot Operating System
ROVs	Remotely Operated Vehicles
SUS	System Usability Scale
TAM	Technology Acceptance Model
TCLE	*Termo de Consentimento Livre e Esclarecido* (Informed Consent Form)
TP	Teach Pendant
UEQ	User Experience Questionnaire
UEQ-S	User Experience Questionnaire - Short
UFSC	Federal University of Santa Catarina
URDF	Unified Robot Description Format
UX	User Experience
XR	Extended Reality

CONTENTS

1 INTRODUCTION 19
1.1 HYPOTHESES 20
1.2 OBJECTIVES 21
1.2.1 General Objective 21
1.2.2 Specific Objectives 21
2 THEORETICAL BACKGROUND 22
2.1 ROBOTIC APPROACHES AIMED AT REDUCING HUMAN EXPOSURE 23
2.1.1 Industrial Robotics 24
2.1.2 Autonomous Robotics 25
2.1.3 Telerobotics 26
2.1.4 State-of-the-Art in Immersive Teleoperation 37
3 HARDWARE AND SOFTWARE INFRASTRUCTURE 48
3.1 HARDWARE 48
3.1.1 FANUC LR Mate 200iD/7L Robot and R-30iB Mate Controller . . . 48
3.1.2 Gripper: MyGripper F100 49
3.1.3 Monitoring Cameras 50
3.1.4 Arduino and Laser Sensor 52
3.1.5 Head-Mounted Display: Meta Quest 3 53
3.1.6 Computers 54
3.2 SOFTWARE 55
3.2.1 Unity 55
3.2.2 Meta Quest Link 55
3.2.3 ROS 56
3.2.4 Roboguide 57
4 DEVELOPMENT OF THE ROBOTIC TELEOPERATION SYSTEMS 58
4.1 IMMERSIVE TELEOPERATION SYSTEM ARCHITECTURE 58
4.1.1 Integration between systems 60
4.2 CONVENTIONAL TELEOPERATION SYSTEM ARCHITECTURE . . 61
4.3 INTEGRATION AND IMPLEMENTATION OF SYSTEM FUNCTIONALITIES 62
4.3.1 Integration of the Gripper into the Teleoperation System 63
4.3.2 Update of Graphical Models and Collision Structures 66
4.3.3 Structure and Integration of Virtual Robotic Models 68
4.3.4 Control modes 73
4.3.5 Safety Mechanisms and Collision Prevention 74
4.3.6 Perception and Sensory Feedback 76
4.3.7 Operator interaction 80

4.3.8 Communication Latency . . . 84
4.3.9 Movement Limits and Mechanical Adjustments . . . 85
4.3.10 Availability and Reproducibility . . . 86
5 COMPARATIVE STUDY . . . 87
5.1 STUDY CHARACTERIZATION . . . 87
5.2 OBJECTIVE OF THE EXPERIMENTAL STUDY . . . 87
5.3 PARTICIPANTS . . . 87
5.3.1 Selection Criteria . . . 88
5.4 DEFINITION OF THE TELEMANIPULATION TASK . . . 88
5.5 EXPERIMENTAL PROCEDURE . . . 89
5.6 EVALUATION METRICS . . . 91
5.6.1 Quantitative Evaluation . . . 91
5.6.2 Qualitative Evaluation . . . 92
5.6.3 Statistical Analysis . . . 95
5.7 ETHICAL CONSIDERATIONS . . . 96
6 RESULTS . . . 97
6.1 PARTICIPANT PROFILE . . . 97
6.2 QUANTITATIVE RESULTS . . . 98
6.2.1 Operational Performance . . . 98
6.2.2 Learning Behavior . . . 100
6.3 QUALITATIVE RESULTS . . . 103
6.3.1 NASA-TLX . . . 103
6.3.2 SUS . . . 105
6.3.3 UEQ-S . . . 106
7 DISCUSSION . . . 108
7.1 OPERATIONAL PERFORMANCE AND LEARNING BEHAVIOR . . . 108
7.2 WORKLOAD, USABILITY, AND USER EXPERIENCE . . . 110
7.3 METHODOLOGICAL CONTRIBUTIONS AND RESEARCH GAP . . . 112
7.3.1 Synthesis of Findings from the Comparative Literature . . . 112
7.3.2 Methodological Patterns Observed in the Literature . . . 114
7.3.3 Validation of the Research Gap . . . 115
7.3.4 Methodological Contribution of this Dissertation . . . 116
8 FINAL CONSIDERATIONS . . . 117
8.1 MAIN FINDINGS . . . 117
8.2 RESEARCH OBJECTIVES AND HYPOTHESES . . . 118
8.3 LIMITATIONS . . . 119
8.4 FUTURE WORK . . . 119
References . . . 121

APPENDIX A – TECHNICAL SPECIFICATIONS OF THE FANUC LR MATE 200ID/7L 136
APPENDIX B – TECHNICAL SPECIFICATIONS OF THE MYGRIPPER F100 137
APPENDIX C – INTEGRATION BETWEEN SYSTEMS 138
APPENDIX D – FANUC ROBOT INITIALIZATION PROCEDURE FOR ROS CONNECTION 147
APPENDIX E – ROS ENVIRONMENT INITIALIZATION PROCEDURE151
APPENDIX F – UNITY ENVIRONMENT INITIALIZATION PROCEDURE 156

1 INTRODUCTION

Technological evolution, particularly in recent decades, has reshaped industrial environments through the increasing adoption of automated systems and robotic technologies, transforming production processes and changing the way occupational safety is addressed. As machines and robots increasingly assume hazardous, unhealthy, or repetitive tasks, human exposure to risk can be reduced, although many industrial sectors still expose workers to dangerous conditions, resulting in workplace accidents and occupational diseases.

In this context, robotic and technological alternatives emerge to mitigate risks and optimize high-risk operations, reducing human exposure to dangerous tasks. Studies, such as Gihleb *et al.* (2022), demonstrate that the use of robots in industries can significantly reduce the incidence of accidents, as evidenced by data from the United States and Germany, where the increase in robot usage is associated with a substantial decrease in workplace accidents.

Some robotic solutions have already been implemented to remove workers from high-risk situations, such as the RDC robot (Negri *et al.*, 2023), which was designed to perform cleaning, painting, and inspection of flat surfaces of hulls on offshore oil platforms, where tests conducted in relevant environments demonstrated high efficiency in resource savings, reduction in process time, and a significant decrease in worker exposure to risk.

Another robotic solution was proposed by Gonçalves *et al.* (2022), aiming at the removal of debris from high-voltage lines, and this approach aims to reduce the costs associated with maintenance operations that depend on human labor, significantly increase task efficiency, and mitigate risks to worker safety by reducing the need for direct contact with power lines.

In the context of high-risk activities, teleoperation stands out as an effective solution (Shukla; Karki, 2016), allowing tasks that previously required direct human intervention to be performed through robotic systems operated remotely. Teleoperation enables a human operator to execute tasks by controlling a machine from a distance (Su, Y.-P. *et al.*, 2023a; Tavakkoli; Wilson; Bounds, 2020), reducing, or even eliminating, human exposure to hazardous environments and high-risk operations.

In its conventional application, teleoperation typically relies on monitors for visualization and on devices such as keyboards or joysticks for control, requiring operators to interact with a three-dimensional environment through a two-dimensional interface, a limitation that can make complex manipulation tasks more difficult and, in some cases, impractical (Whitney *et al.*, 2020; Darvish *et al.*, 2023).

However, remote machine control fails to meet the intuitive aspect of human interaction, as it does not bring teleoperation closer to reality (Triantafyllidis *et al.*, 2020),

and, as a way to reduce this gap between the operator and the remote environment, there has been a growing movement toward the implementation of Mixed Reality (MR) in these operations (Wang, X., 2011).

Although the use of MR in teleoperation applications has advanced significantly in recent years, important challenges remain, including latency in robot responses (Su, Y. *et al.*, 2023), accuracy in robotic control (Esaki; Sekiyama, 2024), acceptance of the technology among different user profiles (Nanda; Maddali; Lazar, A., 2022), and cognitive fatigue caused by prolonged periods of immersion (Shao; Zhou; Liu, Z., 2019). Furthermore, the literature still provides limited evidence regarding the effectiveness of these systems in complex operational environments, particularly when compared with conventional teleoperation approaches, making additional investigations necessary to evaluate not only technical performance, but also usability, user acceptance, and efficiency during task execution.

Considering these challenges, the following research question is proposed: to what extent does the integration of MR into robotic teleoperation systems influence operational performance, usability, and operator perception when compared to conventional control methods in high-risk environments?

1.1 HYPOTHESES

Based on the proposed research question, the following hypotheses were formulated:

- **Hypothesis 1**: MR-based teleoperation will be perceived in a superior way when compared to conventional teleoperation methods, particularly regarding interaction experience and ease of control.
- **Hypothesis 2**: The MR interface will provide a faster learning curve, reducing the time required for operators to achieve stable performance in teleoperation tasks.
- **Hypothesis 3**: The use of MR will provide superior operational performance when compared to conventional teleoperation methods, including lower execution time, fewer collisions, and reduced grasping attempts during task execution.
- **Hypothesis 4**: The use of MR will provide more favorable subjective evaluations regarding cognitive workload, usability, and user experience when compared to conventional teleoperation methods.

The contribution of this study lies in the comparative analysis between MR-based and conventional teleoperation interfaces applied to the control of an industrial robotic arm. Through practical experiments and the collection of quantitative and qualitative

data, this work investigates differences between both approaches regarding operational performance, cognitive workload, usability, and user experience.

Additionally, this dissertation proposes an integrated evaluation framework capable of simultaneously assessing operational, cognitive, usability, and experiential dimensions in robotic teleoperation. This framework was designed to address the gap identified in the literature regarding the limited adoption of holistic evaluation approaches for immersive teleoperation systems.

1.2 OBJECTIVES

This section presents the general objective and the specific objectives that guide this study.

1.2.1 General Objective

Evaluation of MR-based and conventional robotic teleoperation interfaces in terms of operational performance, cognitive workload, usability, and user experience within high-risk environments.

1.2.2 Specific Objectives

The following specific objectives were defined to achieve the general objective:

- Development and implementation of an MR-based and a conventional teleoperation interface for the control of an industrial robotic arm.
- Comparison of immersive and conventional teleoperation interfaces regarding operational performance through quantitative task execution metrics.
- Evaluation of cognitive workload, usability, and user experience associated with immersive and conventional teleoperation interfaces through qualitative evaluation methods.
- Analysis of operator performance during repeated teleoperation tasks using immersive and conventional interfaces.

2 THEORETICAL BACKGROUND

With the advancement of industrial revolutions, human labor has remained fundamental to economic and productive development, while the expansion of industrial activities has also intensified concerns regarding occupational health and safety. In many sectors, workers continue to be exposed to hazardous and unhealthy environments, conditions that contribute to workplace accidents and occupational diseases.

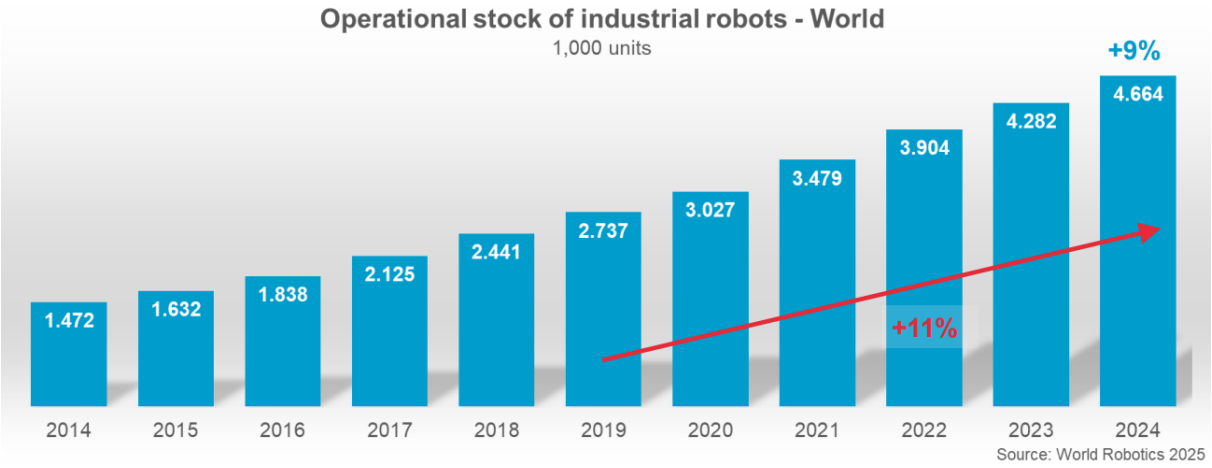
Between 2012 and 2024, according to data from the Digital Observatory of Occupational Health and Safety, launched by the *Ministério Público do Trabalho* (Brazilian Labor Prosecution Office) (MPT) and the International Labour Organization (ILO), 8,824,286 accidents were reported in Brazil, of which 31,981 resulted in worker fatalities. The notification rate is one every 46 seconds, while the frequency of deaths is approximately one every 4 hours (MPT; OIT, 2024).

In response to this reality, worker safety have become a growing concern for companies, reflecting an increasing appreciation of people over material assets, and this shift in focus has driven actions to improve working conditions in high-risk environments, whether through the use of safety equipment or the development of robotic solutions (Berns *et al.*, 2024).

Among the various accident prevention strategies, the reduction of human exposure to risk, commonly referred to as Human Risk Exposure (HRE), stands out as an important approach, since this indicator is directly related to the number of injuries and employee absences (Negri *et al.*, 2023). In this context, robotics has increasingly been employed to reduce HRE by transferring hazardous activities from human operators to automated systems.

The growing adoption of robotic systems in industry can be illustrated by recent data reported by the International Federation of Robotics (IFR). In 2025, the worldwide operational stock reached 4,664,000 industrial robots (Figure 1), an increase of 9% compared with the previous year, while annual installations surpassed half a million units for the fourth consecutive year. In 2024 alone, 542,000 robots were installed, most of them in Asia (74%), followed by Europe (16%) and the Americas (9%) (INTERNATIONAL FEDERATION OF ROBOTICS, 2025).

Figure 1 – Worldwide operational stock of industrial robots



Source: (INTERNATIONAL FEDERATION OF ROBOTICS, 2025).

The relationship between robotics and occupational safety has also been investigated in the literature. In a study conducted by Gihleb *et al.* (2022), which analyzed the adoption of industrial robots and workplace accident rates in the United States and Germany, an increase in the number of robots per worker was associated with a significant reduction in recorded accidents.

These findings highlight the contribution of robotics to risk mitigation in industrial environments, particularly through the automation of repetitive, hazardous, or ergonomically demanding tasks, reducing human exposure to critical conditions and contributing to lower accident and absenteeism rates. In this sense, the adoption of robotic systems is associated with productivity gains, while also contributing to improvements in occupational health and safety.

As robotics became more widespread in industry, different robotic platforms and applications were developed to address the requirements of specific sectors and operational scenarios, expanding the role of automation beyond manufacturing efficiency and strengthening its contribution to worker safety.

2.1 ROBOTIC APPROACHES AIMED AT REDUCING HUMAN EXPOSURE

Robotics is widely used in industrial and operational environments, particularly in applications involving hazardous, unhealthy, or repetitive tasks, where reducing human exposure is often a primary concern. One of the main advantages of robotic systems is the reduction of occupational risks, as they can perform activities that would otherwise expose workers to potentially dangerous conditions, improving safety and reducing direct human involvement in such environments.

To meet the specific demands of different sectors and risk scenarios, robotics has evolved in multiple directions, forming distinct areas of application and research. Although there are several fields of study within robotics (Siciliano; Khatib, 2016), three areas stand out when the focus is on minimizing human exposure to dangerous situations and improving operator safety: Industrial Robotics, Autonomous Robotics, and Telerobotics.

Each of these areas adopts its own operational model, with significant variations regarding the level of system autonomy and the direct involvement of the human operator. The analysis of these approaches makes it possible to understand how different control architectures and interaction strategies impact operational safety, and is especially important in applications involving hostile environments, repetitive tasks, or risks to workers' physical safety.

Although these three categories (industrial robotics, autonomous robotics, and telerobotics) present distinct approaches, their boundaries are not rigid. In many modern systems, there is a convergence between these paradigms, with industrial robots incorporating autonomous functionalities or operating under real-time remote supervision. This application flexibility also means that the same robot model can be adapted to different operational contexts, depending on the control architecture and the environment in which it is deployed.

A notable example of this versatility is the UR5 robotic manipulator, widely used in various scenarios. In the study by Parham M. Kebria *et al.* (2019), the UR5 was employed in a teleoperation application with variable delay, characterizing a telerobotic system. In Pollák *et al.* (2020), the same robot was evaluated in terms of accuracy and repeatability in a programmed motion task, within a typical industrial robotics context. Similarly, Islam, Dias, and Sunda-Meya (2021) used the UR5 coupled with an autonomous mobile base with image-based visual tracking for manipulation tasks, representing an application in the field of autonomous robotics.

2.1.1 Industrial Robotics

Industrial robotics is characterized by the use of automated manipulators to perform repetitive, programmed, and high-precision tasks in controlled environments, such as assembly lines, welding processes, painting, inspection, and packaging. This field has been fundamental in the modernization of manufacturing systems, especially in the context of Industry 4.0, which aims to integrate intelligence, connectivity, and advanced automation into production processes (Siciliano; Khatib, 2016).

According to the classical definition of the Robot Institute of America (RIA), an industrial robot is:

"a reprogrammable multifunctional manipulator designed to move materials, parts,

> *tools or specialized devices through variable programmed motions for the performance of a variety of tasks."* (Siciliano *et al.*, 2009).

Traditionally, industrial robots operate under a pre-programmed automation logic, in which their actions are defined by prior instructions, executed in structured and predictable environments, without the need for real-time adaptation. This approach prioritizes predictability and repeatability, essential characteristics for ensuring high productivity and consistency in industrial processes.

However, as production demands have become more flexible and work environments more dynamic, limitations of traditional industrial robotics have emerged. The need for greater interaction between humans and machines, as well as the demand for safer and more collaborative industrial environments, has driven the development of collaborative robots, known as cobots.

Cobots differ from classical industrial robots by allowing direct and safe interaction with human operators (Colgate; Wannasuphoprasit; Peshkin, 1996). This integration enables humans and robots to work side by side, combining the precision and efficiency of automation with the adaptability and cognitive abilities of workers (Villani *et al.*, 2018). In 2024, more than 64,500 collaborative robots were installed worldwide, representing nearly 12% of all new industrial robot installations and reflecting a 13% increase compared with the previous year (International Federation of Robotics, 2025).

In this context, while traditional industrial robotics remains predominantly associated with repetitive operations in structured environments, cobots were developed to enable safe human–robot collaboration, allowing their use in applications where flexibility and direct interaction with operators are required.

2.1.2 Autonomous Robotics

Autonomous robotics is an area dedicated to the development of systems capable of operating independently, performing complex tasks without the need for continuous human control. These robots have the ability to perceive the environment, interpret sensory data, make decisions, plan trajectories, and adapt their behavior to dynamic and uncertain situations (Fahimi, 2009; Siegwart; Nourbakhsh; Scaramuzza, 2011).

According to Fahimi (2009), autonomous robots combine advanced modeling techniques, motion planning, and control to navigate and operate in partially unknown or constantly changing environments. This autonomy is enabled by the integration of sensors, perception algorithms, artificial intelligence, and real-time control systems, allowing the robot to react to obstacles and adjust its plans as needed to achieve its objectives.

Additionally, Siegwart, Nourbakhsh, and Scaramuzza (2011) define autonomous mobile robots as agents capable of navigating safely and efficiently in unstructured

environments, using mapping, localization, and trajectory planning strategies, as well as possessing capabilities for real-time decision-making. They emphasize that robotic autonomy goes beyond the simple execution of pre-programmed commands, requiring intelligence to deal with uncertainties and the complexity of the real world.

One of the central aspects of autonomous robotics is trajectory planning, which allows the robot to determine safe and efficient routes considering environmental constraints and the physical characteristics of the system (Fahimi, 2009). In addition, advanced adaptive control techniques ensure that the robot maintains robustness and accuracy in its actions in the presence of uncertainties and disturbances.

This field represents a significant advancement over traditional robotics, expanding the scope of applications to unstructured environments and tasks that require high flexibility, such as autonomous vehicles, drones, exploratory robots, and mobile service systems.

2.1.3 Telerobotics

Telerobotics is one of the earliest manifestations of robotics and can be understood, in general terms, as the remote control of robotic systems by a human operator, regardless of the level of autonomy of the robot (Niemeyer *et al.*, 2016; Council *et al.*, 1994).

The term “Telerobotics” derives from the Greek “tele”, meaning “at a distance”, and refers to the physical or conceptual separation between the operator and the operating environment, often imposed by geographical barriers, scale differences, or hazardous environments. In these systems, the brain (human) is separated from the body (robot), and the human-machine interface acts as a bridge between these two elements (Niemeyer *et al.*, 2016).

The central objective of telerobotics is to accurately reproduce the operator’s actions in a remote environment, leveraging human dexterity and judgment for tasks that cannot yet be fully automated (Ferre *et al.*, 2007; Solanes *et al.*, 2022). This is especially applicable in scenarios where full autonomy is unfeasible, whether due to task complexity, environmental variability, or the need for subjective judgment (Solanes *et al.*, 2022; Council *et al.*, 1994). In these situations, teleoperation can be seen as an intermediate solution between industrial robotics and autonomous systems, combining the strength and precision of robots with environmental perception and human decision-making capabilities (Su, Y.-P. *et al.*, 2023b).

Telerobots (robots controlled remotely by human operators regardless of the level of autonomy of the robot) are widely used in contexts where direct human presence is risky, uncomfortable, or impractical, such as underwater operations (Muscolo *et al.*, 2021), space exploration (Stotko *et al.*, 2019; Choi *et al.*, 2017), remote surgeries (Avgousti *et al.*, 2016), explosive disposal, nuclear environments, mining, and

other critical applications (Council *et al.*, 1994; Sheridan, Thomas B, 1989). In these cases, the operator perceives the remote environment through sensory interfaces and acts by sending commands to the robot, in an interaction that can range from direct control to high-level supervision (Ferre *et al.*, 2007).

Within telerobotics, two related terms are frequently used: teleoperation and telemanipulation. Although sometimes used as synonyms, they present conceptual differences. Teleoperation emphasizes task-level operations, focusing on the remote execution of functional actions. Telemanipulation, on the other hand, emphasizes object-level manipulation, focusing on direct interaction with physical elements (Niemeyer *et al.*, 2016).

Another concept widely used to describe the communication architecture in telerobotic systems is the Master-Slave System (Figure 2). In this arrangement, the Master System corresponds to the station where the human operator is physically located and is responsible for generating and transmitting control commands. The Slave System, in turn, refers to the robot that executes actions in the remote environment based on the instructions received from the master system. When, in addition to execution, the slave system also sends feedback signals to the master system (allowing the operator to receive information from the remote environment, such as force feedback and sensor data), the system is characterized as bilateral (Niemeyer *et al.*, 2016; Ferre *et al.*, 2007).

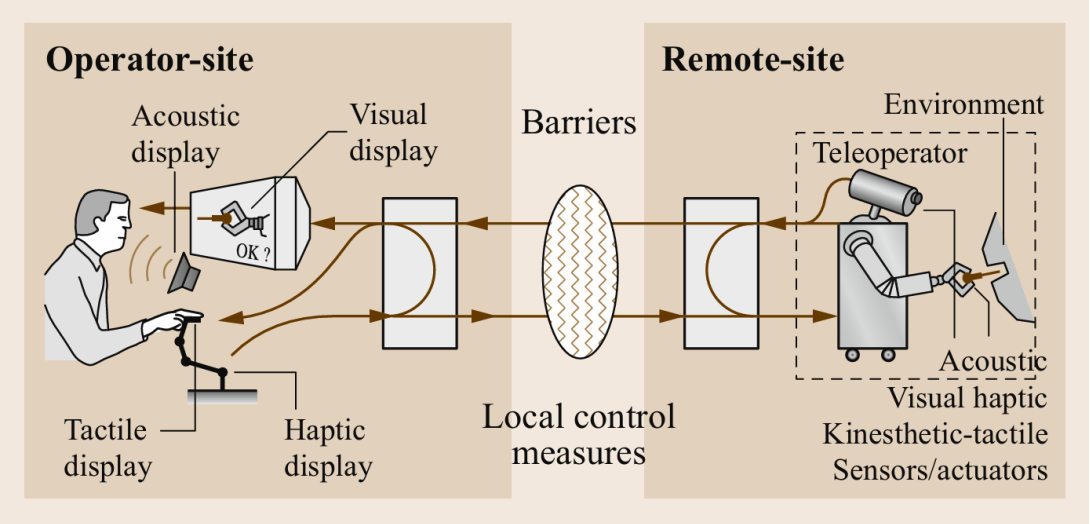
In summary, telerobotics represents an important intersection between robotics and remote human intervention, combining the precise actuation capabilities of robotic systems with the flexibility and judgment of the human operator (Ferre *et al.*, 2007). This integration enables the execution of complex and dangerous tasks in inaccessible or hazardous environments, expanding the boundaries of human action through technology.

2.1.3.1 Historical Context

With a rich and well-established history, telerobotics has its origins in the work of Raymond C. Goertz in the 1940s and 1950s. His early systems were developed to manipulate radioactive materials behind shielded walls, initially using electrical controls based on simple switches to move the axes of the manipulators. However, its operation was limited by the absence of tactile feedback, which led Goertz to design mechanically coupled master-slave systems that allowed natural hand movements and the transmission of forces and vibrations (Niemeyer *et al.*, 2016).

The limitation of these systems was their dependence on mechanical couplings, which restricted the distance between the operator and the environment. Recognizing these constraints, Goertz introduced electrically coupled manipulators, laying the foundation for modern telerobotic systems with bilateral servo control (Niemeyer *et al.*, 2016).

Figure 2 – Overview of a Master-Slave telerobotic system



Source: (Niemeyer *et al.*, 2016).

In the 1960s, the impact of time delay in teleoperation began to be studied, and the concept of supervisory control emerged as an alternative to mitigate communication delay effects. In the 1980s and 1990s, theoretical advances based on Lyapunov stability (Anderson; Spong, 1988) and network theory (Niemeyer; Slotine, 2004) began to guide the development of bilateral systems, significantly expanding research in the field.

From a hardware perspective, the M2 model, developed in 1982 by the Central Research Laboratory in collaboration with the Oak Ridge National Laboratory, was the first telerobotic system to implement force feedback with separation between master and slave electronics. The system was used in military, space, and nuclear applications and served as the basis for the development of the Advanced Servomanipulator (ASM), for remote maintenance tasks (Niemeyer *et al.*, 2016).

In France, also developed for nuclear applications, the Atomic Energy Commission developed bilateral manipulators with computer-assisted functionalities, such as virtual walls and programmed constraints, aiming to improve operator performance (Niemeyer *et al.*, 2016).

With the decline in the use of nuclear energy in subsequent decades, the focus of telerobotics expanded to other areas, such as medicine and underwater operations. Minimally invasive surgery was a milestone in this context, beginning with laparoscopic cholecystectomy in 1987. Several telesurgery systems were developed thereafter, such as the Telepresence Surgery System developed by the Stanford Research Institute and the Laparoscopic Assistant Robotic System (LARS) created by the Watson Research Center. These projects culminated in the development of the da Vinci system, launched by Intuitive Surgical Inc. in 1999, which is now widely used in remote surgical

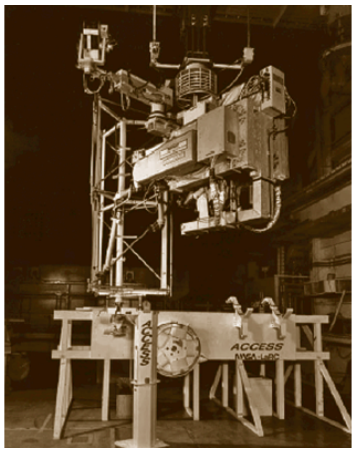
procedures (Niemeyer *et al.*, 2016).

Finally, important milestones such as the transatlantic surgery performed in 2001, in which a surgeon in New York operated on a patient in Strasbourg using the ZEUS system, demonstrated the transformative potential of telerobotics. Although limited by the absence of force feedback, the procedure established the feasibility of performing medical operations remotely with a high degree of precision (Niemeyer *et al.*, 2016).

Figure 3 – Main historical milestones of telerobotics



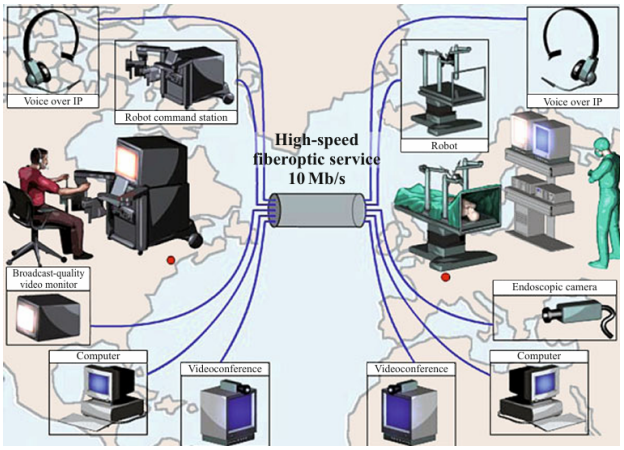
(a) Electrical and mechanical manipulators developed by Raymond C. Goertz (1950s).



(b) CRL Model M2 telerobotic system with force feedback (1982).



(c) *da Vinci* Telerobotic surgical system widely used in telesurgery (1999).



(d) Operation Lindbergh. Transcontinental Telerobotic surgery using the ZEUS (2001).

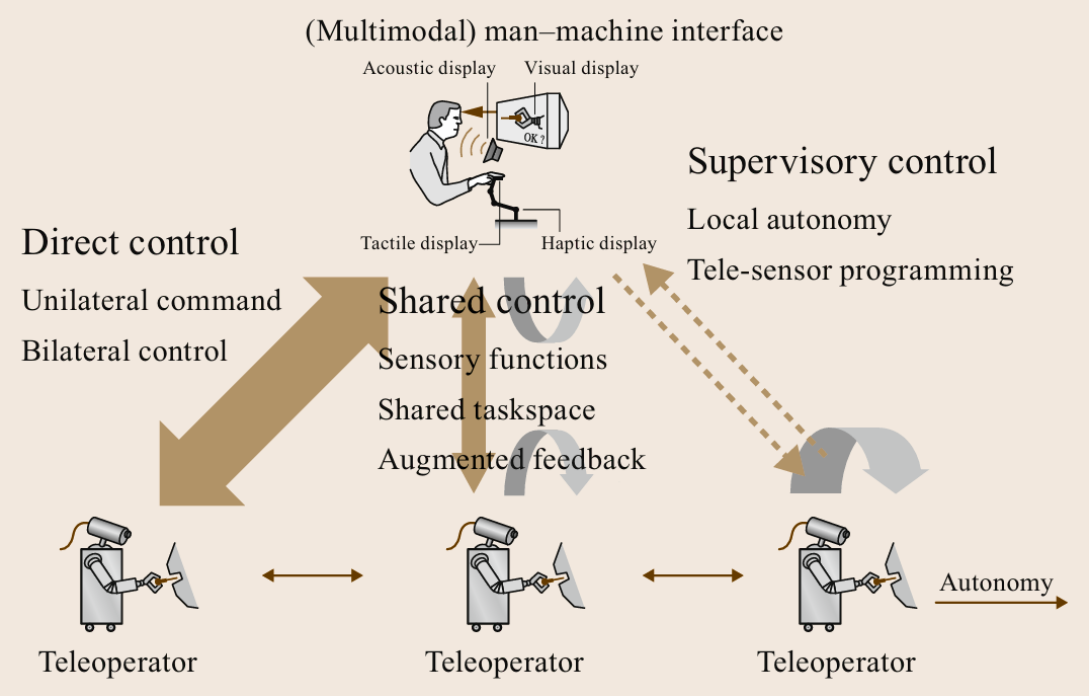
Source: (Niemeyer *et al.*, 2016).

2.1.3.2 Classification of Telerobotic Systems

The classification of telerobotic systems is related to the level of robot autonomy and the level of direct intervention by the human operator. Unlike industrial robotics, in which the robot executes movements or programs without continuous interaction with the user, telerobotic systems establish a continuous channel for the exchange

of information and commands between the operator and the machine. In general, as shown in Figure 4, there are three main control architectures: direct control, shared control, and supervisory control. In practice, however, it is common for real systems to combine elements of more than one of these approaches (Niemeyer *et al.*, 2016).

Figure 4 – Control architectures: Direct, Shared, and Supervisory



Source: (Niemeyer *et al.*, 2016).

2.1.3.2.1 Direct Control

In direct control, all movement of the slave robot is driven by the operator, without any level of local autonomy. This architecture provides a high sense of control and presence, especially when associated with bilateral feedback, in which the operator sends commands to the robot and receives feedback in real time (Niemeyer *et al.*, 2016).

A clear example of this type of system is the control of a vehicle using one or two joysticks, in which commands are proportional to the displacements performed on these devices, allowing control of the vehicle's translation and orientation (Haghjoo, 2018).

On the other hand, direct control requires constant attention from the operator, since even with sensors and environmental modeling, navigation and object manipulation still depend exclusively on continuous human input. This includes tasks such as

avoiding obstacles and precisely positioning the end-effector in complex environments, both static and dynamic (Omarali, 2023).

2.1.3.2.2 Shared Control

Shared control seeks to combine the reliability and sense of presence of direct control with the intelligence and safety guarantees provided by autonomous control. In this model, the slave robot may correct motion commands, regulate subsets of joints or subtasks, or even superimpose additional commands generated autonomously. In scenarios with large communication delays, the operator may be limited to specifying more general trajectory commands, which are refined locally by the robot based on sensory information. This approach also allows the robot to assume subtasks, such as maintaining a stable grasp for long periods or, in surgical applications, compensating for involuntary movements such as the heartbeat by superimposing corrections on the operator's commands (Niemeyer *et al.*, 2016).

A specific application of shared control is the use of virtual constraints, which introduce virtual elements such as surfaces, velocity fields, guides, or tubes into the operator's visual and/or haptic environment. These constraints assist in task execution by limiting movement in forbidden regions or directing it along desired trajectories, modifying or complementing user commands with autonomously generated signals. The concept is analogous to using a ruler to draw a straight line: with assistance, execution is faster and more accurate than performing it freely. Depending on the design of the master system and its controller, these constraints may apply corrective forces or impose position limits, providing an adjustable and programmable level of assistance (Niemeyer *et al.*, 2016).

By leveraging the precision of robotic systems and combining it with human perception and judgment, shared control enables safer, faster, and more intuitive operations. This integration is especially valuable in tasks that require high precision, reducing the operator's cognitive effort and increasing execution efficiency (Niemeyer *et al.*, 2016).

2.1.3.2.3 Supervisory Control

Supervisory control, introduced by Ferrell and Thomas B. Sheridan (1967), is inspired by the analogy of supervising human workers. In this model, the operator provides high-level commands to the robot and receives summarized information about its state and progress. The robot, in turn, executes these directives autonomously, locally closing its own control loops through actuators and sensors. In this way, supervisory control techniques allow the system's autonomy and intelligence to be progressively

transferred to the robot, while the operator maintains close supervision, deciding when and how to intervene (Niemeyer *et al.*, 2016).

A well-known example of supervisory control is the National Aeronautics and Space Administration (NASA) rovers on Mars, in which communication delays of several minutes prevent real-time manual operation. In this case, the operator defines movement objectives, and the rover uses local autonomy and sensory feedback to achieve these objectives, ensuring safe and efficient task execution (Haghjoo, 2018).

2.1.3.3 Conventional Teleoperation

Conventional teleoperation refers to the remote control of robots by human operators using conventional interfaces, such as keyboards, mice, joysticks, and two-dimensional screen. In this model, the operator sends commands to the robot and receives basic visual and sensory information about the remote environment, maintaining direct supervision over the system's actions. This method is widely adopted due to its simplicity and accessibility, and is used in areas where direct human presence is limited or risky (Cornejo *et al.*, 2018).

Joystick-based interfaces, for example, are efficient for operations that require precision and can be implemented in different types of robots, ensuring versatility (Lima *et al.*, 2018). This control configuration is also useful for robots with multiple degrees of freedom, common in complex industrial tasks (Scalera *et al.*, 2018). In the field of medical robotics, the joystick allows surgeons to remotely control robotic equipment in minimally invasive procedures, providing fine and responsive control, which is essential for maintaining surgical precision (Xie; Hou; Wang, S., 2023).

To ensure consistent teleoperation, Parham M Kebria *et al.* (2018) highlights the importance of transparency, which refers to the ability to convey to the operator the sense of the remote workspace through the teleoperation device (Lawrence, 1993). This characteristic is fundamental for the operator to perform tasks effectively and safely. Although control through keyboards and joysticks is widely used in teleoperation systems, there is a growing demand for more efficient interfaces between the user and the robot (Lima *et al.*, 2018).

Conventional control methods are restricted to a two-dimensional screen, despite interacting with a three-dimensional environment, which can make the execution of more complex tasks difficult or even unfeasible (Kebria, Parham M. *et al.*, 2019). This limitation restricts the application of teleoperation, as it does not provide a realistic operator experience, compromising intuition and precision in remote operations.

2.1.3.4 Immersive Teleoperation

The evolution of teleoperation interfaces has increasingly focused on immersive approaches, which have the potential to provide operators with greater spatial percep-

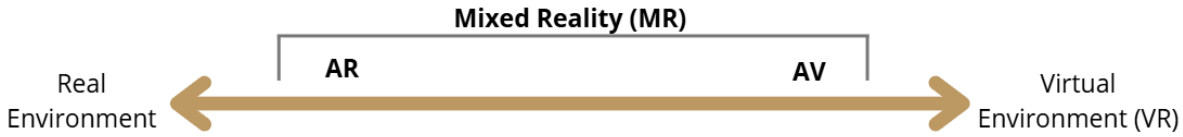
tion, more intuitive control, and improved exploration of the remote environment when compared to conventional interfaces (Omarali, 2023). In recent years, the field of immersive teleoperation has witnessed significant advances, driven by innovations in robotics, haptic interfaces, and networked communication systems (Hernandez-Gobertti *et al.*, 2025).

Immersive teleoperation can be understood as the integration of digital or hybrid environments that allow users to interact intuitively with the remote environment, providing a sense of physical presence at the operation site (Hernandez-Gobertti *et al.*, 2025; Niemeyer *et al.*, 2016; Cambuzat *et al.*, 2018). These systems combine robotic manipulators, haptic interfaces, and visual tracking technologies, enabling the execution of remote tasks with a high degree of realism (Hernandez-Gobertti *et al.*, 2025).

2.1.3.4.1 Concepts and Immersive Technologies

Before discussing teleoperation with immersive technologies, it is essential to define the key concepts associated with this field. One of the most relevant conceptual milestones was proposed by Paul Milgram and Fumio Kishino in 1994, with the introduction of the reality–virtuality continuum (Figure 5). This model presents a continuous axis ranging from a completely real environment, at one end, to a fully virtual environment, at the other. Between these extremes lies the so-called MR, where real and virtual elements coexist and can directly interact (Milgram; Kishino, 1994).

Figure 5 – Adapted representation of the continuum concept



Source: Adapted from Milgram and Kishino (1994)

Based on this taxonomy, three central concepts stand out. Augmented Reality (AR) refers to the view of the physical environment with the overlay of digital information, as seen in devices such as Microsoft HoloLens or mobile applications. In contrast, Augmented Virtuality (AV) describes virtual environments that incorporate real elements, for example, when video streams or sensors from the physical world are integrated into an interactive digital environment. Between these two categories lies MR, which encompasses both AR and AV and represents the space in which real and virtual objects coexist and interact in an integrated manner, enhancing perception and interaction with the environment (Milgram; Kishino, 1994; Skarbez; Smith; Whitton, 2021; Omarali, 2023).

More recently, the term Extended Reality (XR) has emerged as an umbrella concept that encompasses AR, AV, MR, and VR. Unlike the previous categories, XR does not refer to a specific technology, but rather to a set of digitally mediated experiences at different levels of immersion. The “X” in its acronym is interpreted as a variable, representing any form of XR, highlighting its broad and flexible nature (Rauschnabel *et al.*, 2022).

Despite this conceptual organization, some inconsistency in terminology can be observed in the literature. It is not uncommon to find studies that incorrectly classify AR applications as VR, or VR experiences as MR, highlighting the need for greater rigor in the use of this taxonomy (Omarali, 2023; Rauschnabel *et al.*, 2022).

2.1.3.4.2 Mixed Reality-Based Teleoperation

Teleoperation applications, in general, benefit from the use of MR-based interfaces, as they simultaneously combine physical and virtual elements. This overlap is essential when there is manipulation of real objects together with digitally generated information. Since the exact form of this integration may vary, the reality–virtuality continuum proposed by Milgram and Kishino is often used to position different MR configurations (Sun *et al.*, 2020). Although this model also includes AV, the term is rarely used in recent literature. According to Skarbez, Smith, and Whitton (2021), MR has become the most comprehensive and accepted term to describe the integration between real and virtual environments, encompassing both AR and AV within the same conceptual spectrum.

In this context, a key aspect of MR in teleoperation is its potential to enhance telepresence, understood as the sense of presence in the remote environment through the natural transmission of information to the operator. Telepresence can be distinguished into two dimensions: objective and subjective. The objective dimension refers to the use of teleoperation technology to enable the perception and manipulation of remote entities, while the subjective dimension is related to the perceptions and sensations experienced by the user, that is, the level of immersion and the feeling of actually being present in a remote or synthetic environment. This subjective aspect is central, as it reflects how the operator perceives and experiences the teleoperation process (Council *et al.*, 1994).

Despite these descriptions, there is still no widely accepted operational definition of telepresence. Academic debate focuses on three central questions: how to define it in a measurable way, how to create conditions for it to occur, and, above all, what its practical value is. Research indicates that factors such as the exclusion of external stimuli that remind the user of their presence in the immediate physical environment, the predictability and consistency of interactions, the quality and realism of the stimuli

and feedback provided by the system, as well as the user's ability to adapt, are key to the development of telepresence (Council *et al.*, 1994).

Additionally, the relationship between telepresence and performance is also discussed. Although it is common to assume that higher levels of telepresence result in better performance in remote manipulation tasks (Ferre *et al.*, 2007), this correlation is not universal. In certain contexts, such as operations in hazardous environments, a reduced sense of presence may even be desirable, as it avoids psychological exposure to harmful stimuli. Thus, it remains unclear whether telepresence should be pursued as a primary objective in the design of teleoperation systems (Council *et al.*, 1994). Despite recent advances in measuring and modeling this phenomenon, there is still no consensus that telepresence alone ensures better performance (Bechtel *et al.*, 2023; Youssef *et al.*, 2023; Boer, 2021).

In addition, several studies have explored the application of MR in teleoperation across different domains. In healthcare, research indicates its feasibility in surgical procedures, although with limitations in precision, highlighting its potential to enhance control intuitiveness and reduce reliance on conventional interfaces (Ai; Kazanzides; Azimi, 2024). In the industrial sector, MR-based systems have demonstrated gains in efficiency and safety in complex disassembly processes, such as those involving electric vehicle batteries, contributing to reduced execution time and increased operational reliability (Rastegarpanah; Contreras; Stolkin, 2024). In search and rescue operations, comparisons between immersive interfaces show that MR increases operator confidence and situational awareness in real-world scenarios (Cruz Ulloa *et al.*, 2024). These examples highlight the potential of MR to enhance the effectiveness and safety of telerobotic systems across different application contexts.

2.1.3.4.3 Integration and Control Middleware

The effectiveness of an immersive teleoperation system depends not only on the quality of its input and output interfaces but also on the software layer that connects the operator to the robot. This layer is generally composed of middleware, responsible for integrating sensors, actuators, and interaction devices, as well as ensuring real-time communication between the virtual environment and the physical robot (Pizzagalli *et al.*, 2023).

The integration between immersive environments and robotic systems still lacks a consolidated standard. In many cases, researchers and developers need to develop their own solutions to enable communication between different components, which highlights the need for robust middleware that acts as a link between platforms (Wonsick; Padir, 2020).

Despite the absence of a single standard, two widely used tools stand out, playing complementary roles in the development of immersive teleoperation solutions:

Robot Operating System (ROS), established as a reference in robotic applications due to its modularity and flexibility, and **Unity**, a platform focused on building interactive and immersive virtual environments. The combination of these technologies has proven especially promising for integrating human operators with robotic systems in a natural and efficient way (Gallipoli *et al.*, 2024; Kim *et al.*, 2018; Galarza *et al.*, 2023; Zhang *et al.*, 2023; Yim *et al.*, 2022; Udekwe; Seyyedhasani, 2025; Gloumeau; Pettifer, 2024; Whitney *et al.*, 2020; Su, Y. *et al.*, 2022; Whitney *et al.*, 2018).

ROS is an open-source framework widely used in research and industrial applications due to its ability to modularize and standardize communication between processes. It adopts a distributed architecture in which different system components are implemented as *nodes*, allowing each robot functionality (such as motion control, sensor acquisition, or actuator interfacing) to operate independently and cooperatively (ROS Wiki, 2025a). There are four fundamental concepts that need to be understood to grasp how ROS works:

- **Nodes**: Processes or executables that run in parallel, each responsible for a specific task within the system. For example, there may be one node responsible for reading sensor data, another for computing robot trajectories, and another for sending commands to the actuator. Each node is independent but communicates with others through topics, services, or parameters.
- **Messages**: Data structures used for exchanging information between nodes. They can contain simple data (such as integers) or complex data (such as images, point clouds, or transformation matrices). Messages are published on topics or sent via services, ensuring that nodes can share information consistently.
- **Topics**: Asynchronous communication channels based on the publisher/subscriber model. A publishing node sends messages on a given topic, and multiple subscriber nodes receive these messages in real time. For example, a sensor node publishes periodic values, while a control node subscribes to this data to make decisions.
- **Services**: A synchronous communication mechanism in a request–response format. A client node sends a request to a server node, which processes the operation and returns a response. Services are used when it is necessary to ensure that an operation is completed before proceeding, such as requesting the computation of a specific trajectory.

This modular architecture allows ROS to be adaptable, supporting everything from educational robots to industrial systems (Vairavasamy *et al.*, 2022; Garg; Kuts; Anbarjafari, 2021). The interaction between nodes, messages, topics, and services enables the system to operate in a distributed, coordinated, and real-time manner, allowing

the integration of sensors, actuators, and teleoperation interfaces in a synchronized manner.

Another essential element for integrating robots into ROS is **Unified Robot Description Format (URDF)**. This format uses XML files to describe the robot's kinematic and geometric structure. URDF enables the creation of a digital representation of the robot, which can be used in simulations and visualization tools, and serves as the basis for motion planning algorithms (ROS Wiki, 2025g).

In this context, **MoveIt** emerges as one of the most important packages within the ROS ecosystem. It is a platform for robotic motion planning that integrates kinematics algorithms (forward and inverse), trajectory planning, and collision detection. MoveIt uses the URDF model to compute optimized trajectories between initial and final states defined by the operator (Coleman *et al.*, 2014).

For visualization of the robot and control interfaces within the ROS environment, the **Rviz** software is used. This 3D visualization tool allows real-time display of data such as point clouds, images, reference frames, trajectories, and joint states. Additionally, Rviz is used to validate the URDF model, verify communication between nodes, and inspect topics, supporting both system development and operation (ROS Wiki, 2025c).

On the other hand, **Unity** constitutes the interface layer responsible for enabling immersive interaction between the operator and the robot during teleoperation. It provides a three-dimensional representation of the robot, allowing the user to visualize it from different perspectives, enhancing the sense of telepresence in the system. In addition, the platform integrates XR devices, enabling gestures, commands, and movements of the operator to be translated into corresponding robot actions. The physics engine built into Unity helps evaluate motions beforehand, allowing the simulation of collisions, forces, and mechanical constraints, which enables simulation-based training and consequently reduces risks during real robot teleoperation. Another important aspect is its extensibility in C#, making it possible to implement specific controllers that process input signals from different modalities, such as handheld controllers, gestures, or voice commands, and provide visual, auditory, or haptic feedback when integrated with haptic devices (Unity Technologies, 2025e).

In this work, ROS was used to manage communication between the different components of the system, MoveIt was used for trajectory planning, and Unity was used to implement the immersive user interface, with the integration of these tools forming the basis of the proposed teleoperation system.

2.1.4 State-of-the-Art in Immersive Teleoperation

Despite the growing popularity of robotic teleoperation based on immersive technologies, this field is still in an early stage of development. It is not clearly established how conventional control and visualization techniques can be systematically transferred

and adapted to immersive human–robot interaction interfaces. This gap highlights the need for comparative studies that investigate, in a structured manner, both the potential benefits and the inherent limitations of applying these technologies in practical and industrial scenarios (Omarali, 2023).

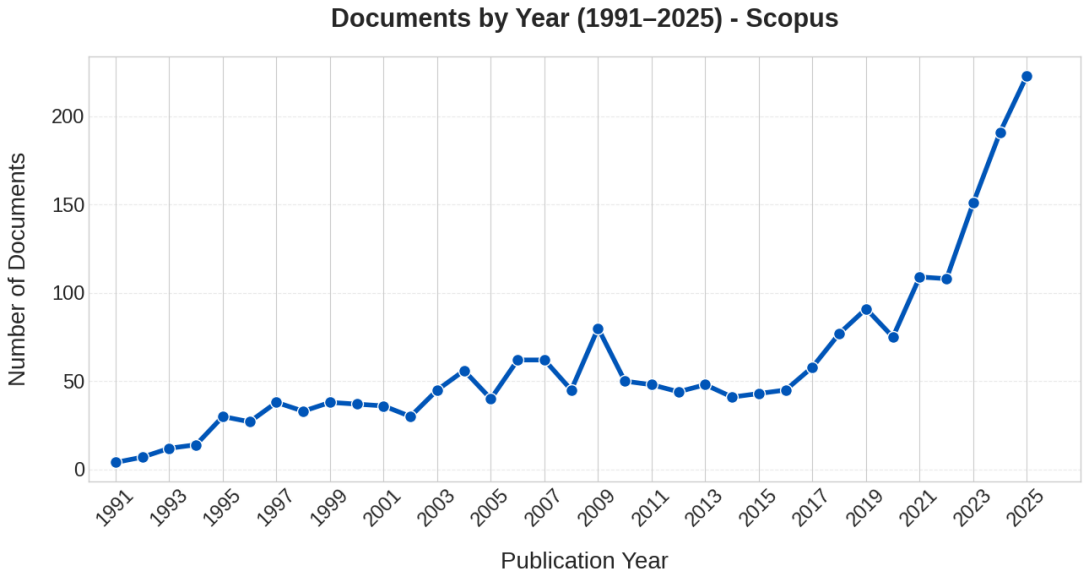
The relevance of this topic can also be observed in the recent trend of growth in scientific publications related to the subject. A structured search conducted in the Scopus database in April 2026 revealed a significant increase in the number of studies addressing immersive technologies applied to teleoperation. This growth, particularly evident since 2017, reflects the increasing interest of the scientific community in exploring the potential of these approaches to improve performance, safety, and operator experience in industrial environments.

A key factor driving this growth is the increased availability of hardware for immersive technologies. The reduction in costs of devices such as Head-mounted displays (HMDs), tracking sensors, and haptic controllers, combined with compatibility with development platforms, has expanded access for researchers and institutions to advanced experimental environments (Al-Ansi *et al.*, 2023). This increased accessibility of equipment enables the execution of more complex and diverse studies, promoting the investigation of new human–robot interaction paradigms and the validation of immersive techniques in real-world scenarios.

Although the number of studies is increasing, there is still a relatively limited number of in-depth investigations on the integration of immersive technologies in teleoperation. In the survey conducted in Scopus, only 2,127 publications corresponding to the defined keywords (Table 1) were identified, covering the period from 1991 to April 2026, as shown in Figure 6. For better visualization of the growth trend, the figure presents data up to 2025.

As discussed in Section 2.1.3.4.1, most immersive teleoperation applications fall within the MR spectrum (as they combine virtual and real elements). However, it is not uncommon for studies to incorrectly classify VR applications as MR, or to use the term XR to broadly refer to MR. To avoid excluding relevant studies due to the lack of terminological standardization, a comprehensive search string was adopted, covering the most common terms in the literature on immersive interfaces applied to teleoperation. Table 1 presents the search string used.

Figure 6 – Number of comparative studies found in the Scopus database using the search string from Table 1



Source: Adapted from Elsevier (2026).

Table 1 – Search string used in the general review

(TITLE-ABS-KEY (("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND teleoperation)) OR
(TITLE-ABS-KEY (("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND telemanipulation)) OR
(TITLE-ABS-KEY (("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND "remote operation"))

Source: Author (2026).

This strategy made it possible to encompass studies that use different terms to describe this emerging technology, which still lacks a consolidated nomenclature, providing a more comprehensive analysis of the state of the art and reflecting the terminological diversity present in the literature (Rauschnabel *et al.*, 2022).

2.1.4.1 Comparison of Conventional and MR Interfaces

Comparative studies are essential for objectively evaluating the performance of different technologies, identifying their advantages and limitations, and providing a solid basis for informed decisions regarding their adoption in practical environments (Lau; Holbrook, 2017). In the context of robotic teleoperation, the comparison between conventional and immersive interfaces plays a key role in understanding to what extent

MR can effectively contribute to improving human–robot interaction.

These analyses allow for an objective assessment of whether the increased level of immersion provided by MR translates into measurable improvements in performance, accuracy, and usability, or whether its effects remain restricted to the perceptual and subjective domain of the user experience.

With the aim of obtaining a specific overview of the current state of comparative research, a new structured search was conducted in the Scopus database in April 2026, with the objective of identifying studies that simultaneously analyze conventional and immersive methods applied to the teleoperation of robotic arms. This step aimed to refine the previous survey to works that present results directly relevant to the purpose of this dissertation.

To ensure a broad and inclusive search, the term “COMPAR*” was added to the general search string previously used (Section 2.1.4), in order to capture all variations of the term “comparing” (such as “comparison”, “comparative”, etc.), including studies that describe analyses between different approaches. This strategy resulted in a total of 563 publications identified (Figure 7). For better visualization of the growth trend, data from 2026 were excluded, and the figure was limited to include data up to 2025. The complete search string used in the query is presented in Table 2.

Table 2 – Search string for comparative studies

(TITLE-ABS-KEY(("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND teleoperation AND (compar* OR versus OR "vs" OR evaluation))) OR
(TITLE-ABS-KEY(("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND telemanipulation AND (compar* OR versus OR "vs" OR evaluation))) OR
(TITLE-ABS-KEY(("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND "remote operation" AND (compar* OR versus OR "vs" OR evaluation)))

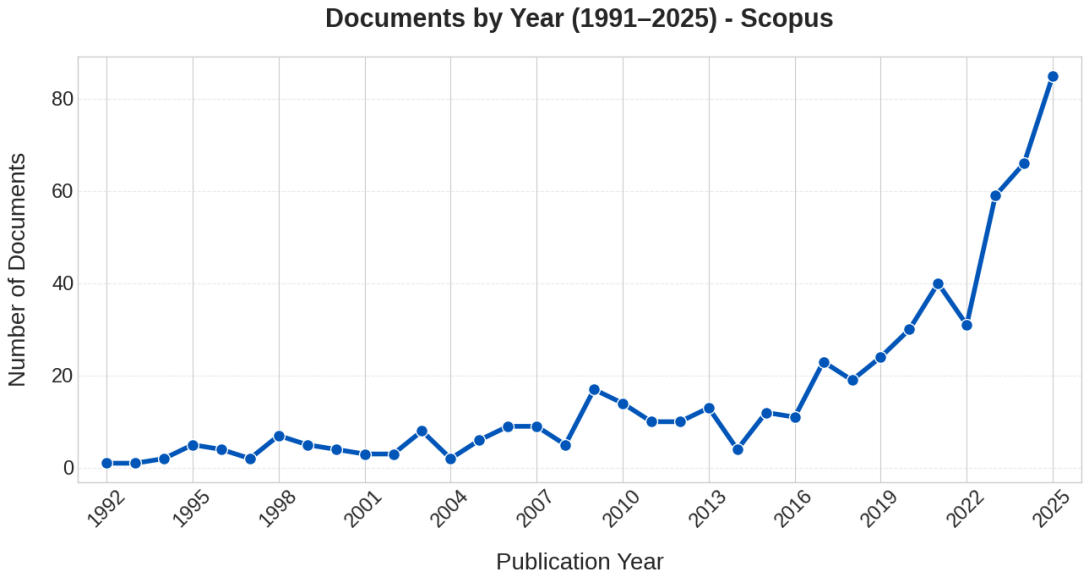
Source: Author (2026).

Each identified document was individually evaluated and subjected to a screening process based on inclusion criteria aligned with the objective of this work. Only studies that:

(i) employed a robotic manipulator as a central component of the teleoperation system, excluding works focused on mobile vehicles, drones, or Remotely Operated Vehicles (ROVs) without an attached robotic arm;

(ii) performed teleoperation in a real physical environment, rather than only in simulation;

Figure 7 – Number of comparative studies identified in the Scopus database using the search string from Table 2

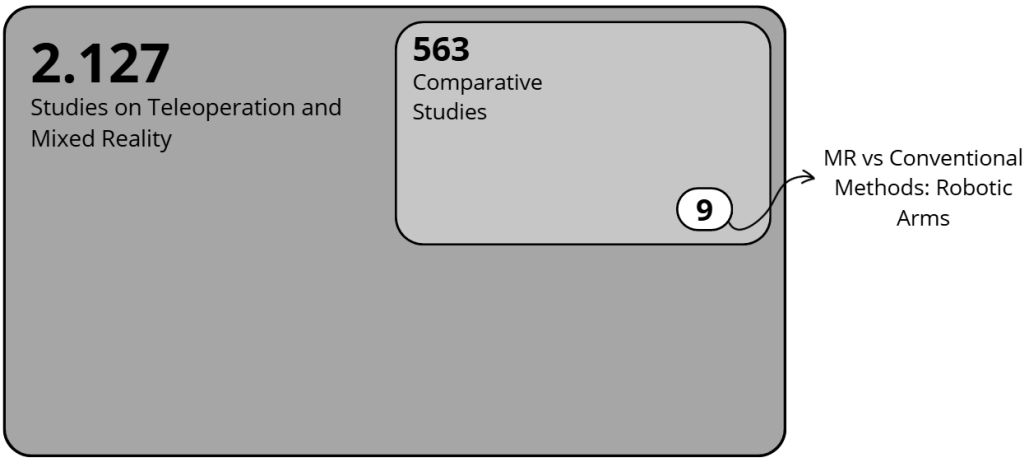


Source: Adapted from (Elsevier, 2026).

(iii) presented both quantitative and qualitative comparative data between conventional and immersive teleoperation methods were selected.

After applying these criteria, 9 studies fully met the requirements defined for the final analysis. This number represents approximately 1.6% of the 563 results and about 0.4% of the 2,127 publications identified in the general search presented earlier, indicating that comparative studies involving real robotic arms are significantly less frequent within the body of research on immersive teleoperation, as illustrated in Figure 8.

Figure 8 – Schematic overview of the refinement of the identified studies



Source: Author (2026).

Table 3 – Synthesis of comparative studies between conventional and immersive interfaces

Author / Year	Study Objective	Quantitative Metrics				Qualitative Metrics							
		Time	Accuracy / Performance	Success Rate	Collisions / Errors	NASA-TLX	SUS	Ease of Use	GTQ	TAM	MEC-SPQ	Difficulty Scale	Likability
Pryor *et al.* (2023)	Compare conventional and MR interfaces in a high-risk and high-latency environment.	X	X	X	–	X	–	–	–	–	–	X	–
LeMasurier *et al.* (2024)	Compare a 2D interface with MR for planning in a mobile manipulator robot.	–	X	–	X	X	X	X	X	–	–	–	–
Whitney *et al.* (2020)	Compare conventional teleoperation and MR in precise pick-and-place tasks.	X	–	–	–	X	X	–	–	–	–	–	X
Condori *et al.* (2025)	Evaluate an MR interface for position control of a SCARA manipulator compared to the conventional method.	X	–	X	–	X	X	–	–	–	–	–	–
Yunpeng Su *et al.* (2022)	Investigate the use of MR for teleoperation in inspection tasks compared to the conventional method.	X	–	X	–	X	–	–	–	X	–	–	–
Wu, Zhao, and Li (2025)	Evaluate intuitive teleoperation with imitation-based mapping in MR for a robotic arm.	X	–	X	–	X	X	–	–	–	–	–	–
Kanazawa, Sato, and Morita (2024)	Compare different levels of immersion in teleoperation focusing on performance and user experience.	X	–	–	–	X	–	–	–	–	–	–	–
Jankowski and Grabowski (2015)	Evaluate the usability of an MR vs conventional interface in a manipulator attached to a mobile robot.	X	–	X	–	–	–	X	–	–	X	–	X
Hasan, Yasar, and Iqbal (2026)	Evaluate teleoperation interfaces in multiple robotic manipulation scenarios.	X	–	X	–	X	–	–	–	–	–	–	X

Source: Author (2026)

Table 3 presents a synthesis of the nine identified studies, including the type of application, the main objective, and the metrics used in each study. All studies focused on directly comparing conventional and immersive teleoperation systems employing real robotic manipulators. Although the experimental scenarios and robotic platforms differed considerably, clear methodological trends can be observed regarding the evaluation of human performance and operator perception.

In general, the most recurrent quantitative metrics were execution time, positional accuracy, success rate, and number of collisions, representing objective indicators of operational performance.

Similarly, among the subjective measures, usability was most commonly assessed through the System Usability Scale (SUS) (Brooke *et al.*, 1996), whereas cognitive and physical workload during task execution was predominantly evaluated using the NASA Task Load Index (NASA-TLX) (Hart; Staveland, 1988). Both instruments present strong psychometric validation, understood as the set of evidence supporting the reliability, internal consistency, and validity of the evaluated constructs (Kline, 2015), and are widely adopted in human–robot interaction research (Mortazavi *et al.*, 2024).

Although a variety of subjective instruments were identified across the reviewed studies, this diversity does not necessarily correspond to a broader coverage of evaluation dimensions.

For example, the Gas Tank Questionnaire (GTQ) (Monfort *et al.*, 2018) constitutes an alternative workload-oriented instrument, addressing dimensions that overlap with those traditionally assessed through NASA-TLX. Likewise, the Technology Acceptance Model (TAM), proposed by Fred D Davis (1989), evaluates perceived usefulness, ease of use, and intention to adopt, whereas the Ease-of-Use (EoU) (Davis, F. D., 1989) focuses on the perceived ease of interacting with a system. Similarly, subjective difficulty ratings (*difficulty rating*) (Pryor *et al.*, 2023) assess the perceived complexity of task execution. Although conceptually distinct and developed for different purposes, these instruments remain closely associated with workload, usability, and task-related performance, dimensions already well established through NASA-TLX and SUS.

Likewise, measures of likability or user preference (*system likability*) (Whitney *et al.*, 2020) occupy an intermediate position between pragmatic and experiential evaluations. While they provide an indication of which interface users prefer, they do not reveal whether that preference is driven by usability, workload, immersion, or other experiential factors.

The main exception is the MEC Spatial Presence Questionnaire (MEC-SPQ), which evaluates spatial presence and immersion, dimensions that extend beyond task performance and usability. Nevertheless, even this instrument remains focused on a specific experiential aspect of interaction rather than providing a broader assessment of user experience.

From a user experience perspective, these subjective measures can be broadly interpreted according to two complementary dimensions (Hassenzahl, 2001). Pragmatic dimensions are associated with task execution and system utility, encompassing aspects such as workload, usability, ease of use, usefulness, and task difficulty. Hedonic dimensions, in contrast, are associated with experiential aspects of the interaction, including immersion, stimulation, engagement, enjoyment, and the emotional value attributed to the system.

As highlighted by Apraiz, Lasa, and Mazmela (2023), emotional and experiential factors such as trust and satisfaction remain comparatively underexplored within human–robot interaction research. The findings of the present review reinforce this observation, as most of the subjective instruments identified in the reviewed studies primarily target pragmatic aspects of interaction, whereas comparatively few explicitly address experiential or hedonic dimensions. This distribution is summarized in Table 4.

Table 4 – Predominant user experience dimensions addressed by the subjective instruments identified in the reviewed studies

Instrument	Pragmatic focus	Hedonic focus
NASA-TLX	High	None
SUS	High	None
GTQ	High	Moderate
EoU	High	None
TAM	High	None
Difficulty Rating	High	None
Likability	Moderate	Moderate
MEC-SPQ	Limited	High

Source: Author (2026), based on the primary constructs evaluated by each instrument.

This imbalance is particularly relevant in the context of immersive technologies. Many of the reviewed studies attribute positive outcomes to characteristics such as intuitiveness, natural interaction, immersion, engagement, or user preference (Whitney *et al.* (2020); Yunpeng Su *et al.* (2022); Condori *et al.* (2025); Wu, Zhao, and Li (2025)). However, these characteristics are frequently inferred from workload, usability, acceptance, or performance measures rather than being evaluated directly through dedicated user experience instruments.

Consequently, it often remains unclear whether users preferred a given interface because it was easier to use, less cognitively demanding, more immersive, more stimulating, or simply more enjoyable.

These limitations reinforce the need for instruments capable of evaluating user experience in a standardized and statistically validated manner, especially in immersive contexts. In this scenario, the User Experience Questionnaire (UEQ) (Laugwitz; Held; Schrepp, 2008) stands out as it is widely used in User Experience (UX) research

and recognized as one of the most employed questionnaires in recent human–robot interaction studies (Mortazavi *et al.*, 2024).

Unlike the instruments identified in the reviewed studies, the UEQ evaluates both pragmatic and hedonic aspects of user experience, allowing the investigation of experiential factors such as stimulation, novelty, and attractiveness, in addition to the usability and workload related aspects traditionally considered in teleoperation studies.

For this reason, the UEQ constitutes a methodological complement to the SUS and NASA-TLX questionnaires. While the SUS evaluates perceived usability and the NASA-TLX evaluates workload, the UEQ simultaneously assesses pragmatic and hedonic dimensions of user experience. Thus, this study adopts the combination of these three instruments, which will be detailed in Section 5.6.2, as a standardized framework for the comparative evaluation of conventional and immersive robotic teleoperation interfaces.

The adopted methodological framework also shows conceptual convergence with the principles of the international standard ISO 9241-11:2018 (International Organization for Standardization, 2018). This standard defines quality in use through three main dimensions: effectiveness, efficiency, and satisfaction within a specified context of use. In this research, effectiveness is assessed through objective performance measures, including task success rate, accuracy, and collision occurrence, while efficiency is evaluated through execution time and perceived workload using the NASA-TLX questionnaire. Satisfaction is examined through the complementary application of the SUS and UEQ-S instruments. Although both contribute to the assessment of user satisfaction, they address distinct aspects of the interaction: SUS focuses on perceived usability and the functional quality of the system, whereas UEQ-S captures the users' overall experience, encompassing both pragmatic and hedonic qualities. Together, these quantitative and qualitative measures provide a comprehensive evaluation of teleoperation performance, workload, usability, and user experience, maintaining alignment with the quality-in-use framework proposed by ISO 9241-11:2018.

Finally, to verify the originality of this integrated evaluation framework, a structured search was conducted in the Scopus database in April 2026 with the objective of identifying studies that simultaneously employed the SUS, NASA-TLX, and UEQ within immersive teleoperation contexts. Table 5 presents the search string used in this stage.

The search returned no results, indicating that, to date, there are no records of immersive teleoperation studies that simultaneously combine the three questionnaires within the same experimental protocol.

Subsequently, a new search was conducted in the same database, removing terms related to immersion, that is, without including the expressions (virtual reality", VR", mixed reality", MR", or "XR"), in order to also encompass conventional teleoperation studies. This search returned only two additional studies (Martynov *et al.*, 2023;

Table 5 – Search string used to identify studies that simultaneously apply SUS, NASA-TLX, and UEQ

(TITLE-ABS-KEY (("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND teleoperation AND (SUS OR "system usability scale") AND (NASA OR "NASA-TLX" OR "task load index") AND (UEQ OR "user experience questionnaire"))) OR
(TITLE-ABS-KEY (("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND telemanipulation AND (SUS OR "system usability scale") AND (NASA OR "NASA-TLX" OR "task load index") AND (UEQ OR "user experience questionnaire"))) OR
(TITLE-ABS-KEY (("virtual reality" OR VR OR "mixed reality" OR MR OR "extended reality" OR XR) AND "remote operation" AND (SUS OR "system usability scale") AND (NASA OR "NASA-TLX" OR "task load index") AND (UEQ OR "user experience questionnaire")))

Source: Author (2026)

Cabrera *et al.*, 2024). However, neither satisfied the inclusion criteria. One study addressed the teleoperation of a drone platform without the use of immersive technologies (Martynov *et al.*, 2023), while the other was outside the scope of comparative evaluations between immersive and conventional teleoperation systems (Cabrera *et al.*, 2024). Consequently, no additional studies were included.

These results suggest that, although well-established instruments exist for evaluating usability, workload, and user experience, their integration within a single evaluation framework remains uncommon in teleoperation research and was not identified in the immersive teleoperation literature reviewed in this study. This absence is particularly relevant considering that the UEQ complements traditional usability and workload measures by incorporating both pragmatic and hedonic dimensions of user experience. Consequently, the combined use of SUS, NASA-TLX, and UEQ represents a methodological approach capable of providing a more comprehensive assessment of operator performance and perception than that typically reported in the reviewed literature.

2.1.4.2 Identified Gap and Research Contribution

The findings of the literature review indicate a progressive methodological convergence regarding the evaluation of operational performance, usability, and cognitive workload in robotic teleoperation. Objective metrics such as execution time, success rate, accuracy, and collisions are widely adopted, while the SUS and NASA-TLX have become established references for the assessment of usability and workload, respectively.

However, the review also revealed that this convergence is not accompanied by an equivalent consolidation in the evaluation of user experience. Although several subjective instruments were identified, most of them remain focused on pragmatic aspects of interaction, including usability, ease of use, usefulness, technology acceptance, workload, and task difficulty. By contrast, comparatively few approaches explicitly address experiential or hedonic dimensions of the user experience.

Therefore, the identified gap does not lie in the absence of subjective evaluation instruments, but rather in the limited integration of pragmatic and hedonic dimensions within a standardized evaluation framework. While workload and usability are consistently assessed through validated instruments, no methodological convergence was observed for the evaluation of broader user experience dimensions in immersive teleoperation studies.

The main contribution of this dissertation is the proposal of an integrated evaluation framework that combines objective performance metrics with the complementary application of NASA-TLX, SUS, and UEQ. Within this framework, NASA-TLX evaluates cognitive workload, SUS evaluates perceived usability, and UEQ evaluates both pragmatic and hedonic dimensions of user experience. This combination enables a more comprehensive evaluation of operator perception by integrating workload, usability, and user experience within a single psychometrically validated framework.

Furthermore, the proposed framework remains conceptually aligned with the quality-in-use principles defined in ISO 9241-11:2018 (International Organization for Standardization, 2018), integrating effectiveness, efficiency, and satisfaction within a unified evaluation approach. In this way, the research contributes not only by applying an additional questionnaire, but by proposing a more comprehensive methodology for the comparative evaluation of conventional and immersive robotic teleoperation interfaces.

3 HARDWARE AND SOFTWARE INFRASTRUCTURE

This chapter describes the hardware and software infrastructure used in the experimental system, including the devices, sensors, computers, controllers, and development tools employed throughout the implementation and evaluation stages.

3.1 HARDWARE

3.1.1 FANUC LR Mate 200iD/7L Robot and R-30iB Mate Controller

For the experiments, the industrial robot FANUC LR Mate 200iD/7L (Appendix A) was used, a six-degree-of-freedom manipulator designed for applications that require precision, agility, and repeatability. The model has a nominal payload capacity of up to 7 kg, a maximum reach of 911 mm, and repeatability of ± 0.01 mm.

The controller used was the R-30iB Mate, the manufacturer's standard unit for this line of manipulators. It is responsible for interpreting the commands sent to the robot and managing motion, safety, and communication parameters. In the experimental architecture, the R-30iB Mate acts as a link between the teleoperation interfaces and the physical manipulator, allowing the execution of commands sent by the teleoperation interfaces. Like the robot itself, the controller was identical in all experimental conditions, ensuring that the comparison focused exclusively on the teleoperation interfaces.

In this study, the robot–controller set was used as the physical execution platform, receiving position and trajectory commands from both the conventional and immersive interfaces. The choice of this model is justified by three main factors:

- availability of the equipment at SENAI Institute for Innovation in Manufacturing Systems (ISI-SM);
- industrial-grade positioning accuracy and repeatability, reducing the influence of the robotic platform on the comparison between control interfaces;
- compatibility with ROS libraries and packages, allowing integration with simulation tools, trajectory planning, and MR systems.

Figure 9 – FANUC LR Mate 200iD/7L robot and R-30iB Mate controller



(a) FANUC LR Mate 200iD/7L robot.



(b) R-30iB Mate controller.

Source: (a) (FANUC, 2025b); (b) (FANUC, 2025a).

3.1.2 Gripper: MyGripper F100

The gripper used in the experiments was the MyGripper F100 model (Figure 10), from *Elephant Robotics*. It is a compact electric actuator designed for educational and laboratory applications, whose selection was mainly motivated by its native compatibility with ROS. This feature includes support for control libraries, integration with simulation environments, and real-time data acquisition, features that simplified integration with the manipulator and the software infrastructure used in this work.

From an operational point of view, the gripper has an adjustable stroke of up to 100 mm, multiple torque levels for force control, a nominal payload capacity of 500 g, and repeatable positioning accuracy of ± 0.5 mm, parameters sufficient for the proposed manipulation tasks (Section 5.4). The complete technical specifications are presented in Appendix B.

As with the other physical components, the gripper is not a comparison variable, but acts as an auxiliary element necessary for the execution of teleoperation tasks.

Figure 10 – MyGripper F100



Source: (Elephant Robotics, 2025a).

3.1.3 Monitoring Cameras

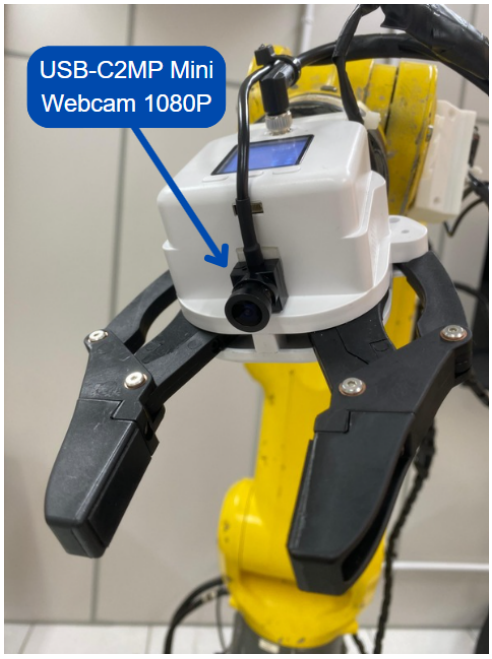
For monitoring of the physical environment and transmission of images to the virtual environment, two cameras were used with different purposes. The first was positioned at a fixed point to cover the entire robot workspace, while the second was attached to the manipulator gripper, with the objective of enabling close visualization of the grasping and object manipulation process.

The camera attached to the robot, model Securexpert USB-C2MP Mini Webcam 1080P, was selected due to its compact size and also for offering a wider field of view than conventional webcams. This model has the following technical specifications:

- **Video resolution:** up to 1920×1080 pixels at 30 FPS (Full HD); also compatible with 720p and 480p;
- **Lens:** 2.8 mm (120° ultra wide-angle);
- **Video format:** MJPEG and YUY;
- **Compatibility:** Windows (XP/7/8/10), Linux, iOS, and Android;

The Figure 11 shows the camera attached to the robot gripper.

Figure 11 – Camera attached to the gripper



Source: Author (2026).

A Logitech C270 Webcam was used as the fixed camera and positioned on a tripod facing the robot (Figure 12). This setup allowed the entire workspace to be recorded during the experiments, supporting the subsequent analysis of task execution and performance metrics. Its main technical specifications are:

- **Video resolution:** 1280×720 pixels at 30 FPS (HD);
- **Field of view (FOV):** 60°;
- **Focus:** fixed, with a minimum distance of 0.5 m;
- **Interface:** USB 2.0;
- **Compatibility:** Windows, macOS, and Linux.

Figure 12 – Fixed camera positioned for workspace visualization



Source: Author (2026).

3.1.4 Arduino and Laser Sensor

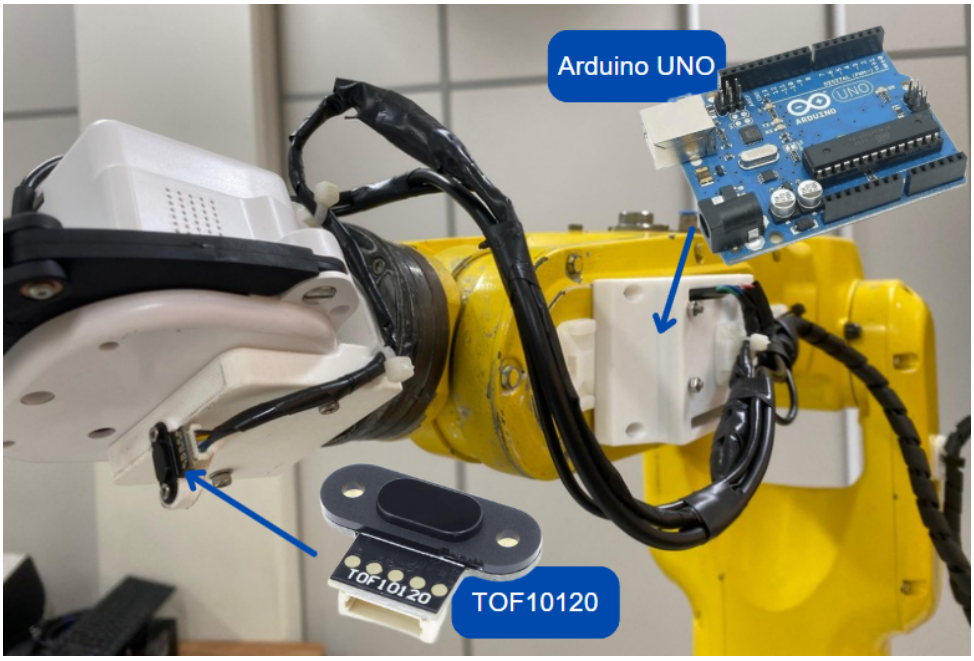
To complement the operator's spatial perception during teleoperation tasks, an auxiliary distance measurement system was implemented using an Arduino Uno microcontroller (Arduino SA, 2024) and a TOF10120 laser sensor (Sharp Corporation, 2025). The system was intended to provide a visual indication of the distance between the robot gripper and objects in the environment, contributing to the sense of depth and distance within the virtual environment.

The TOF10120 operates according to the time-of-flight principle, calculating distance from the travel time of an infrared light signal reflected by the target object, and according to the manufacturer, the measurement range extends from 10 to 1800 mm (Sharp Corporation, 2025).

The Arduino Uno was used to acquire the sensor measurements and transmit them to ROS in real time. These data were subsequently incorporated into the both teleoperation interface, where the measured distance was presented to the operator as an additional visual cue during robot manipulation.

The Arduino Uno and the TOF10120 were selected because of their low cost, availability, and straightforward integration with the proposed architecture. Figure 13 shows the Arduino installed in a protective enclosure and the sensor mounted on the robot gripper.

Figure 13 – Arduino Uno and TOF10120 sensor installed on the robot



Source: Author (2026).

3.1.5 Head-Mounted Display: Meta Quest 3

For the developed immersive interface, the Meta Quest 3 HMD was adopted. The choice is mainly justified by its wide compatibility with the Unity development environment, the platform used to build the teleoperation module, and by its stable support in Windows systems, used as the computational base (of the immersive system) in the experiments (Figure 14).

The Meta Quest 3 offers a resolution of 2064 × 2208 pixels per eye, a refresh rate of up to 120 Hz, and a field of view of approximately 110°, parameters that allow high visual clarity and improved spatial perception during task execution. The headset also provides inside-out tracking of position and orientation, which eliminates the need for external sensors and contributes to the practicality of experimental setup (Meta, 2025b).

Figure 14 – Meta Quest 3 HMD



Source: (Meta, 2025b).

3.1.6 Computers

To enable a teleoperation system, which assumes a physical distance between the operator and the robot, the use of two computers is required: one controlled by the operator (master) and another responsible for direct robot control (slave). This configuration follows the classic *master–slave* architecture, in which interaction and execution functions are distributed across different processing units (Niemeyer *et al.*, 2016). The complete system architecture will be detailed in Section 4.1 and Section 4.2.

The master computer is a Dell Precision 7780, configured with a 13th-generation Intel Core i9-13950HX processor, 64 GB of RAM, an NVIDIA RTX 5000 Ada GPU (16 GB GDDR6), and a 4 TB NVMe SSD, running Windows 11. This configuration was necessary due to the graphical and processing demands associated with using the Meta Quest 3 together with the Unity environment.

The slave computer consists of a Dell Inspiron 15 3520, equipped with an Intel Core i5-1235U processor, 12 GB of RAM, and a 512 GB SSD, running Ubuntu 20.04. The choice of this operating system is due to its wide compatibility with ROS and support for integration between the software modules used in the experiments.

3.2 SOFTWARE

3.2.1 Unity

The MR interface was developed using Unity version 2022.3.45f1, a Long Term Support (LTS) release. This version was selected due to its stability, long-term support, and compatibility with the XR development tools adopted in this work. Another factor considered in the selection of Unity was its accessibility for academic and research applications through educational licensing programs (Unity Technologies, 2025e).

The implementation of MR interaction was enabled through the use of the *XR Interaction Toolkit* (Unity Technologies, 2025g), together with *OpenXR* (Unity Technologies, 2025b). While the former provided the basis for developing interaction mechanisms such as hand tracking, object manipulation, and spatial control, the latter ensured compatibility with different XR devices, standardizing communication between Unity and the hardware. This combination provided native support for the Meta Quest 3 headset and enabled the development of custom C# controllers for mapping operator movements to robot actions.

Figure 15 – Unity logo



Source: (Unity Technologies, 2025a)

3.2.2 Meta Quest Link

Meta Quest Link is Meta's official software for connecting the HMD to a computer, with native support for Windows (Meta, 2024). It allows applications such as Unity to run on the computer while receiving tracking and input data from the headset and controllers. In this work, Meta Quest Link was used to connect the Meta Quest 3 to Unity, enabling the visualization of the MR environment and the tracking of the controller movements in real time.

Figure 16 – Meta Quest Link logo



Meta Quest Link

Source: (Meta, 2024).

3.2.3 ROS

In this project, ROS 1 Noetic (ROS Wiki, 2025b) was adopted, running on Ubuntu 20.04, due to compatibility with ROS-Industrial (ROS-Industrial, 2024). This infrastructure allowed the integration of the FANUC LR Mate 200iD/7L manipulator with MoveIt, responsible for trajectory planning, collision checking, and motion execution.

Figure 17 – ROS and Noetic logos



Source: (ROS Wiki, 2025b).

3.2.3.1 ROS-Industrial

ROS-Industrial extends ROS to industrial applications by providing packages, models, and standardized interfaces that facilitate the integration of commercial robots with simulation and control systems (ROS-Industrial, 2024). For this work, the repository dedicated to FANUC robots was particularly relevant, as it provides drivers, configuration files, and URDF models compatible with MoveIt, allowing access to the robot controller's low-level interface and enabling direct command transmission to the joints and execution of trajectories planned by MoveIt.

The standardization promoted by ROS-Industrial also supports reproducibility and scalability, allowing the methodology developed for the FANUC LR Mate 200iD/7L to be adapted to other compatible industrial manipulators.

Figure 18 – ROS-Industrial logo



Source: (ROS-Industrial, 2024).

3.2.4 Roboguide

Roboguide is FANUC's offline simulation software, intended for programming, validation, and testing of industrial robots. It allows the creation of detailed virtual models of the robot and its work environment, enabling the simulation of movements, trajectories, and operation routines without interrupting production. In addition to offline programming in Teach Pendant (TP), Roboguide provides tools for reach analysis, collision detection, and trajectory optimization (FANUC, 2025c).

4 DEVELOPMENT OF THE ROBOTIC TELEOPERATION SYSTEMS

This chapter presents the development of the robotic teleoperation system used in this work, in which ROS serves as the central communication and control framework for the FANUC LR Mate 200iD/7L manipulator. This infrastructure is responsible for motion planning, collision checking, robot state monitoring, and trajectory transmission to the industrial controller, as well as for the integration of sensors and additional components used in the experimental environment.

From this shared ROS infrastructure, two teleoperation interfaces were developed: (i) an immersive MR interface implemented in Unity and executed on the Meta Quest 3 HMD; and (ii) a conventional interface based directly on the MoveIt and RViz tools, operated using a keyboard, mouse, and 2D monitors. The following sections present the communication architecture, hardware and software integrations, safety resources, and developments carried out for each teleoperation platform.

4.1 IMMERSIVE TELEOPERATION SYSTEM ARCHITECTURE

The developed architecture consists of an enhanced variation of the proposal presented by Hernandez *et al.* (2024), preserving the master-slave model and incorporating its framing within the shared control mode (Section 2.1.3.2.2), while also expanding its capabilities, including wireless image transmission and inverse kinematics operation.

The master system is composed of the Meta Quest 3 HMD and its controllers, connected to the computer via Meta Quest Link, which enables the execution of Unity externally to the device. In Unity, the operator input signals are converted into spatial position and orientation commands for the robot and subjected to virtual constraints that delimit forbidden operating regions, restricting the transmission of actions outside the operational limits. These processed commands are then transmitted over the network to ROS through the ROS-TCP Connector.

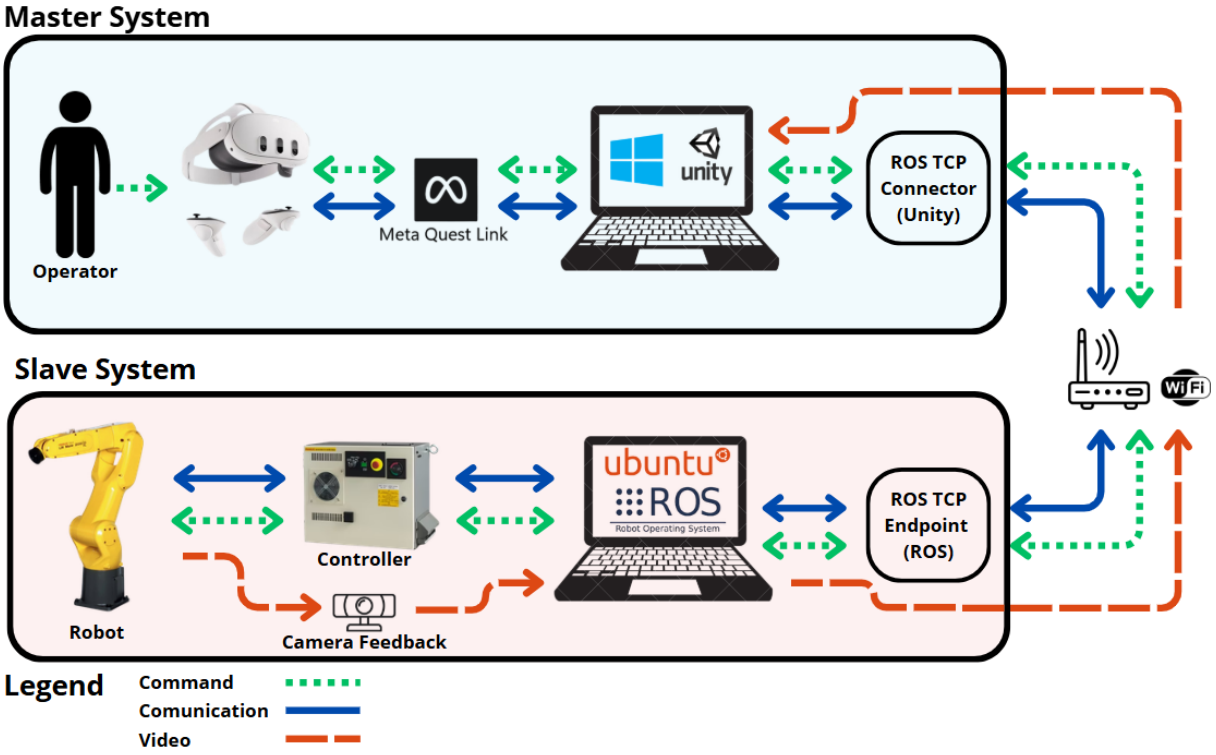
In the slave system, the ROS-TCP Endpoint receives the information coming from Unity and, through ROS and together with MoveIt, communicates with the FANUC LR Mate 200iD/7L robot controller, thus enabling trajectory planning, collision checking, and the application of kinematic and dynamic constraints, ensuring the execution of movements on the physical manipulator.

As a result, the operator commands pass through two complementary validation layers: virtual barriers in Unity (master), which perform geometric restrictions, and trajectory planning in MoveIt (slave), which evaluates kinematic and dynamic constraints. Together, these layers verify whether the commanded actions are both feasible and safe, characterizing the solution as a case of shared control.

Figure 19 presents the architecture of the immersive teleoperation system, high-

lighting the command flows (green dashed lines) and communication flows (solid blue lines), both implemented over the TCP/IP wireless network protocol (Parziale *et al.*, 2006), responsible for reliable data transmission between devices, although subject to latency variations common in teleoperation environments. Additionally, the video flow from the monitoring cameras (orange dashed lines) is transmitted through the HTTP protocol (Gourley; Totty, 2002), suitable for continuous real-time image transmission directly to the master computer.

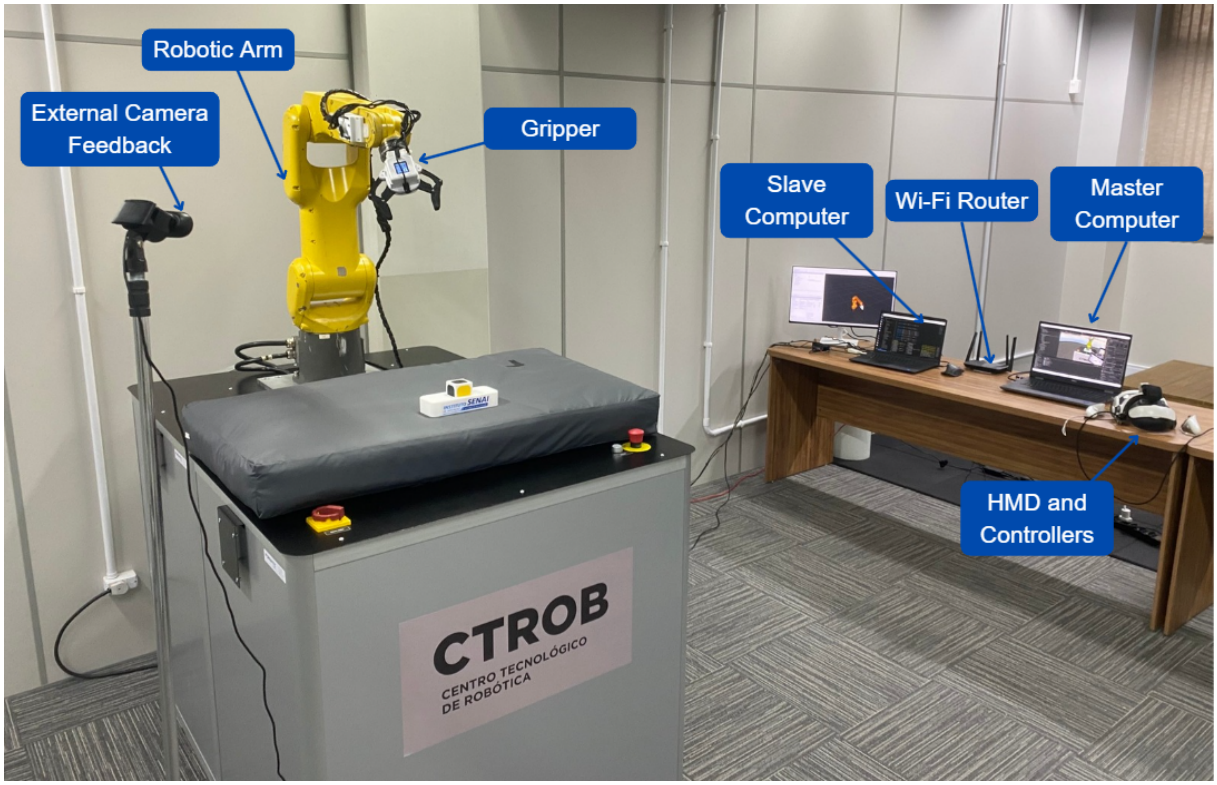
Figure 19 – Master-slave communication and command architecture of the immersive teleoperation system



Source: Author (2026).

To complement the architecture description, Figure 20 presents a real image of the immersive teleoperation physical setup, illustrating the arrangement of the equipment and its integration within the developed teleoperation environment.

Figure 20 – Physical setup of the immersive teleoperation system



Source: Author (2026).

4.1.1 Integration between systems

The developed teleoperation system is based on the integration of four main components, connected through three fundamental hardware and software interconnections:

- **HMD** $\leftrightarrow$ **Unity**: responsible for capturing the operator's movements and updating the virtual environment.
- **Unity** $\leftrightarrow$ **ROS**: allows immersive commands to be converted into robotic references in ROS.
- **ROS** $\leftrightarrow$ **FANUC**: establishes communication between the planning system and the physical manipulator.

These three interconnections form the basis of the teleoperation architecture, enabling command transmission, trajectory planning, collision checking, and the return of sensory states to the immersive environment.

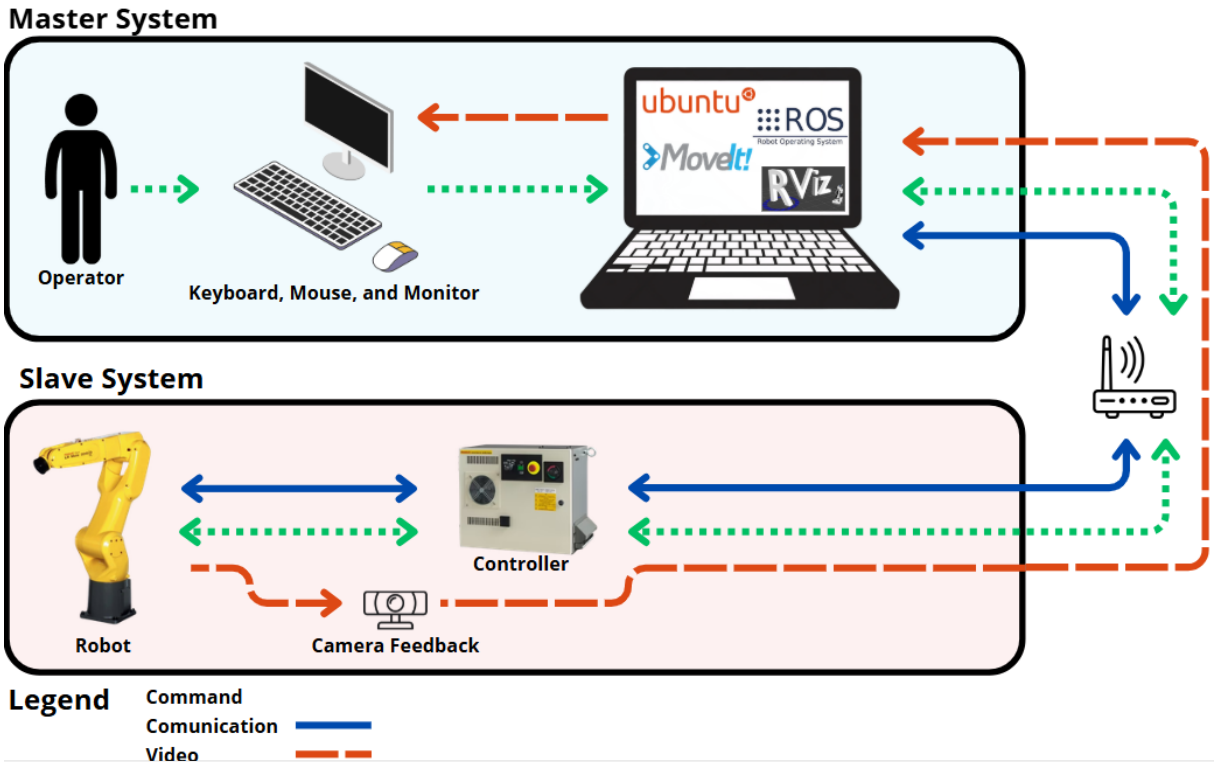
The complete configuration procedures for each stage (including installation of the Unity-ROS packages, ROS-Industrial adjustments for the FANUC, KAREL program parameterization, and initialization procedures) are detailed in Appendix C.

4.2 CONVENTIONAL TELEOPERATION SYSTEM ARCHITECTURE

The conventional teleoperation architecture was conceived as an adaptation of the immersive teleoperation system (Section 4.1), with one fundamental difference: the user interaction layer is performed directly within the ROS environment, using MoveIt and RViz for planning, monitoring, and executing manipulator actions. The master computer therefore integrates the operation interface and command processing, while the slave system consists only of the robot controller, which executes the planned movements through bidirectional communication. Manipulator control is performed through the MoveIt and RViz graphical interfaces using a keyboard and mouse, while the monitoring cameras provide a 2D view of the physical environment in real time.

Figure 21 presents the conventional teleoperation architecture, in which the operator interacts directly with the master system through a computer running ROS. Similarly to the immersive communication architecture, commands (green dashed lines) and data communication (solid blue lines) are transmitted through the TCP/IP protocol, while the monitoring cameras are directly connected to the master computer (orange dashed lines), allowing the operator to visually monitor the environment during task execution.

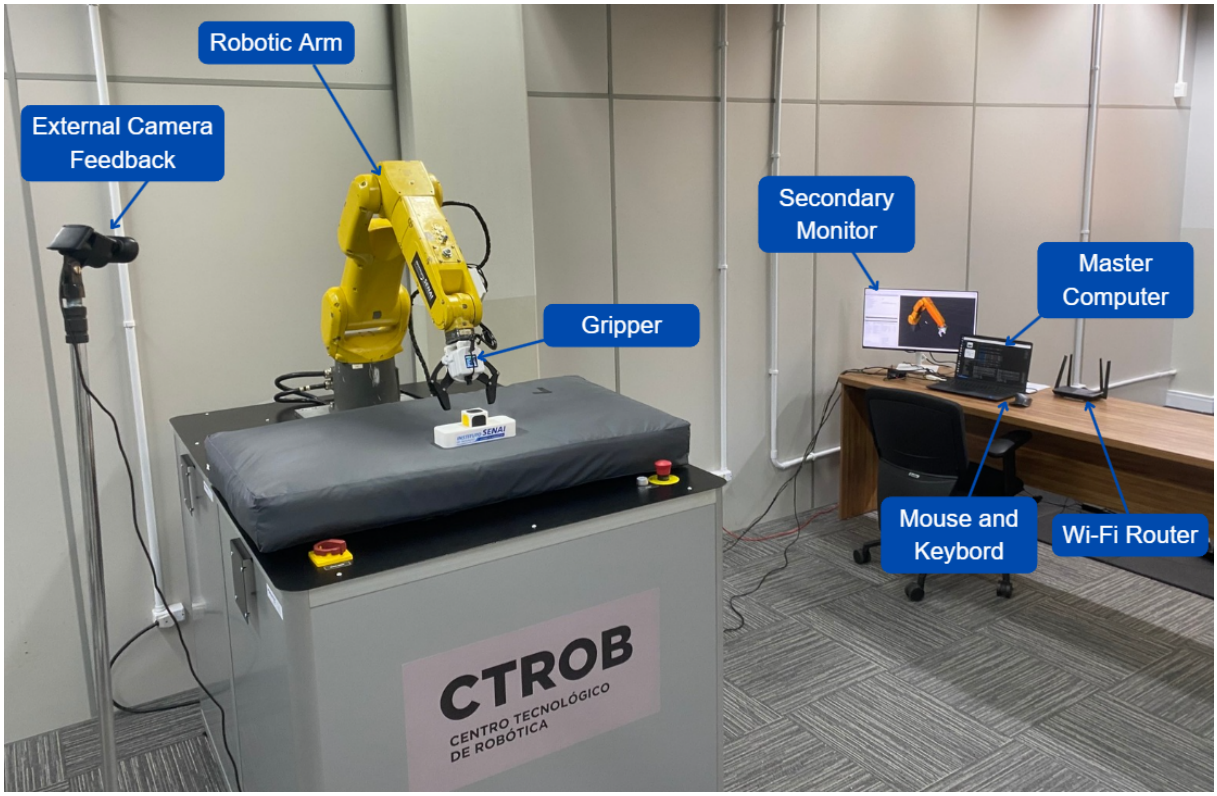
Figure 21 – Master-slave communication and command architecture of the conventional teleoperation system



Source: Author (2026).

Figure 22 shows the physical setup used in conventional teleoperation, highlighting the main equipment and its organization within the experimental environment.

Figure 22 – Physical setup of the conventional teleoperation system



Source: Author (2026).

In a typical scenario, conventional teleoperation could use two separate computers, replicating the configuration adopted in the immersive system: one dedicated to operator interaction and another responsible for trajectory processing and communication with the real robot. However, for this implementation, a unified configuration was adopted, concentrating interface and processing on a single computer. This decision simplified the operational structure and, although it reduced the maximum possible teleoperation distance due to the need for direct cable connections, it preserved all essential elements for comparison between the approaches, ensuring functional equivalence between the conventional and immersive modes.

4.3 INTEGRATION AND IMPLEMENTATION OF SYSTEM FUNCTIONALITIES

Once communication between Unity, ROS, and the FANUC robot had been established, the functionalities required for system operation were implemented. These functionalities include manipulator control, gripper operation, safety mechanisms, per-

ception resources, and the interaction interfaces used by the operator in both operating environments.

Functional equivalence between the conventional and immersive modes was maintained throughout the development process, so that the same teleoperation actions were available on both platforms. This was necessary to enable direct comparisons between the two modes during the experiments.

4.3.1 Integration of the Gripper into the Teleoperation System

The MyGripper F100 gripper has an official ROS package provided by Elephant Robotics (Elephant Robotics, 2025b), originally developed to operate only on the manufacturer's own robots or in a standalone manner. To enable its use together with the FANUC robot, adaptations to the description package and initialization structure were required. These modifications were implemented to preserve the modular structure of the project, allowing future replacements or adjustments without compromising the operation of the robot.

4.3.1.1 Geometric and functional integration

During the integration process, the system architecture was designed to avoid direct modifications to the official FANUC robot package (Appendix C), preserving compatibility with future versions of the model and maintaining the independence between the robot and gripper packages.

The objective was to allow the gripper model to be displayed and operated in an integrated manner with the robot within the ROS environment (RViz), without requiring its description to be fully incorporated into the FANUC URDF. The adopted solution was the creation of a fixed joint (*fixed joint*) connecting the last robot link (`tool0`) to the gripper base link (`gripper_base`), establishing a coupling point between the two models. Position and orientation parameters were defined based on the CAD model of the robot-gripper assembly, ensuring geometric alignment between the FANUC flange and the gripper. The definition of this fixed joint is presented in Listing 4.1.

```
<joint name="gripper_fixed_joint" type="fixed">
  <parent link="tool0"/>
  <child link="gripper_base"/>
  <origin xyz="0.0044955 0 0.041999" rpy="1.5708 0 1.5708"/>
</joint>
```

Listing 4.1 – Excerpt from the definition of the fixed joint between the FANUC robot and the F100 gripper.

Each description was maintained within its respective package, preserving the logical separation between components and allowing both systems to be loaded independently. The robot was configured to reference the gripper description during the initialization process, thereby linking both systems during initialization. This reference is defined in the main FANUC initialization file, as shown in Listing 4.2.

```
<xacro:include filename="$(find gripper_description)/urdf/force_control_gripper/
    pro_force_gripper.urdf" />
```

Listing 4.2 – Inclusion of the gripper model in the main robot URDF.

With this configuration, the gripper `.launch` file (responsible for loading the package nodes and parameters into the ROS environment) must be executed first so that its components are correctly initialized. Afterwards, the robot *launch* file is started, establishing the connection between the main robot model and the already active gripper. With this initialization sequence, ROS loads both models in parallel, allowing the robot and gripper to operate together within RViz.

4.3.1.2 Restructuring of Communication Topics and Aggregator Node

In the original version of the gripper package, it was not possible to execute it simultaneously with the FANUC robot, since both systems initialized nodes with the same names (such as *robot_state_publisher* and *joint_state_publisher*) and published messages to the same topic, `/joint_states` (responsible for continuously transmitting the position, velocity, and effort values of the robot joints within the ROS environment). This duplication generated communication conflicts, preventing the system from correctly managing both instances and compromising the combined visualization in RViz.

To solve this problem, a modular approach based on topic separation and the creation of an intermediate node was adopted. The gripper `.launch` file was modified to publish its messages to a new topic named `/joint_states_gripper`, isolating its information from the main system communication channel. Subsequently, the `joint_state_aggregator` node was developed to receive the gripper data and republish it to the global `/joint_states` topic.

This modification eliminated the communication conflicts and enabled the robot arm and gripper to be visualized and operated together within the same ROS environment, while the `joint_state_aggregator` node acts as an intermediate layer between both packages, making it possible to run them simultaneously without modifying the original FANUC package.

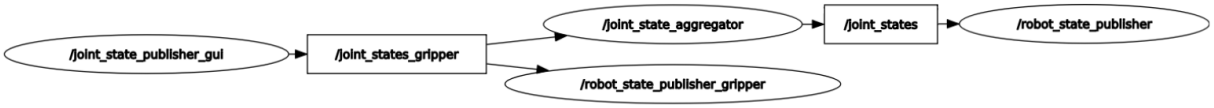
Figure 23 illustrates the difference between the original and modified gripper system architectures. The upper portion (Figure 23a) shows the original configura-

tion, in which the gripper published directly to the same topic as the robot, generating communication conflicts. The lower portion (Figure 23b) presents the modified version, where the `joint_state_aggregator` node centralizes the publications and enables both components to operate simultaneously.

Figure 23 – Comparison between the original and modified gripper system architectures



(a) Original version with topic conflicts.



(b) Modified version with the *Joint State Aggregator* node.

Source: Author (2026).

4.3.1.3 Configuration of Gripper Operating Modes

Two distinct `.launch` files were defined for gripper control, one for the conventional teleoperation mode and another for the immersive mode.

In the conventional mode, control is performed directly within ROS through a simple graphical user interface (Graphical User Interface (GUI)), which provides a slider for manually adjusting the finger opening. In the immersive mode, the opening and closing commands originate from Unity based on the operator's actions on the HMD controller and are then forwarded to ROS to actuate the physical gripper.

4.3.1.4 Rendering and control of the gripper in Unity

In the Unity environment, the gripper model was imported from the same `URDF` file used in ROS through the *URDF Importer*, and attached to the end-effector of the virtual manipulator. This attachment is performed by the `Follow_Gripper` script, which links the gripper to a reference object associated with the robot tool and periodically updates its position and orientation based on configurable *offsets* (required to compensate for the physical displacement of the gripper support relative to the origin of `link_6`). This configuration allows the gripper to rigidly follow the end-effector while maintaining geometric alignment and avoiding physics conflicts in Unity.

The opening command is generated by the `GripperUnityPublisher` script, which continuously reads the trigger value from the HMD controller, normalizes this value within a range between 0 (closed) and 1 (open), and periodically publishes a message to the `/joint_states_gripper` topic. In ROS, this topic is used to actuate the physical gripper, and its values are incorporated into the aggregated `/joint_states` topic.

Finally, the `JointStateReceiverGripper` script, also developed in Unity, subscribes to the `/joint_states` topic, maps the gripper joint names to the corresponding links of the virtual model, and updates the joint positions in real time, keeping the virtual gripper synchronized with the physical one.

4.3.2 Update of Graphical Models and Collision Structures

The original three-dimensional robot models (*meshes*) presented a low level of detail, resulting in a simplified visual representation within the RViz environment. To improve visual fidelity and spatial perception during teleoperation, the `.STL` files were replaced with more detailed versions, resulting in a more realistic representation of the manipulator within the ROS environment.

In addition to the visual update, a new simplified collision geometry was incorporated to represent the gripper. Instead of individually considering each finger, a collision box fixed to `tool0` (the virtual end-effector reference used in ROS as the standard tool attachment frame) was created, encompassing the working volume of the tool. This approach simplified the collision model and defined a safety envelope slightly larger than the physical dimensions of the gripper, providing an additional margin during trajectory planning and execution. The definition of this geometry was inserted directly into the `lrmate200id7l_macro.xacro` file, responsible for the kinematic description of the model, as illustrated in Listing 4.3.

```
<link name="${prefix}collision_box">
  <collision>
    <origin xyz="0.005 0 0.09" rpy="0 0 0"/>
    <geometry>
      <box size="0.065 0.19 0.1585"/>
    </geometry>
  </collision>
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <mass value="0.01"/>
    <inertia ixx="0.0001" ixy="0.0" ixz="0.0"
             iyy="0.0001" iyz="0.0" izz="0.0001"/>
  </inertial>
</link>

<joint name="${prefix}tool0-collision_box" type="fixed">
  <origin xyz="0 0 0" rpy="0 0 0"/>
  <parent link="${prefix}tool0"/>
  <child  link="${prefix}collision_box"/>
</joint>
```

Listing 4.3 – Definition of the collision box in the `lrmate200id7l_macro.xacro` file.

The introduction of the collision box also required modifications to the `fanuc_lrmate200id7l.srdf` file, which contains the semantic information used by MoveIt, including the *Allowed Collision Matrix*, the original gripper collision definitions were disabled, and the collision box was adopted as the collision volume associated with the tool. Part of these modifications is shown in Listing 4.4.

```
<disable_collisions link1="tool0"      link2="gripper_base"  reason="Adjacent"/>
<disable_collisions link1="tool0"      link2="collision_box" reason="attached"/>
<disable_collisions link1="link_6"     link2="collision_box" reason="adjacent"/>
<disable_collisions link1="gripper_base"   link2="collision_box"  reason="never"/>
<disable_collisions link1="gripper_left1"  link2="collision_box"  reason="never"/>
<disable_collisions link1="gripper_left2"  link2="collision_box"  reason="never"/>
<disable_collisions link1="gripper_right1" link2="collision_box"  reason="never"/>
<disable_collisions link1="gripper_right2" link2="collision_box"  reason="never"/>
```

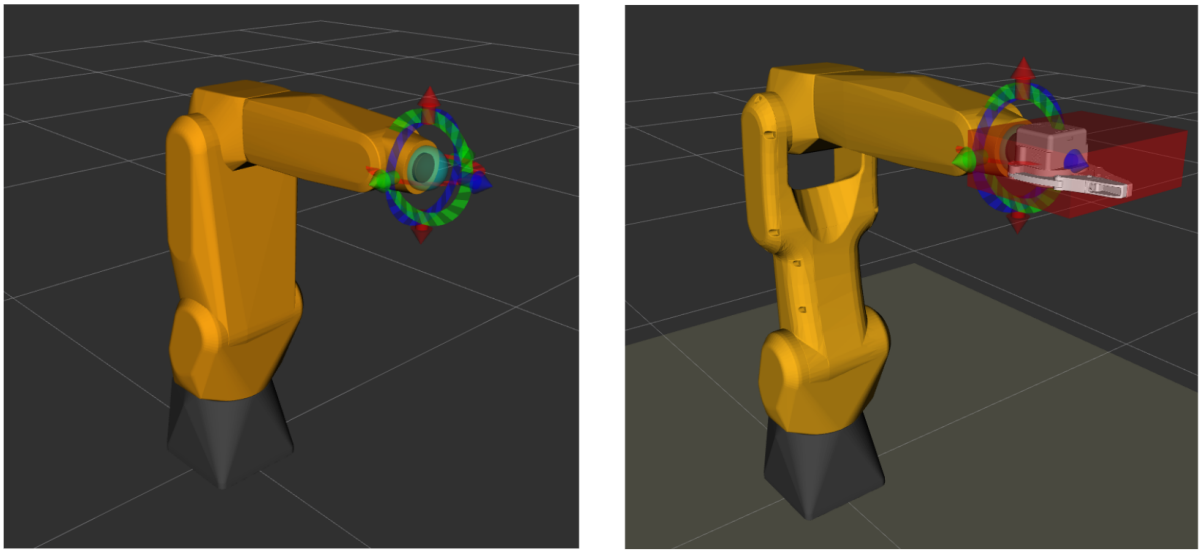
Listing 4.4 – Excerpt from the `fanuc_lrmate200id7l.srdf` file containing adjustments to the collision matrix.

These definitions limit collision checking to interactions between the robot-gripper

assembly and the external environment, avoiding unnecessary collision detections during trajectory planning.

Figure 24 compares the original and updated models. On the left, the original version is shown, using simplified meshes. On the right, the updated version can be seen, with more detailed models and the collision box added to the gripper.

Figure 24 – Comparison between the original and updated robot and gripper models



(a) Original simplified *meshes*.

(b) Updated version with new *meshes* and the gripper collision box.

Source: Author (2026).

4.3.3 Structure and Integration of Virtual Robotic Models

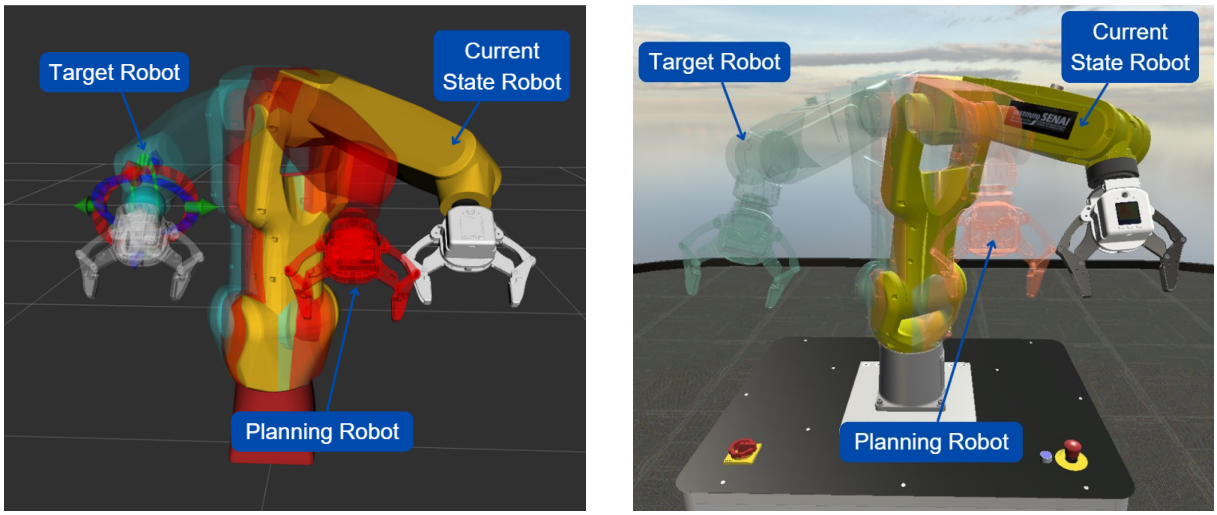
To keep the same resources available in both teleoperation modes, the Unity environment was developed to reproduce the robot visualizations used in RViz and MoveIt. In the ROS environment, the robotic scene is composed of three manipulator representations:

- Target Robot;
- Planning Robot;
- Current State Robot.

The same structure was implemented in Unity and synchronized with ROS, allowing the operator to access the same robot information in both environments. This approach helped ensure that differences between the teleoperation modes were related to the interaction method rather than to the available visualization resources.

Figure 25 shows the robotic representations in ROS and Unity. On the left, the RViz environment displays the Target Robot, Planning Robot, and Current State Robot. On the right, the equivalent implementation in Unity is shown, reproducing the same visual references within the immersive environment.

Figure 25 – Comparison between the robotic representations in the ROS and Unity environments



(a) Representation of the three robots in RViz/-MoveIt.

(b) Equivalent representation developed in Unity.

Source: Author (2026).

4.3.3.1 Target robot

The target robot represents the model directly controlled by the operator in the virtual environment, functioning as the main interface for defining the desired pose of the manipulator tool. Unlike the other models, which were imported from the FANUC URDF description, the target robot was built directly in Unity using the *RobotDynamicsLibrary* (EricVoll, 2021), modified to meet the specific requirements of this teleoperation system. This C# library includes the structures required to define kinematic chains and perform inverse kinematics calculations for robotic manipulators.

The robot structure is defined in the `RevoluteRobot` script, which creates the kinematic chain through the sequential addition of revolute joints and links, as illustrated in Listing 4.5.

```
Robot Ro = new Robot()
    .AddJoint('z', new Vector(0, 0, 0.17))
    .AddJoint('y', new Vector(-0.0500, 0, 0.160))
    .AddJoint('y', new Vector(0, 0, 0.4400))
    .AddJoint('x', new Vector(-0.08850, 0, 0.0350))
    .AddJoint('y', new Vector(-0.331500, 0, 0))
    .AddJoint('x', new Vector(-0.0800, 0, 0));

Links = Ro.Links;
Robot = Ro;
```

Listing 4.5 – Definition of the target robot kinematic chain in Unity.

Each call to the `AddJoint()` method adds a rotational joint around the specified axis (`'x'`, `'y'` or `'z'`) and defines the relative displacement of the next link in space, based on the geometric parameters of the FANUC LR Mate 200iD7L. This configuration allows the Unity model to maintain the same geometric structure and the same degrees of freedom as the physical manipulator, as presented in Appendix A, which details the robot dimensions.

The definition of the limits and initial values of each joint is performed in the `RobotBase` class, responsible for managing the manipulator joint states. This script is used to perform the forward and inverse kinematics calculations, storing parameters such as angular limits, number of links, convergence tolerance, and maximum algorithm iterations. Listing 4.6 exemplifies the configuration of the joint limits in radians, ensuring that the movements of the virtual model remain compatible with the mechanical constraints of the real robot:

```
jointLimits = new List<JointLimits>
{
    new JointLimits(-2.96706, 2.96706),
    new JointLimits(-2.53073, 1.74533),
    new JointLimits(-1.13446,  3.71683),
    new JointLimits(-3.316125, 3.316125),
    new JointLimits(-1.8326, 1.8326),
    new JointLimits(-2.96706, 2.96706)
};
```

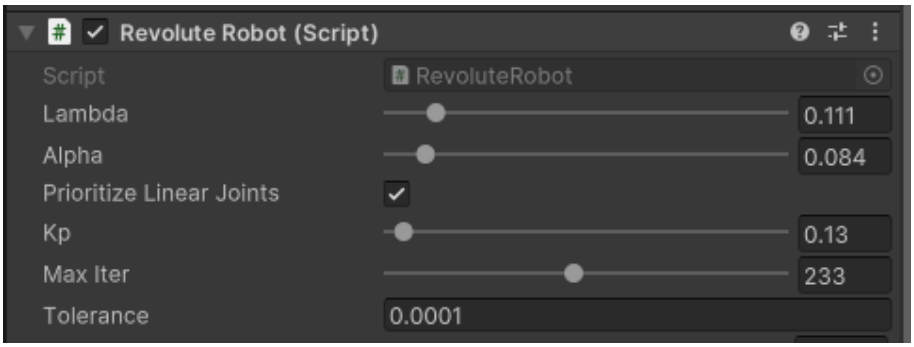
Listing 4.6 – Definition of the joint limits in the `RobotBase` script.

The values of these joints and their limits are detailed later in subsection 4.3.9, where the adopted modifications and restrictions applied to each manipulator axis are

discussed.

The robot movement is based on numerical inverse kinematics using the damped least squares method, also implemented in the `RobotBase` class. This method was adopted because it reduces instabilities near singular configurations and improves convergence during the inverse kinematics calculations (Wampler, 1986; Siciliano *et al.*, 2009). In the `RevoluteRobot` component (Figure 26), several parameters can be adjusted to calibrate the algorithm behavior: the `Lambda` term defines the damping factor; `Alpha` controls the angle update step; `Kp` acts as a proportional gain for smoothing; `Max Iter` limits the number of iterations; and `Tolerance` establishes the maximum admissible error between the desired pose and the pose reached by the end-effector.

Figure 26 – Configuration parameters of the `RevoluteRobot` component in Unity, used in the inverse kinematics calculation of the target robot



Source: Author (2026).

During immersive teleoperation, the operator interacts with the robot through the HMD controllers, and when the target robot is manipulated, the controller movements are mapped to the end-effector pose, with numerical inverse kinematics being continuously applied in Unity to update the virtual manipulator joints in real time. The target robot acts only as a visual reference and is not directly executed by the physical robot, allowing the operator to define and evaluate the desired robot configuration before trajectory planning and execution. Additional details about the operation of the *grip* button and the control actions are presented in Section 4.3.7.

The position and orientation of the virtual gripper are continuously published to ROS through the `unity_interactive_marker_feedback_node.py` node, which converts the messages received from the `/unity_pose` topic into the *InteractiveMarkerFeedback* format used by MoveIt, allowing the interactive marker displayed in RViz (Figure 25a) to follow the movements performed in Unity in real time. Unity transmits only the end-effector pose, while MoveIt uses this information to perform the inverse kinematics and trajectory planning calculations.

4.3.3.2 Planning robot

The planning robot is used to represent the trajectories generated by MoveIt during the motion planning process, allowing the operator to preview the path that will be executed by the physical robot. In the conventional teleoperation environment, this same visualization is provided by the planning robot displayed in RViz. To preserve this visualization approach in the immersive environment, an equivalent planning robot was implemented in Unity.

The model was imported into Unity using the *URDF Importer*, preserving the kinematic structure of the FANUC manipulator, and during normal operation it remains hidden, being displayed only when a new trajectory is planned. Whenever the operator requests a movement, Unity receives the messages published on the `/move_group/display_planned_path` topic, which contain the sequence of joint positions generated by MoveIt during the planning process.

These messages are processed by the `PlannedPathReceiver` script, which reads the joint positions and reproduces the planned motion in Unity, converting the received values into an animation of the manipulator and updating the robot configuration throughout the trajectory. Once the animation is completed, the planning robot is hidden again, leaving only the target robot and the current state robot visible in the scene.

In this way, the operator can inspect the trajectory generated by MoveIt directly within the immersive environment, using the same planning reference available in RViz before sending the movement to the physical robot.

4.3.3.3 Current state robot

The current state robot corresponds to the visual representation of the physical manipulator, updated in real time from the joint data received from ROS. In the conventional teleoperation environment, this information is displayed in RViz through the robot model synchronized with the physical manipulator. To provide the same reference in the immersive environment, an equivalent current state robot was implemented in Unity.

Similarly to the planning robot, the model was imported into Unity through the *URDF Importer*, preserving the original kinematic structure and geometric characteristics. The synchronization between the physical robot and the virtual model is performed by the `JointStateReceiver` script, developed in Unity, which subscribes to the `/joint_states` topic and continuously interprets the messages published by ROS. Each received message contains the joint names and their angular positions in radians; the script maps these names to the corresponding links in Unity, converts the values into degrees, and updates the respective articulation components of the model frame by frame. In this way, the movement of the physical robot is continuously reproduced in the virtual environment.

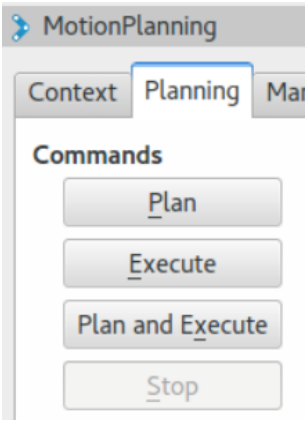
Visually, the current state robot is displayed using the colors of the real robot, differentiating it from the planning robot (red) and the target robot (blue). This color coding facilitates the operator understanding of the overall system state, allowing the distinction between what is being executed on the real robot, what has been planned, and what is being defined as the target for the next movement.

4.3.4 Control modes

Two control modes were implemented in the teleoperation system: the *planning and execution* mode and the *automatic mode*. Both modes rely on the same ROS and MoveIt infrastructure and use the same topics and services, differing only in how planning and execution commands are triggered in the conventional and immersive environments.

Figure 27 shows the standard MoveIt *Plan*, *Execute*, and *Plan Execute* buttons, whose operation served as the reference for the implementation of these control modes.

Figure 27 – Standard MoveIt planning and execution buttons



Source: Author (2026).

4.3.4.1 Planning and execution mode

This mode follows the traditional MoveIt workflow, in which the operator defines a pose for the target robot and separately triggers the trajectory planning and execution stages. In the conventional environment, the operator uses the *Plan* and *Execute* buttons of the MoveIt interface to generate and validate motion plans, following the standard trajectory planning and execution procedure adopted by MoveIt.

In the immersive environment, the same behavior was adapted to the MR context through two ROS services (request and response mechanisms used for direct communication between nodes) implemented in the `fanuc_moveit_bridge.py` node, developed to integrate MoveIt with the teleoperation system. In this architecture, the

`/plan_trajectory` service generates the motion plan in MoveIt based on the received pose, while the `/execute_trajectory` service sends the most recent plan to the robot controller for execution.

These services are triggered from Unity through two C# scripts: `ButtonAPlan`, associated with the HMD controller button A, which sends the planning command, and `ButtonBExecute`, associated with button B, which executes the planned movement.

4.3.4.2 Automatic mode

As an alternative to manual step-by-step execution, the automatic mode was developed, in which the system performs the *planning and execution* cycle sequentially and autonomously. This mode is particularly useful for short movements or tasks that require small pose corrections, reducing the need for separate manual commands for planning and execution and making teleoperation more continuous.

In the conventional environment, the automatic mode is executed directly in ROS through the `fanuc_moveit_bridge.py` node. This node continuously monitors updates to the gripper position and orientation, comparing each newly received pose with the last executed trajectory. When a pose change is detected, the system automatically triggers the `group.go()` command, initiating a new planning and execution sequence without requiring operator intervention. This behavior can be dynamically enabled or disabled through the `/set_auto_mode` service, accessible through a simple graphical interface described in greater detail in Section 4.3.7.

In the immersive environment, the automatic mode is controlled by the `AutoModeToggle` script, associated with the HMD controller button Y, which switches between manual and automatic modes. When automatic mode is activated, the script sends commands to the `/set_auto_mode` ROS service, reproducing in the immersive environment the same behavior implemented in the conventional system. During its activation, the `ButtonAPlan` and `ButtonBExecute` scripts are temporarily disabled, preventing interference from manual commands.

4.3.5 Safety Mechanisms and Collision Prevention

Collision prevention in the system was structured into two complementary layers: (i) early verification in the Unity environment, which acts preventively by blocking the transmission of potentially unsafe commands; and (ii) geometric verification in MoveIt, which validates whether the planned trajectory is physically safe before execution.

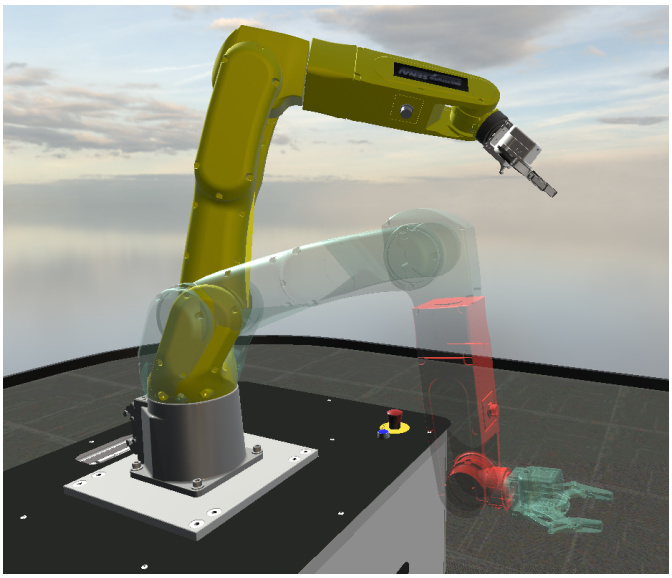
4.3.5.1 Unity layer: preventive blocking and visual signaling

In the immersive environment, the target robot and its gripper use the `ChangeColorOnCollision` and `ChangeColorOnCollision_garra` scripts, responsible for identifying

intersections between the robot and scene objects. When a collision occurs, the original material of the model is replaced with opaque red and the public flag `isColliding` is activated, as shown in Figure 28. Once contact ceases, the original material is restored and the flag returns to `false`.

These flags are checked by the `ButtonAPlan`, `ButtonBExecute`, and `AutoModeToggle` scripts before any request is sent to the ROS services `/plan_trajectory`, `/execute_trajectory`, or `/set_auto_mode`. If a collision is active, the command is blocked and the console displays a warning, preventing the execution of unsafe movements.

Figure 28 – Color change of the target robot in the Unity environment during collision detection



Source: Author (2026).

This layer highlights collisions in the immersive scene and blocks commands associated with colliding configurations before they are transmitted to ROS.

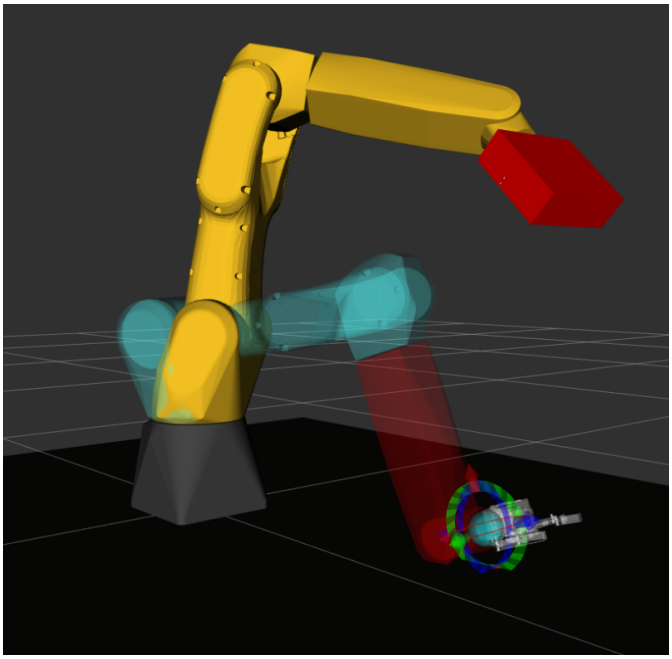
4.3.5.2 ROS/MoveIt layer: geometric models and collision matrix

At the planning level, MoveIt performs collision checking based on the volumes defined in the URDF and the permissions configured in the `.srdf` file. This behavior is native to MoveIt and ensures that each trajectory is geometrically validated before being sent to the robot controller. When the planner detects a collision, the involved elements are highlighted in red in the RViz interface (Figure 29), clearly indicating to the operator that the proposed pose or path is invalid.

The gripper uses a dedicated *collision box*, detailed in Section 4.3.2. This simplified representation provides an additional safety margin for the end-effector. The

Allowed Collision Matrix was configured only to permit unavoidable contacts between adjacent links, following the standard MoveIt configuration.

Figure 29 – Example of collision detection by MoveIt in the RViz interface



Source: Author (2026).

4.3.5.3 Virtual protection barrier

In addition to the previous verification layers, the system includes a virtual safety barrier positioned within the workspace region. This barrier was implemented as an additional collider in Unity and as a collision volume in MoveIt, acting as the lower boundary of the manipulator operational space.

The barrier is not rendered to the operator, keeping the environment visually clean and simulating a realistic teleoperation scenario in which physical elements are not mapped or rendered in the virtual model. Despite being invisible, the collider prevents the gripper from exceeding the lower limit and colliding with the table surface.

When an intersection with this barrier occurs, Unity blocks the transmission of new commands, while MoveIt rejects trajectories that violate this limit, preventing movements below the workspace boundary from being planned or executed.

4.3.6 Perception and Sensory Feedback

During teleoperation, the operator relies on information from the physical environment to monitor the robot, evaluate distances, and perform manipulation tasks, for this purpose, the system incorporates monitoring cameras, distance sensing, and visual

and auditory feedback resources, distributed between the ROS and Unity environments, to support these activities and provide additional references during operation.

4.3.6.1 Monitoring cameras

The monitoring cameras, described in Section 3.1.3, were integrated into the system to provide real-time visualization of the physical environment during teleoperation. Two cameras are used in the experimental setup: one fixed camera, positioned to provide an overall view of the workspace, and another attached directly to the robot gripper, providing a closer view of the manipulation area and the interaction with objects.

The captured images are made available through *MJPEG-Streamer* (Tom Stöveken, 2021), a server widely used in robotic applications for HTTP video streaming. Execution is performed from the Linux terminal using the command presented in Listing 4.7

```
./mjpg_streamer -i "input_uvc.so -d /dev/video0 -r 1280x720 -f 30" -o "output_http.so -
    w ./www -p 8080"
```

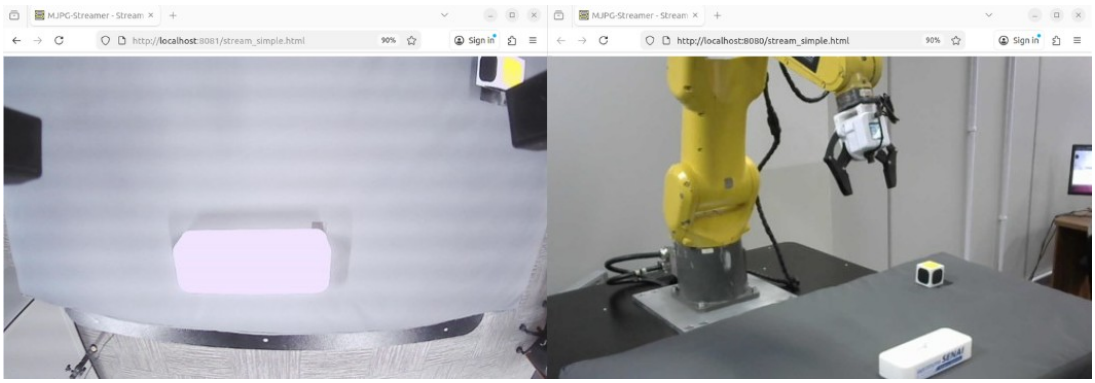
Listing 4.7 – Initialization of the HTTP video server using MJPEG-Streamer.

The "`-i`" parameter defines the USB capture module (`input_uvc.so`), associated with device `/dev/video0`, with a resolution of 1280×720 pixels at 30 FPS, while the "`-o`" parameter creates the HTTP server on port `8080`. The stream can be accessed through the address `http://IP_HOST:8080/?action=stream`, compatible with browsers and external applications.

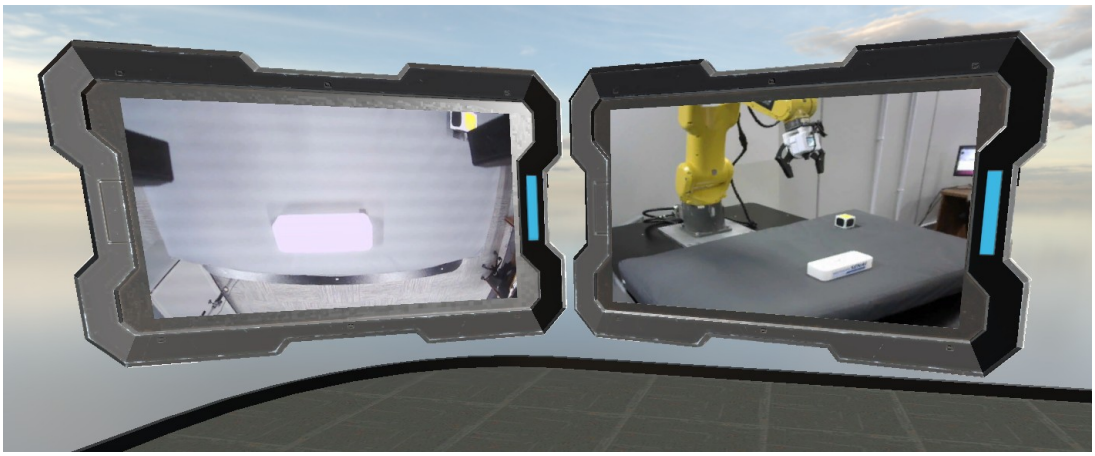
In the conventional mode, visualization of the physical environment is performed through the continuous MJPEG stream transmitted by the server, allowing the operator to monitor task execution directly through a web browser, with a continuous view of the workspace and robot behavior. In the immersive mode, the same server is accessed using a different strategy, where Unity employs the script `StreamReceiver` to capture individual frames at intervals of approximately 100 ms (adjustable according to the application requirements). This approach provides near real-time updates while avoiding excessive network and graphical processing load.

Figure 30 illustrates the two visualization approaches implemented for the camera monitoring system, showing the video transmission in the conventional mode and its equivalent representation in the immersive environment.

Figure 30 – Comparison between the camera visualizations in the conventional and immersive modes



(a) Camera visualization in the conventional mode.



(b) Camera visualization in the immersive mode.

Source: Author (2026).

4.3.6.2 Laser distance sensor

The TOF10120 optical sensor, installed at the end of the gripper and connected to the system through an Arduino Uno microcontroller, continuously measures the distance between the gripper and nearby objects. Data acquisition is performed through the I^2C interface (digital communication channel between the sensor and the Arduino), and the obtained values are sent to ROS through the `/tof_distance` topic using the *rosserial* library (which enables message exchange between the Arduino and ROS). In the `Laser.ino` code, executed on the Arduino, a moving average filter was also implemented to reduce small measurement variations and provide greater stability, with an approximate update rate of 10 Hz.

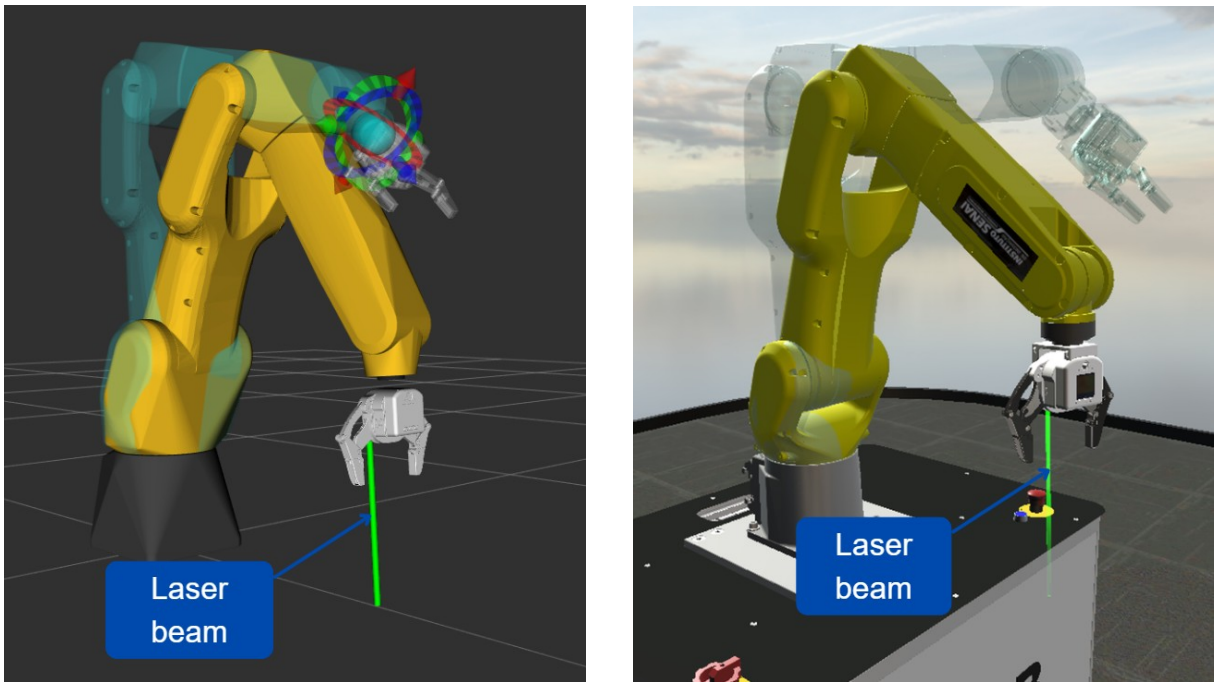
In the ROS environment, the sensor readings are interpreted and visually displayed in RViz by the `tof_tool0_marker.py` node, launched through the `tof_tool0_marker.launch` file. This node creates a cylindrical marker attached to the `tool0` frame, whose height is proportional to the value measured by the sensor. The `.launch` file

allows adjustment of parameters such as `offset_z`, used to correct the distance between the sensor and the gripper reference frame, and `radius`, which defines the beam thickness. In this way, the visual representation can be calibrated without requiring modifications to the URDF model. The marker color changes automatically according to the measured distance: when the object is outside the grasping region, the beam is displayed in green, indicating a safe distance; as the object approaches the configured manipulation range, the marker changes to red, indicating that the object is within the grasping region. This color variation provides the operator with a visual indication of proximity to objects or potential collisions during teleoperation.

In the MR environment, the same behavior is reproduced in Unity through the `LaserMarker` script, which renders the laser beam based on the data received from ROS. The script adjusts the dimensions and position of a cylinder aligned with the gripper Z-axis, using a reference point fixed to the sensor support. A minimum variation threshold is applied to avoid small visual fluctuations, and the beam material changes color according to the same criteria adopted in the ROS environment, reproducing the sensor indication in the virtual model.

Figure 31 illustrates the two forms of representation of the laser distance sensor in the ROS and MR environments, highlighting the visual correspondence between the beam projected in *RViz* and the one rendered in Unity.

Figure 31 – Comparison between the laser distance sensor representations in the ROS and MR environments



(a) Laser beam visualization in RViz. (b) Laser beam visualization in Unity.

Source: Author (2026).

4.3.6.3 Visual and auditory feedback

In addition to the physical sensors and monitoring cameras, the system incorporates a visual and auditory feedback layer responsible for translating robot events into perceptible responses for the operator. This layer acts as a link between environment perception and user interaction, reinforcing situational awareness during teleoperation.

In the conventional mode, the feedback is predominantly visual, presented in the RViz interface through status messages, collision indicators, and color changes in the robot models. The color variations associated with the target robot, used to indicate collisions or invalid trajectories, were detailed in Section 4.3.5. These visual elements assist the operator in interpreting the system state and validating planning and execution actions.

In the immersive mode, visual and auditory feedback are integrated into the Unity environment. Contextual tips displayed directly on the Meta Quest 3 controllers indicate the available commands and their current status, while audio cues are used to signal events such as switching between control modes. The auditory feedback was introduced to provide additional confirmation of user actions, since interaction in the immersive environment does not include the mouse clicks and other interface sounds naturally present in the conventional mode.

4.3.7 Operator interaction

Although the conventional and immersive teleoperation modes provide the same main functionalities, they differ in the interaction devices and visualization resources used by the operator. In the conventional mode, interaction takes place directly through the RViz interface, using a keyboard, mouse, and 2D monitor, whereas in the immersive mode the operator interacts through the Meta Quest 3 controllers and tracking system, manipulating the robot within a three-dimensional MR environment.

4.3.7.1 Interaction in the conventional environment

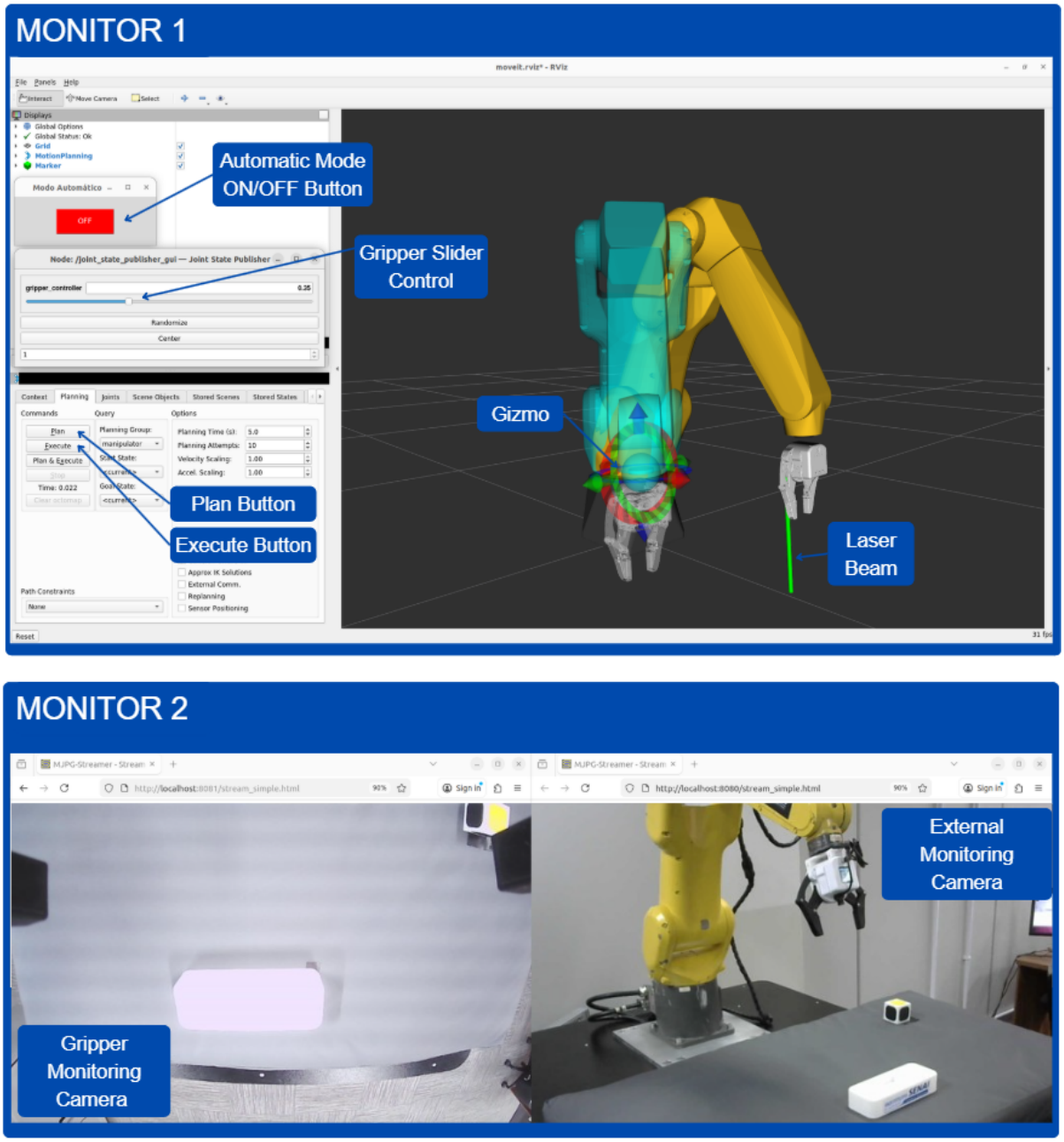
In the conventional mode, operator interaction takes place mainly through RViz, supported by the MoveIt tools and the monitoring camera images displayed on a dedicated 2D monitor, as described in Section 3.1.3. In this environment, the operator uses the mouse to manipulate the robot interactive marker (*gizmo*), represented by translation and rotation axes, allowing the desired position and orientation of the end-effector to be defined intuitively.

From this interface, the operator can switch between the control modes described in Section 4.3.4, choosing between manual planning and execution or the automatic mode according to the task requirements. Gripper control is performed separately through a *slider*, which continuously adjusts the opening distance of the fingers.

In addition to the visual monitoring provided by the cameras, the system also displays in RViz the laser beam generated by the sensor attached to the gripper, detailed in Section 4.3.6.2.

Figure 32 illustrates the conventional teleoperation interface, including the interactive marker, control tools, monitoring cameras, and laser sensor visualization used during the experiments.

Figure 32 – Conventional environment interface



Source: Author (2026).

4.3.7.2 Interaction in the immersive environment

In the immersive environment, the operator enters a real-scale three-dimensional scene, visualizing the robot and its surroundings directly in MR. Through the HMD, the operator can approach the manipulator, change the point of view, and walk around the workspace, selecting the most suitable perspective for task execution. The monitoring camera images described in Section 3.1.3, are displayed on two floating virtual screens that can be repositioned and oriented freely by the operator during teleoperation.

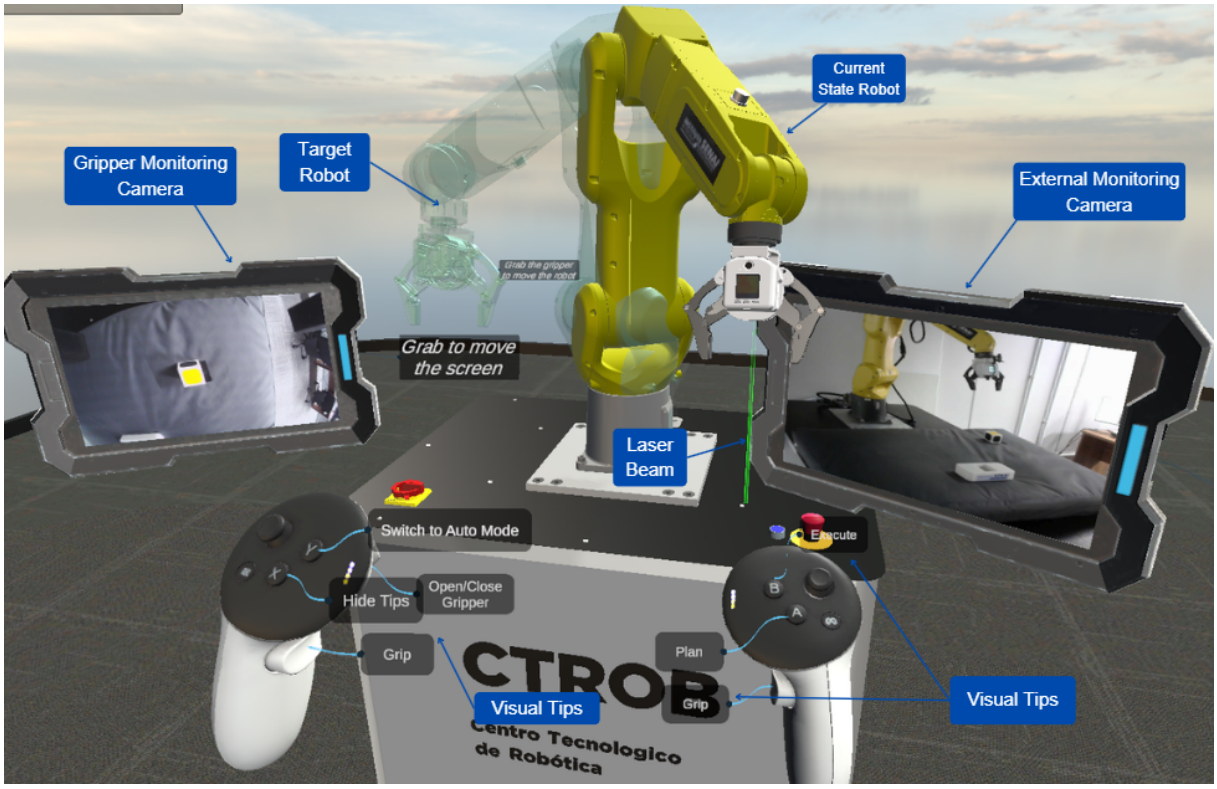
The desired pose is defined through the virtual gripper of the target robot. To initiate control, the operator brings the controller close to the tool region and presses the *grip* button, virtually grasping the robot end-effector. While the *grip* button remains pressed, the controller translation and rotation movements are transferred to the target robot in real time, allowing the gripper pose to be adjusted directly within the immersive environment.

Planning and execution commands were also integrated into the HMD controllers. Specific buttons are assigned to trigger trajectory planning (*Plan*), execute a previously calculated plan (*Execute*), and directly use the combined (*Plan Execute*) function, according to the control modes presented in Section 4.3.4. Another button is used to enable or disable the automatic mode, while the *Hide Tips* function allows the visual indications associated with each controller button to be shown or hidden after the user becomes familiar with the command mapping.

Gripper control follows the same interaction logic adopted in the conventional mode, with the operator using the controller *trigger* to continuously adjust the finger opening, closing the gripper when pressing the trigger and opening it when releasing it, while the corresponding motion is updated in both the virtual model and the physical robot. The immersive interface also incorporates the laser beam visualization attached to the gripper, together with the visual and auditory feedback resources described in Section 4.3.6.

Figure 33 presents the immersive teleoperation interface, including the virtual robots, floating camera screens, controller interactions, and feedback elements available to the operator.

Figure 33 – Immersive environment interface



Source: Author (2026).

4.3.7.3 Comparative analysis between the interaction environments

Table 6 summarizes the main functions available in each teleoperation mode and their respective interaction mechanisms. Although the interaction devices differ between the conventional and immersive environments, the same core functionalities were maintained in both interfaces, allowing the comparison to focus on the interaction method adopted in each system.

Table 6 – Comparison between interaction mechanisms in the conventional and immersive modes

Function	Conventional Environment	Immersive Environment
Target pose definition	Manipulation of the RViz *gizmo* using the mouse	Adjustment of the 3D *gizmo* through the controller (*grip* button)
Trajectory planning	*Plan* button in the MoveIt panel	*A* button on the controller
Movement execution	*Execute* button in the MoveIt panel	*B* button on the controller
Mode switching (manual/automatic)	On/Off button in the GUI	*Y* button on the controller
Gripper control	GUI *slider*	Progressive pressure on the right *trigger*
Robot motion monitoring	Visual feedback in RViz (3D model)	Direct observation of the robot at real scale in the MR scene
Monitoring camera visualization	External 2D monitor (separate window)	Interactive floating screens integrated into the scene

Source: Author (2026).

4.3.8 Communication Latency

The latency of the teleoperation systems was evaluated by considering the different stages involved in command transmission and execution.

For the immersive teleoperation system, two latency components were measured. The first corresponds to the communication between Unity and the ROS/MoveIt framework through the ROS-TCP Connector. This value was obtained from the time interval between the execution command generated in Unity and its reception by the MoveIt execution service, resulting in an average latency of approximately 12 ms.

The second component corresponds to the interval between the reception of the execution command by MoveIt and the moment at which the physical robot begins to move. Experimental measurements indicated an average latency of approximately 330 ms for this stage, mainly associated with trajectory processing and communication with the robot controller.

Considering both stages, the overall latency of the MR teleoperation system was approximately 342 ms between the execution command issued by the operator and the beginning of robot motion.

In the conventional teleoperation system, the Unity and ROS-TCP Connector communication layer is absent. Consequently, the latency is limited to trajectory processing and robot response, resulting in an average delay of approximately 330 ms.

Although the MR architecture introduces an additional delay of approximately 12 ms, this value represents less than 4% of the total measured latency. Furthermore, the overall response time remains within the limits commonly reported in the teleoperation literature. According to Farajiparvar, Ying, and Pandya (2020), delays below approximately 400 ms generally do not lead to significant degradation in operator performance, whereas larger delays tend to have a more noticeable impact on usability and task execution.

4.3.9 Movement Limits and Mechanical Adjustments

To preserve the integrity of the physical components, the movement limits of the manipulator were redefined. These modifications became necessary due to the installation of the gripper and the Arduino module, whose power and communication cables were routed externally along the robotic arm (Figure 34), consequently, part of the original joint range had to be restricted.

Figure 34 – Cable routing and peripheral integration



Source: Author (2026).

The new limits were applied at three levels of the system: (i) in the Unity environment, through the `jointLimits` structure of the `RobotBase` script; (ii) in ROS, through the `URDF` and `MoveIt` configuration files; and (iii) directly in the FANUC controller, ensuring that all control layers shared the same operational constraints.

Table 7 presents a comparison between the original FANUC joint limits and the adjusted limits adopted in the developed teleoperation system.

Table 7 – Comparison between original and adjusted joint limits of the FANUC LR Mate 200iD/7L

Joint	**Original Limits**		**Adjusted Limits**	
	Min.	Max.	Min.	Max.
J1	-170°	$+170^\circ$	-170°	$+170^\circ$
J2	-145°	$+100^\circ$	-145°	$+100^\circ$
J3	-70°	$+213^\circ$	-65°	$+213^\circ$
J4	-190°	$+190^\circ$	-190°	$+190^\circ$
J5	-125°	$+125^\circ$	-105°	$+105^\circ$
J6	-360°	$+360^\circ$	-170°	$+170^\circ$

Source: Author (2026).

4.3.10 Availability and Reproducibility

To support the reproducibility of the experiments and the reuse of the proposed system, all developed files were organized and made publicly available in two dedicated repositories:

- **Unity repository:**
  `https://github.com/matheusnicolashernandez/FANUC_MR_TELEOPERATION_UNITY`

  This repository contains the C# scripts, interactive objects, 3D models, and the complete scene used with the Meta Quest 3 headset.

- **ROS repository:**
  `https://github.com/matheusnicolashernandez/FANUC_MR_TELEOPERATION_ROS`

  This repository includes the FANUC robot configuration packages, gripper control scripts, auxiliary nodes developed in Python, and Arduino code. The entire ROS structure was encapsulated in a Docker container already configured with the required dependencies, simplifying installation and execution.

The repository structures and usage instructions allow users with basic knowledge of ROS, Unity, and Docker to fully reproduce the teleoperation system implemented in this work.

To assist the initialization process, the complete procedures for launching the ROS and Unity environments are described in Appendices E and F.

5 COMPARATIVE STUDY

This chapter presents the experimental design used to compare immersive teleoperation with the conventional interface. It describes the participants, the experimental task, the adopted procedures, and the evaluation metrics that structure the study.

5.1 STUDY CHARACTERIZATION

This work is characterized as a comparative experimental study involving two teleoperation systems: an immersive interface (Section 4.1) and a conventional interface (Section 4.2), both evaluated under controlled conditions. The independent variable corresponds to the teleoperation interface, while the dependent variables are related to participant performance and user experience (Gil, 2008).

From a usability perspective, experimental evaluations can be classified as formative or summative (Sauro; Lewis, James R., 2016). Formative evaluations focus on identifying interaction issues and supporting iterative refinement, whereas summative evaluations aim to compare system performance through quantitative and qualitative metrics. The present study is predominantly summative, as it focuses on comparing participant performance and perception between the two teleoperation approaches.

5.2 OBJECTIVE OF THE EXPERIMENTAL STUDY

This study aimed to compare performance and user experience between immersive teleoperation and the conventional interface using quantitative metrics and subjective evaluations. The analysis considered indicators related to execution time, success rate, usability, workload, and user perception during task execution.

5.3 PARTICIPANTS

The definition of the number of participants in usability studies is directly related to the objectives of the evaluation. In summative and comparative studies such as the present work, the focus lies on controlled comparisons between experimental conditions rather than statistical generalization to a broader population (Albert; Tullis, 2022).

Sauro and James R. Lewis (2016) indicate that samples of around 10 participants are often sufficient to identify most usability issues in well-defined tasks, in addition to allowing the collection of relevant quantitative metrics. However, Alroobaea and Mayhew (2014) emphasize that there is no universally appropriate number of participants, since the sample size depends on factors such as system complexity, task scope, and user variability. Complementarily, Macefield (2009) indicate that usability studies may adopt ranges between 3 and 20 participants for problem identification and between 8 and

25 participants in comparative studies, with 10 to 12 participants being a common reference.

Furthermore, sample representativeness is more relevant than its absolute size, and a non-representative sample cannot be compensated for through statistical techniques (Sauro; Lewis, James R., 2016).

Based on these criteria, 14 participants took part in the study, a sample size that falls within the ranges commonly reported for comparative usability evaluations in the literature.

5.3.1 Selection Criteria

Participants were intentionally selected from individuals with engineering backgrounds and experience related to robotics, including engineers, master's graduates, and PhD holders. This selection ensured that all participants possessed sufficient technical knowledge to understand the operational principles, control mechanisms, and human–robot interaction aspects involved in the evaluated interfaces.

According to Hernandez *et al.* (2025), heterogeneous user samples may be appropriate for studies focused on general technology acceptance. However, evaluations intended to identify technical limitations and usability issues benefit from participants whose profiles are closer to those of the target users (Kujala, 2003). For this reason, only participants with engineering-related backgrounds and prior knowledge of robotics were included in the study.

The same study also reports that engineering professionals tend to provide more analytical and technically detailed feedback, which can contribute to a more precise assessment of interaction characteristics and operational limitations.

To minimize potential biases associated with familiarity, participants without significant prior experience with the evaluated interfaces were prioritized. This helped ensure that the obtained results reflected the characteristics of the teleoperation approaches themselves rather than previous exposure to specific systems.

5.4 DEFINITION OF THE TELEMANIPULATION TASK

The robotic task was designed to evaluate performance and usability characteristics during object telemanipulation in a controlled environment. The main evaluated aspects were:

- **Positioning accuracy:** ability to move the manipulator to a desired position and orientation.
- **Three-dimensional and depth perception:** ability to estimate distances and spatial relationships between objects and the environment.

- **Object rotation and alignment:** ability to adjust object orientation according to task requirements.
- **Visual attribute perception:** ability to identify relevant visual characteristics of the manipulated object.
- **Manipulation speed and efficiency:** ability to complete the task efficiently while avoiding unnecessary movements.
- **Spatial planning and coordination:** ability to organize and execute movements while avoiding errors and collisions.

The experimental scenario consisted of a rectangular table containing a cube with four black faces and two yellow faces, together with a target base positioned at the center of the workspace. Initially, the cube was placed away from the base with one yellow face oriented upward.

The proposed task (Figure 35) consisted of telemanipulating the cube, rotating it until one of the black faces was oriented upward, and subsequently positioning and inserting the cube into the target base.

The proposed task enabled the integrated evaluation of the previously defined performance and usability aspects. Positioning accuracy and depth perception were required during cube approach and insertion into the target base. Rotation and alignment abilities were evaluated during orientation adjustment, while visual perception was involved in identifying the correct cube faces before manipulation. Manipulation efficiency was assessed through execution time and movement organization, whereas spatial planning was reflected in the sequence of approach, rotation, repositioning, and insertion actions performed throughout the task.

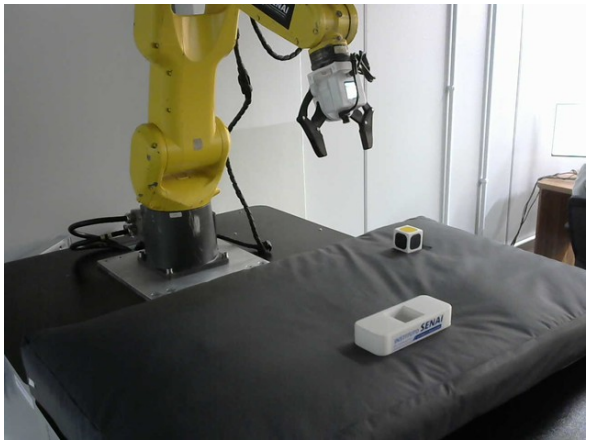
5.5 EXPERIMENTAL PROCEDURE

According to Jonathan Lazar, Feng, and Hochheiser (2017), a *within-subjects* design is one in which all participants are exposed to all experimental conditions, allowing direct comparisons between different methods or interfaces under the same control variables. This model reduces variability associated with individual differences, since each participant acts as their own control, increasing statistical sensitivity and result consistency.

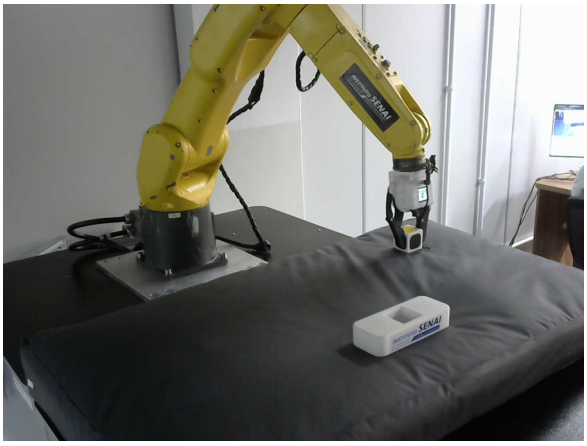
Based on this approach, all participants interacted with both teleoperation interfaces during the experiment. To minimize possible learning and fatigue effects, participants were divided into two groups of seven individuals. Group A started with the immersive interface and subsequently used the conventional interface, while Group B followed the inverse order.

Each experimental session was conducted individually in a controlled environment, with an average duration of approximately 90 minutes per participant. The proce-

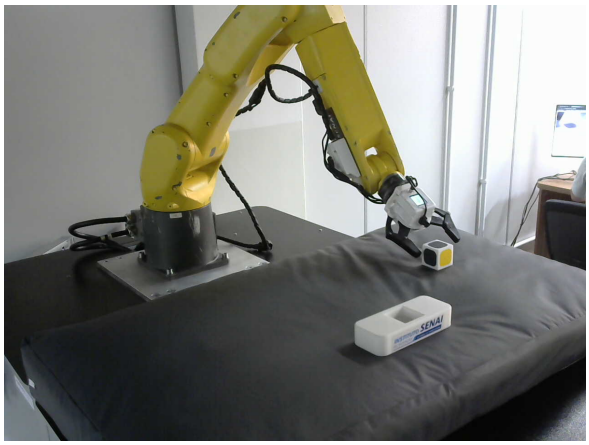
Figure 35 – Telemanipulation Task



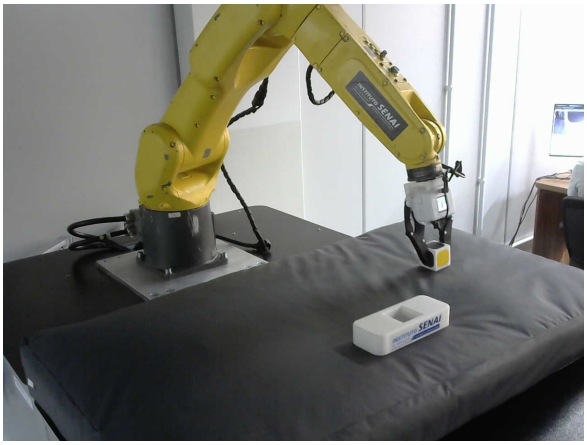
(a) Initial position



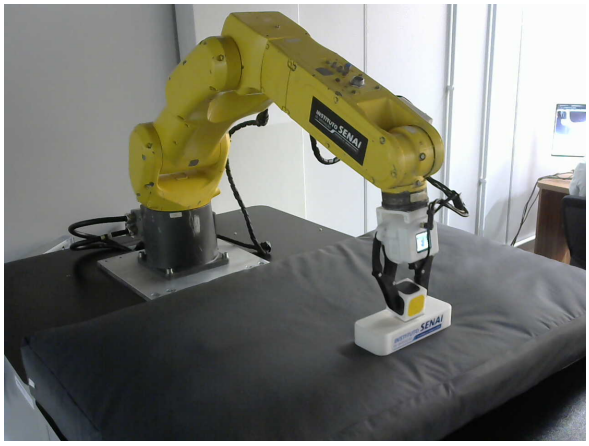
(b) Approach and grasp of the cube



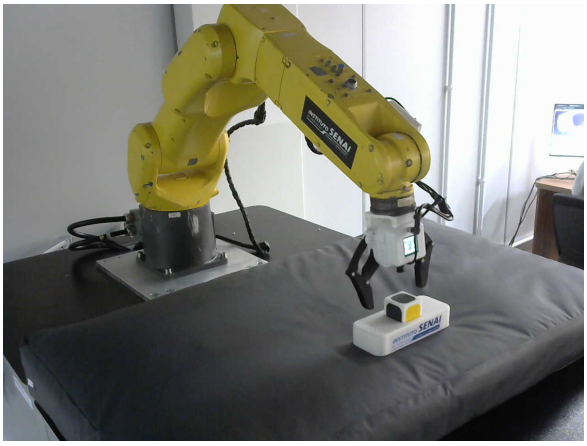
(c) Cube rotation



(d) Gripper repositioning



(e) Movement for cube insertion



(f) Task completed

Source: Author (2026).

dure consisted of the following steps:

1. **Initial briefing:** explanation to the participant of the general objectives of the study, ethical considerations, and signing of the *Termo de Consentimento Livre e Esclarecido* (Informed Consent Form) (TCLE), as well as the operating principles of the two evaluated teleoperation interfaces.
2. **Supervised training:** before starting the tests on each platform, a brief training session was conducted to familiarize the participant with the system controls and functionalities.
3. **Free adaptation period:** after training, the participant was given approximately 5 to 10 minutes to freely explore the system, aiming to adapt to the equipment, control dynamics, and spatial perception of the workspace. During this period, the participant was allowed to perform a preliminary execution of the proposed task under supervision.
4. **Task execution:** each participant then performed the task three consecutive times, with performance data recorded according to the criteria described in Section 5.6.1. A maximum time limit of 10 minutes was established for each execution. Trials exceeding this limit were considered incomplete.
5. **Qualitative evaluation:** after completing the executions on one of the interfaces, the participant answered the qualitative evaluation questionnaires presented in Section 5.6.2.
6. **Break between conditions:** after completing the questionnaires, a break of approximately 5 to 10 minutes was provided before the participant proceeded to the second interface.

Except for the initial briefing, all steps were repeated for both interfaces under equivalent experimental conditions.

5.6 EVALUATION METRICS

The evaluation combined quantitative and qualitative metrics in order to assess both operational performance and participant perception during teleoperation tasks.

5.6.1 Quantitative Evaluation

Quantitative metrics aim to measure participant performance in a measurable and comparable way. In usability studies, it is common to use multiple indicators, since different metrics capture distinct aspects of user interaction with the system (Sauro; Lewis, James R., 2016). To evaluate participant performance, the following parameters were considered:

- **Execution time:** total time required to complete the proposed task;

- **Number of collisions:** count of unintended contact events between the robotic manipulator and objects in the environment;
- **Number of grasp actions:** count of events in which the robotic gripper performs a grasp;
- **Success rate:** proportion of successfully completed tasks relative to the total number of attempts.

These variables were manually recorded during task execution and subsequently used for statistical comparison of the two teleoperation approaches.

5.6.2 Qualitative Evaluation

The qualitative evaluation was conducted using standardized questionnaires applied after each experimental condition. This stage aimed to assess perceived workload, usability, and user experience during teleoperation.

Additionally, open-ended comments were collected from participants during task execution and recorded by the researcher, with the aim of capturing spontaneous perceptions, encountered difficulties, and suggestions for improvement. According to Sauro and James R. Lewis (2016), comments and open-ended data constitute a relevant source of information in usability studies and can be organized, categorized, and analyzed alongside structured metrics.

The instruments used in the qualitative evaluation are presented below:

5.6.2.1 NASA Task Load Index (NASA-TLX)

The NASA-TLX is a questionnaire developed by NASA to assess the workload perceived by a participant during task execution (Hart; Staveland, 1988). Recent review studies indicate that NASA-TLX is one of the most widely used instruments for measuring workload in human–computer interaction experiments (Mortazavi *et al.*, 2024).

The method evaluates six main dimensions: **Mental Demand**, **Physical Demand**, **Temporal Demand**, **Performance**, **Effort**, and **Frustration**. Each of these dimensions represents a specific aspect of the cognitive and physical workload perceived by the user during the execution of an interactive task.

- **Mental Demand**: how mentally demanding the task was;
- **Physical Demand**: how physically demanding the task was;
- **Temporal Demand**: how rushed or time-pressured the participant felt;
- **Performance**: how well the participant believes they performed the task;
- **Effort**: how much effort the participant had to exert to achieve the obtained performance;

- **Frustration**: how insecure, irritated, stressed, or frustrated the participant felt during execution.

Each dimension is rated on a continuous scale from 0 to 100, ranging from “very low” to “very high”. The Performance item is inverted before calculating the overall workload (higher values indicate better performance and, therefore, lower perceived workload). After each task execution, participants complete the questionnaire by assigning a numerical value to each dimension.

In this study, a variation of the original NASA-TLX questionnaire, known as *Raw TLX* (or *RTLX*), was used (Byers; Bittner; Hill, 1989). This simplified version retains the same six dimensions of the original instrument but eliminates the weighting step between them, calculating a simple average of the participant ratings. Studies indicate that this modification produces results highly correlated with the weighted version, demonstrating equivalence in terms of validity and sensitivity (Hart, 2006; Grier, 2015). For this reason, *Raw TLX* is widely adopted in applied research due to its practicality and ease of use, being especially recommended in experiments with a limited number of participants, without compromising result reliability.

Table 8 presents the six dimensions of the NASA-RTLX and their respective descriptions.

Table 8 – Definitions of NASA-RTLX dimensions

Dimension	Anchors	Description
Mental Demand	Low / High	How much mental and perceptual activity was required? Was the method easy or demanding, simple or complex?
Physical Demand	Low / High	How much physical activity was required? Was the method easy or demanding, slack or strenuous?
Temporal Demand	Low / High	How much time pressure did you feel due to the pace at which the method or method elements occurred? Was the pace slow or rapid?
Performance	Good / Poor	How successful were you in performing the method? How satisfied were you with your performance?
Effort	Low / High	How hard did you have to work (mentally and physically) to accomplish your level of performance?
Frustration Level	Low / High	How irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the method steps?

Source: Adapted from Byers, Bittner, and Hill (1989).

5.6.2.2 System Usability Scale (SUS)

The SUS is one of the most widely used instruments for evaluating the perceived usability of interactive systems (Brooke *et al.*, 1996). Its broad adoption is associated with its simplicity of application, consistency of results, and ability to provide a global usability assessment (Pavlova, 2023). Recent review studies also indicate SUS as one

of the most frequently adopted questionnaires in evaluations involving interactive and immersive systems (Zanatta *et al.*, 2022; Mortazavi *et al.*, 2024).

The questionnaire consists of ten statements evaluated using a five-point Likert scale, ranging from “Strongly disagree” (1) to “Strongly agree” (5) (Likert, 1932; Joshi *et al.*, 2015). The final SUS score is calculated according to the standard scoring procedure proposed by Brooke *et al.* (1996). Odd-numbered items (S1, S3, S5, S7, and S9) are scored as the selected scale value minus 1, while even-numbered items (S2, S4, S6, S8, and S10) are scored as 5 minus the selected scale value. The resulting sum is multiplied by 2.5, producing a final usability score ranging from 0 (low usability) to 100 (high usability).

Table 9 presents the statements that compose the SUS questionnaire.

Table 9 – System Usability Scale (SUS) statements

No.	**Statement**
S1	I think that I would like to use this system frequently.
S2	I found the system unnecessarily complex.
S3	I thought the system was easy to use.
S4	I think that I would need the support of a technical person to be able to use this system.
S5	I found the various functions in this system were well integrated.
S6	I thought there was too much inconsistency in this system.
S7	I would imagine that most people would learn to use this system very quickly.
S8	I found the system very cumbersome to use.
S9	I felt very confident using the system.
S10	I needed to learn a lot of things before I could get going with this system.

Source: Adapted from Brooke *et al.* (1996).

5.6.2.3 User Experience Questionnaire (UEQ-S)

The UEQ is a standardized instrument widely used to evaluate user experience in interactive systems (Laugwitz; Held; Schrepp, 2008). Recent review studies indicate UEQ as one of the most frequently adopted methods for evaluating user experience in human–computer interaction research (Mortazavi *et al.*, 2024).

The reduced version of the questionnaire, known as User Experience Questionnaire - Short (UEQ-S), was adopted in this study and was originally developed as a faster and more practical alternative while preserving the conceptual structure of the original instrument (Hinderks, 2017). The UEQ-S consists of eight items evaluated through pairs of opposite adjectives using a seven-point semantic differential scale ranging from -3 to +3.

The questionnaire evaluates user experience through two main dimensions:

- **Pragmatic Quality**: related to usability, efficiency, clarity, and predictability during interaction;

- **Hedonic Quality**: related to stimulation, attractiveness, and emotional engagement during interaction.

The dimensions and their respective items are presented in Table 10.

Table 10 – Dimensions and items of the UEQ-S questionnaire

Dimension	Negative adjective		Positive adjective	Item
Pragmatic Quality	obstructive	o o o o o o o	supportive	1
	complicated	o o o o o o o	easy	2
	inefficient	o o o o o o o	efficient	3
	confusing	o o o o o o o	clear	4
Hedonic Quality	boring	o o o o o o o	exciting	5
	not interesting	o o o o o o o	interesting	6
	conventional	o o o o o o o	inventive	7
	usual	o o o o o o o	leading edge	8

Source: Adapted from Hinderks (2017).

For each participant, Pragmatic Quality and Hedonic Quality scores were computed as the arithmetic mean of their respective items. Positive values indicate more favorable user experience perceptions, whereas negative values indicate less favorable evaluations.

5.6.2.4 Questionnaire Administration

The NASA-TLX, SUS, and UEQ-S questionnaires were applied sequentially after each experimental condition. NASA-TLX was administered immediately after task execution to capture perceived workload while still recent in memory (*post-task*), whereas SUS and UEQ-S were applied after completion of the interaction condition to assess overall usability and user experience perceptions (*post-test*) (Hodrien; Fernando, 2021).

The questionnaires were applied in the following order:

- **NASA-TLX**: applied immediately after the three task repetitions;
- **SUS**: applied after the workload assessment to evaluate perceived usability;
- **UEQ-S**: applied last to assess overall user experience, including emotional and hedonic aspects.

Additionally, open-ended comments were collected throughout the experiment to record spontaneous perceptions, encountered difficulties, and participant suggestions.

5.6.3 Statistical Analysis

Statistical analyses were performed in the R statistical computing environment (R Core Team, 2026) using a within-subjects approach, since all participants interacted with both teleoperation interfaces.

Normality assumptions were assessed using the Shapiro–Wilk test (Shapiro; Wilk, 1965). When normality assumptions were satisfied, paired t-tests were applied for

comparisons between conditions. Otherwise, Wilcoxon signed-rank tests were used for non-parametric comparisons.

Statistical significance was established at a 95% confidence level (α = 0.05). Effect sizes were calculated using Cohen's *d* for parametric analyses and rank-biserial correlation (*r*) for non-parametric analyses (Cohen, 1988). Effect sizes were interpreted according to conventional thresholds, with values of *d* = 0.2, 0.5, and 0.8 representing small, medium, and large effects, respectively, and values of *r* = 0.1, 0.3, and 0.5 representing small, medium, and large effects.

5.7 ETHICAL CONSIDERATIONS

This study was conducted in accordance with the ethical guidelines established by Federal University of Santa Catarina (UFSC) and with national regulations for research involving human participants, in accordance with Resolution No. 510/2016 of the *Conselho Nacional de Saúde* (National Health Council) (CNS).

The project was submitted to and approved by the *Comitê de Ética em Pesquisa* (Research Ethics Committee) (CEP) of UFSC (CAAE: 95074725.0.0000.0121; Approval No. 8.307.489, approved on March 25, 2026), ensuring compliance with the principles of respect for participants' dignity, privacy, and autonomy.

All participants were properly informed about the objectives, procedures, and potential risks and benefits of the study, and informed consent was obtained through the TCLE, in accordance with national ethical guidelines.

Participation was voluntary and could be withdrawn at any time without any penalty. All collected information was treated confidentially and used exclusively for academic and research purposes. The data were anonymized and securely stored, with no individual identification of participants.

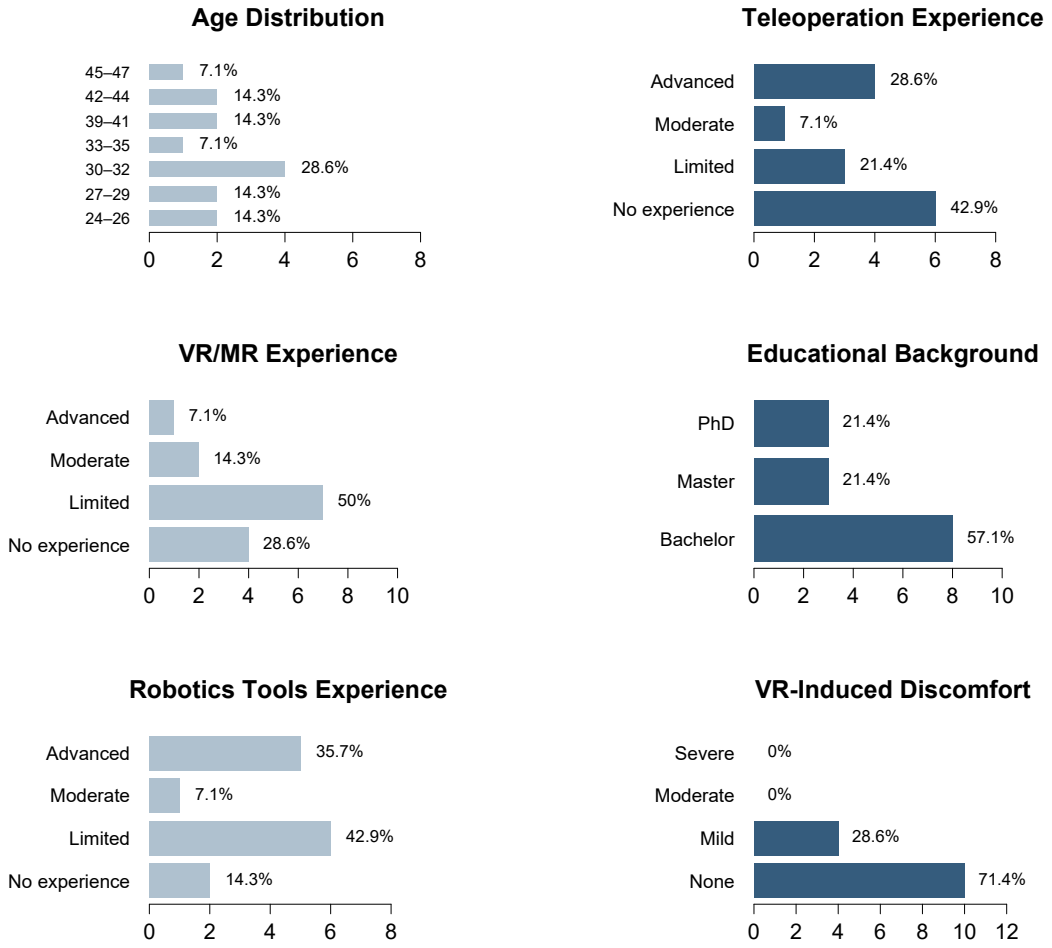
6 RESULTS

This chapter presents the results obtained from the experimental evaluation of the proposed teleoperation approaches. The results are organized into participant profile, quantitative results, and qualitative results.

6.1 PARTICIPANT PROFILE

The participant group consisted of 14 professionals working in the robotics field who held an engineering degree (bachelor's, master's, or PhD level). The participants were aged between 24 and 47 years and included both early-career professionals and experienced engineers and researchers. Figure 36 summarizes the main demographic and professional characteristics of the participants included in this study.

Figure 36 – Summary of the participants' profile in the study



Source: Author (2026).

Regarding prior experience with immersive technologies (VR/MR), four participants reported no prior experience, seven reported limited experience, two reported moderate experience, and one reported advanced experience. In relation to robotic teleoperation, six participants reported no prior experience, three reported limited experience, one reported moderate experience, and four reported advanced experience.

Additionally, participants showed different levels of familiarity with robotics tools such as ROS, Gazebo, MoveIt, and RViz. Two had no prior experience, six had limited experience, one had moderate experience, and five declared advanced experience, indicating a generally high level of familiarity with robotic systems.

With regard to discomfort associated with the use of immersive technologies, most participants (10 out of 14) did not report symptoms, while four reported mild discomfort, with no cases of moderate or severe discomfort observed.

6.2 QUANTITATIVE RESULTS

This section presents the quantitative results obtained during the experimental evaluation, focusing on operational performance and learning behavior across repeated trials under both teleoperation approaches.

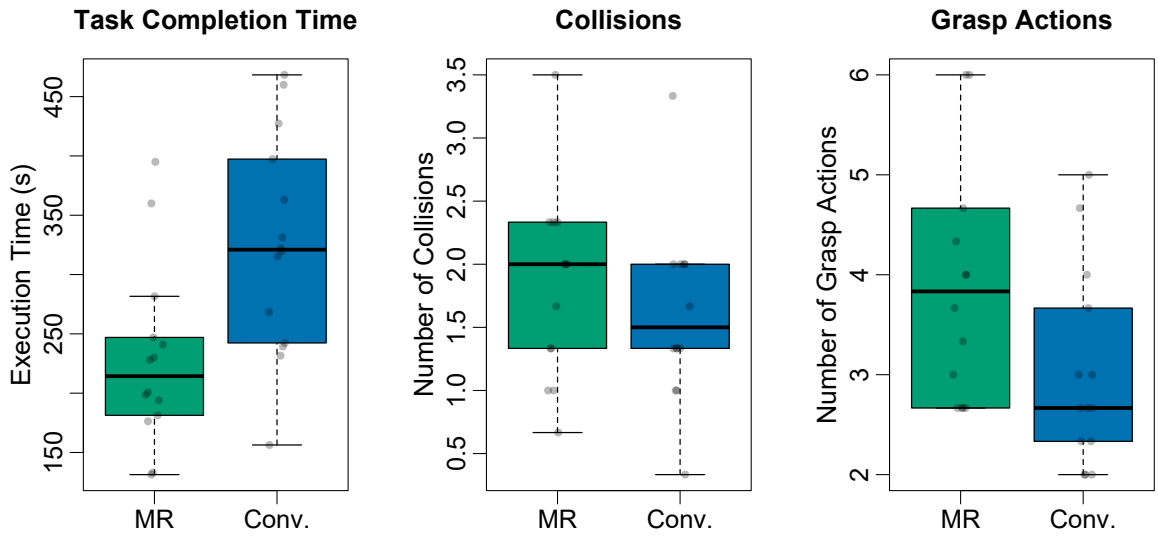
6.2.1 Operational Performance

Operational performance was evaluated through task completion rate, execution time, number of collisions, and number of grasp actions.

High completion rates were observed in both approaches. A total of 41 out of 42 trials (97.6%) were successfully completed using MR, while 40 out of 42 trials (95.2%) were completed using the conventional interface. In the unsuccessful trials, participants exceeded the predefined time limit and were unable to complete the task.

Figure 37 presents the distribution of the evaluated performance metrics.

Figure 37 – Operational performance metrics



Source: Author (2026).

Table 11 summarizes the statistical comparison between both approaches.

Table 11 – Summary of quantitative performance results.

Variable	**MR (M ± SD)**	**Conv. (M ± SD)**	**t(13)**	**p-value**	**Cohen's d**
Execution Time (s)	228.43 ± 75.81	324.48 ± 92.25	-4.07	0.001	-1.13
Collisions	2.15 ± 1.49	1.74 ± 0.89	1.12	0.282	0.33
Grasp Actions	4.12 ± 1.59	3.00 ± 0.98	2.47	0.028	0.84

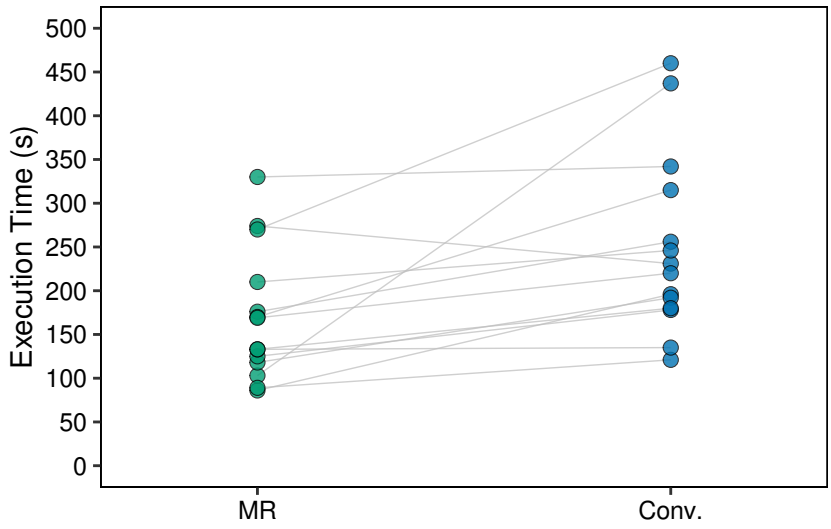
Source: Author (2026).

The assumption of normality was verified using the Shapiro–Wilk test, with no violations observed for the evaluated variables. Accordingly, paired t-tests were conducted for the operational performance comparisons.

Regarding execution time, lower values were observed in MR (228.43 ± 75.81 s) compared to the conventional interface (324.48 ± 92.25 s). The difference was statistically significant ($t(13) = -4.07$, $p = 0.001$), with a large effect size ($d = -1.13$).

Figure 38 presents the best execution time achieved by each participant, highlighting the individual differences between MR and conventional operation.

Figure 38 – Best execution time achieved by each participant



Source: Author (2026).

Collision counts remained relatively similar between approaches. MR resulted in an average of 2.15 ± 1.49 collisions, while the conventional interface presented 1.74 ± 0.89. No statistically significant difference was observed ($t(13) = 1.12$, $p = 0.282$, $d = 0.33$).

A higher number of grasp actions was observed in MR (4.12 ± 1.59) compared to the conventional interface (3.00 ± 0.98). The difference was statistically significant ($t(13) = 2.47$, $p = 0.028$), with a large effect size ($d = 0.84$). The minimum possible number of grasp actions required to complete the task was two, corresponding to one grasp for cube manipulation and a second grasp for final positioning.

6.2.2 Learning Behavior

Besides comparing overall performance, the study also examined how participant behavior changed across the repeated trials. The learning behavior analysis considered both execution time and grasp-action efficiency in order to evaluate short-term user adaptation during interaction with the robotic system.

One participant who did not complete all execution trials was excluded from this stage of the analysis due to incomplete data.

6.2.2.1 Execution-Time Learning Behavior

Table 12 presents the execution time statistics across repeated trials. Although both interfaces exhibited reductions in mean execution time throughout the experiment, substantial variability between participants was also observed, particularly in the conventional condition.

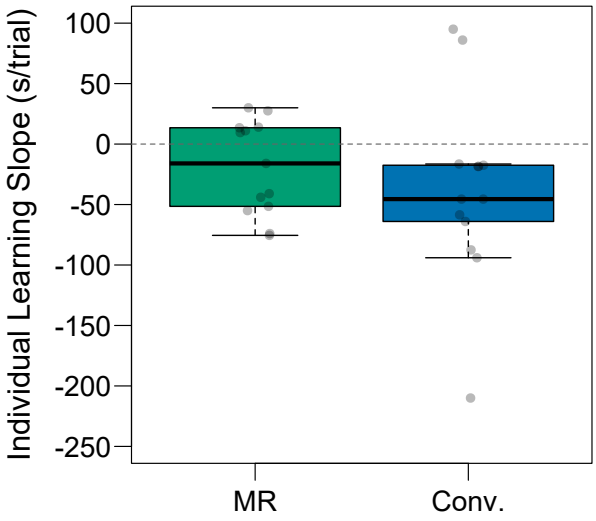
Table 12 – Execution time statistics across repeated trials (s).

Trial	MR (M ± SD)	Conv. (M ± SD)	MR Min–Max	Conv. Min–Max
1	228.2 ± 93.2	351.8 ± 143.8	89–440	121–600
2	237.2 ± 87.5	314.8 ± 125.2	103–415	166–538
3	189.5 ± 69.9	275.6 ± 97.0	86–330	135–446

Source: Author (2026).

To further investigate temporal performance variations across repeated trials, execution-time slopes were calculated for each participant throughout the experiment. Figure 39 illustrates the resulting learning slopes for each participant under both approaches. Negative slope values indicate reductions in execution time throughout the trials, whereas positive values indicate increases or unstable variations in execution time across repetitions.

Figure 39 – Individual execution-time learning slopes



Source: Author (2026).

Table 13 summarizes the results of the individual execution-time slope analysis.

Table 13 – Summary of individual execution-time learning slope analysis.

MR (M ± SD)	Conv. (M ± SD)	t(12)	p-value	Cohen's d
-19.35 ± 38.84	-38.08 ± 76.97	0.81	0.432	0.30

Source: Author (2026).

Normality assumptions were verified using the Shapiro–Wilk test, with both slope distributions compatible with normality (MR: $p = 0.0916$; Conv.: $p = 0.1780$). A paired t-test was therefore applied for the comparison.

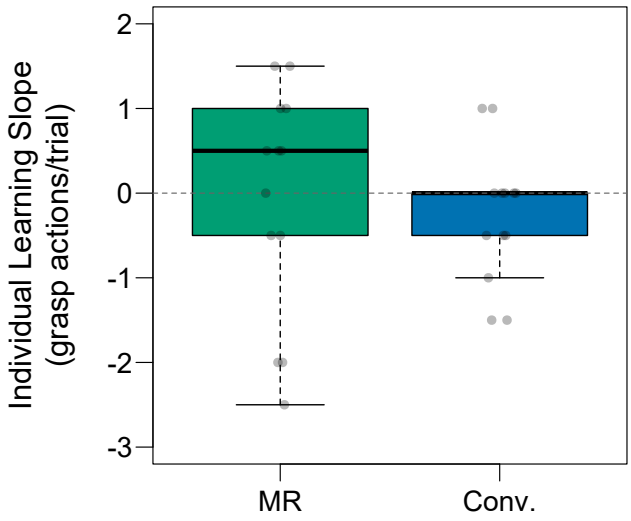
The MR interface presented a mean execution-time slope of -19.35 ± 38.84 s/-trial, whereas the conventional interface presented -38.08 ± 76.97 s/trial. Although the conventional approach exhibited a numerically steeper reduction in execution time throughout the trials, no statistically significant difference was observed between approaches ($t(12) = 0.81$, $p = 0.432$). The calculated effect size was small ($d = 0.30$).

However, individual performance occasionally fluctuated between trials due to isolated operational errors, contributing to the variability observed in the calculated slopes.

6.2.2.2 Grasp-Action Learning Behavior

Figure 40 presents the individual grasp-action learning slopes calculated across the repeated trials. Negative slope values indicate reductions in the number of grasp actions throughout the experiment, potentially indicating reductions in grasp-action variability throughout the experiment.

Figure 40 – Individual grasp-action learning slopes



Source: Author (2026).

Table 14 summarizes the results of the individual grasp-action learning slope analysis.

Table 14 – Summary of individual grasp-action learning slope analysis.

MR (M $\pm$ SD)	**Conv. (M $\pm$ SD)**	**t(12)**	**p-value**	**Cohen's d**
-0.08 $\pm$ 1.35	-0.27 $\pm$ 0.78	0.51	0.620	0.17

Source: Author (2026).

Normality assumptions were also verified for the grasp-action slope distributions using the Shapiro–Wilk test (MR: p = 0.1057; Conv.: p = 0.1865). Accordingly, paired t-tests were applied for the statistical analysis.

The MR interface presented a mean grasp-action slope of -0.08 ± 1.35 grasp actions/trial, while the conventional interface presented -0.27 ± 0.78 grasp actions/trial. Although the conventional approach again exhibited a slightly larger numerical reduction throughout the trials, no statistically significant difference was observed between approaches ($t(12) = 0.51$, $p = 0.620$). The effect size was very small ($d = 0.17$), indicating minimal observable difference between interfaces regarding grasp-action variation across repeated trials.

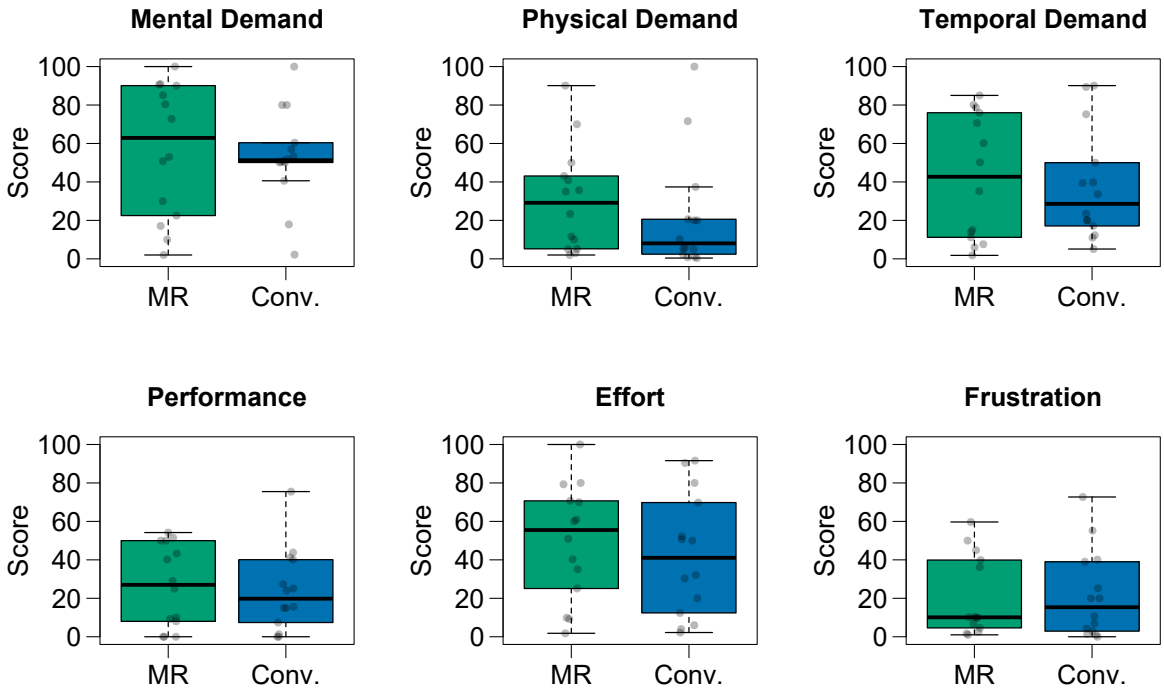
6.3 QUALITATIVE RESULTS

This section presents the subjective evaluation results obtained from the NASA-TLX, SUS, and UEQ-S questionnaires. The analyses focused on perceived workload, usability, and user experience during interaction with the teleoperation interfaces. Details of the questionnaires and their application procedure are presented in Section 5.6.2.

6.3.1 NASA-TLX

The workload analysis was conducted using the NASA-TLX questionnaire, computed through the Raw TLX approach as the arithmetic mean of the six evaluated dimensions. Figure 41 illustrates the distribution of the NASA-TLX dimension scores for both approaches.

Figure 41 – NASA-TLX dimension scores.



Source: Author (2026).

Table 15 summarizes the NASA-TLX results for all evaluated dimensions and for the overall workload score.

Table 15 – Summary of NASA-TLX results.

Dimension	**MR (M ± SD)**	**Conv. (M ± SD)**	**Statistic**	**p-value**	**Effect Size**
Mental Demand	56.81 ± 34.65	53.19 ± 24.39	W = 65	0.451	r = 0.21
Physical Demand	30.35 ± 27.03	21.49 ± 29.73	t(13) = 1.06	0.308	d = 0.31
Temporal Demand	42.26 ± 32.39	37.62 ± 28.65	t(13) = 0.43	0.675	d = 0.15
Performance	26.46 ± 21.47	23.66 ± 21.10	t(13) = 0.38	0.711	d = 0.13
Effort	49.52 ± 30.18	42.26 ± 31.84	t(13) = 0.58	0.575	d = 0.23
Frustration	20.54 ± 20.69	21.38 ± 22.68	t(13) = -0.11	0.912	d = -0.04
NASA-TLX Total	37.66 ± 20.47	33.27 ± 17.51	t(13) = 0.62	0.549	d = 0.23

Source: Author (2026).

Note: The Mental Demand dimension violated normality assumptions and was therefore analyzed using the Wilcoxon signed-rank test with rank-biserial correlation effect size (r). The remaining dimensions were analyzed using paired t-tests with Cohen's d effect sizes.

Normality assumptions were assessed using the Shapiro–Wilk test for the NASA-TLX score differences. Since the Mental Demand dimension violated normality assump-

tions (p = 0.0096), the Wilcoxon signed-rank test was applied for this dimension. The remaining dimensions, including the overall NASA-TLX score, satisfied normality assumptions and were therefore analyzed using paired t-tests.

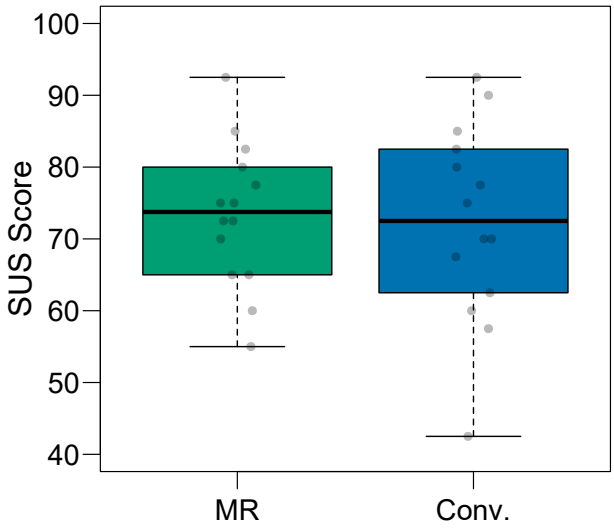
The MR approach presented an overall NASA-TLX score of 37.66 $\pm$ 20.47, whereas the conventional approach presented 33.27 $\pm$ 17.51. No statistically significant difference was observed for the overall workload score ($t(13)$ = 0.62, p = 0.549), with a small effect size (d = 0.23).

Across the individual NASA-TLX dimensions, no statistically significant differences were observed ($p > 0.05$ for all comparisons). Effect sizes also remained small across all evaluated dimensions.

6.3.2 SUS

The usability analysis was conducted using the SUS, allowing the evaluation of participant perceptions regarding usability, ease of interaction, and overall system acceptance. Figure 42 presents the SUS scores obtained for both approaches, while Table 16 summarizes the usability results.

Figure 42 – SUS scores



Source: Author (2026).

Table 16 – Summary of SUS results.

Variable	MR (M $\pm$ SD)	Conv. (M $\pm$ SD)	t(13)	p-value	Cohen's d
SUS Score	73.39 $\pm$ 10.08	72.32 $\pm$ 13.78	0.21	0.839	0.09

Source: Author (2026).

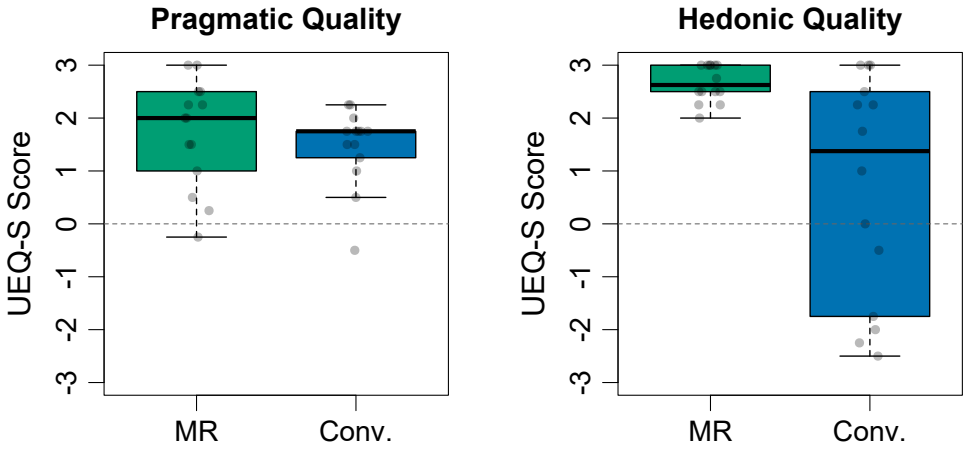
Normality assumptions for the SUS score differences were verified using the Shapiro–Wilk test, with no violations observed (p = 0.758). A paired t-test was therefore applied for the comparison.

The MR approach presented a mean SUS score of 73.39 $\pm$ 10.08, whereas the conventional approach presented 72.32 $\pm$ 13.78. No statistically significant difference was observed between approaches ($t(13)$ = 0.21, p = 0.839). The calculated effect size was negligible (d = 0.09).

6.3.3 UEQ-S

The user experience analysis was conducted using the UEQ-S questionnaire, considering both Pragmatic Quality and Hedonic Quality dimensions associated with the interaction experience. Figure 43 presents the UEQ-S scores obtained for both approaches, while Table 17 summarizes the results.

Figure 43 – UEQ-S scores



Source: Author (2026).

Note: Pragmatic Quality reflects task-oriented aspects of the interaction, such as efficiency and usability, whereas Hedonic Quality reflects non-task-oriented aspects, including stimulation, enjoyment, and user engagement.

Table 17 – Summary of UEQ-S results.

Dimension	**MR (M $\pm$ SD)**	**Conv. (M $\pm$ SD)**	**Statistic**	**p-value**	**Effect Size**
Pragmatic Quality	1.71 $\pm$ 1.01	1.46 $\pm$ 0.73	t(13) = 0.62	0.544	d = 0.28
Hedonic Quality	2.66 $\pm$ 0.35	0.70 $\pm$ 2.14	t(13) = 3.35	0.005	d = 1.32

Source: Author (2026).

The assumption of normality was verified using the Shapiro–Wilk test for both UEQ-S dimensions, with no significant deviations from normality observed (p = 0.671

for Pragmatic Quality and $p = 0.054$ for Hedonic Quality). Paired t-tests were therefore applied for both comparisons.

For Pragmatic Quality, the MR approach presented a score of 1.71 ± 1.01, whereas the conventional approach presented 1.46 ± 0.73. No statistically significant difference was observed between approaches ($t(13) = 0.62$, $p = 0.544$), with a small effect size ($d = 0.28$).

For Hedonic Quality, the MR approach presented substantially higher scores (2.66 ± 0.35) compared to the conventional approach (0.70 ± 2.14). The difference was statistically significant ($t(13) = 3.35$, $p = 0.005$), with a large effect size ($d = 1.32$).

7 DISCUSSION

This chapter discusses the results obtained from the experimental evaluation of the immersive teleoperation system in comparison with the conventional interface. The discussion integrates the quantitative and qualitative findings presented in Chapter 6, relating the observed results to participant perceptions and the literature reviewed throughout this dissertation.

7.1 OPERATIONAL PERFORMANCE AND LEARNING BEHAVIOR

The operational performance results indicate that the immersive teleoperation interface provided advantages in task execution speed during the evaluated manipulation task. The significantly lower execution times observed in the MR condition suggest that the immersive interaction enabled more fluid and continuous robot movements, reducing the amount of discrete adjustments required during task execution. This behavior was highly consistent across participants, with 13 out of the 14 participants achieving faster execution times in MR when compared with their own performance in the conventional interface, as illustrated in Figure 38.

One possible explanation for this result is the interaction workflow adopted in each condition. Unlike the conventional interface, which required sequential axis-by-axis adjustments through RViz manipulation widgets, the MR interface allowed simultaneous control of robot position and orientation through direct spatial interaction. As a consequence, participants could perform movements in a more natural and continuous manner, reducing the dependence on intermediate corrections during manipulation tasks.

The immersive interface also appeared to reduce part of the cognitive translation normally required in conventional teleoperation systems. In the MR condition, participants directly associated hand movement with robot motion in three-dimensional space, whereas the conventional interface required indirect interpretation through camera perspectives, visual widgets, and discrete manipulation controls. Several participant comments reinforce this interpretation. Participant P05 reported that “depth perception using the 2D screen was much more difficult” in the conventional interface, while describing the MR interaction as “easier and more intuitive”. Similarly, participant P03 stated that the conventional interface “required more concentration” during task execution. These observations are consistent with previous studies that associate immersive visualization with improvements in spatial awareness and teleoperation interaction (Candeloro *et al.*, 2015; Elor *et al.*, 2021).

Despite the execution-time advantages, the MR interface also presented important limitations regarding fine manipulation tasks. During the experimental sessions, some participants unintentionally released the object while repositioning the robot tar-

get due to confusion between the trigger responsible for gripper actuation and the grab interaction used for robot manipulation in the immersive interface (Figure 33). In contrast, gripper control in the conventional interface was performed through a dedicated slider (Figure 32), which reduced the likelihood of unintended releases during robot manipulation. Consequently, some participants required additional grasp attempts during task execution in the MR condition, contributing to the higher number of grasp actions recorded for the MR interface.

Another important aspect involves the automatic control mode, which relied on sequential motion planning and execution stages. As described in Section 4.3.4.2, this behavior introduced short interruptions between user command and robot motion. Multiple participants reported that this behavior negatively affected interaction fluidity. Participant P14 stated that the delay “interfered considerably with the user experience” and suggested that the immersive interface would become substantially better if movements were more fluid and responsive.

The interruptions became particularly noticeable during precise positioning tasks, where participants frequently needed to adapt their interaction rhythm to the planning-execution cycle of the system in order to perform small corrections. This behavior became more perceptible in the MR interface because, in addition to the interruptions introduced by the automatic control mode, robot positioning relied directly on continuous hand movement. In contrast, the conventional interface relied on more segmented adjustments through manipulation gizmo in RViz.

Overall, these observations indicate that, despite the advantages of immersive interaction for spatial positioning and robot guidance, precise manipulation tasks still require careful motor control from the operator, particularly during object handling and final positioning.

The collision results also provide insights regarding interaction behavior during teleoperation, because although no statistically significant differences were observed between conditions, observations during the experiments suggest that participants adopted different movement strategies depending on the interface. In the immersive condition, some participants performed faster and more direct movements, occasionally leading to abrupt approaches and unintended contacts. In contrast, the conventional interface required more segmented adjustments through manipulation gizmo interaction in RViz, where movements and rotations were adjusted one axis at a time. As a result, movements in the conventional interface tended to become more controlled and incremental than those observed in MR, where interactions were generally more fluid and intuitive during manipulation tasks.

Regarding learning behavior, no statistically significant differences were observed between learning slopes. However, the interpretation of short-term adaptation should be approached cautiously, since execution times were occasionally influenced by

isolated operational events during specific trials, such as unintended object releases or alignment difficulties during insertion attempts. In several cases, participants completed one trial rapidly, experienced delays in a subsequent repetition due to a specific operational error, and then returned to faster performance levels in later trials. Consequently, variations in execution time do not necessarily indicate the absence of learning, but may also reflect the operational variability inherent to telemanipulation tasks (Figure 35).

Another possible explanation for the absence of larger execution-time learning slopes in MR is that several participants already achieved relatively fast execution times during the initial trials, reducing the magnitude of observable improvement throughout repetition. In contrast, participants using the conventional interface frequently started with substantially longer execution times, allowing larger reductions across subsequent trials.

7.2 WORKLOAD, USABILITY, AND USER EXPERIENCE

The subjective evaluation results provide important insights into how participants perceived the immersive and conventional teleoperation interfaces beyond operational performance metrics. Although the quantitative performance analysis showed differences in execution time and grasp behavior, the workload, usability, and user experience assessments demonstrate that the interaction experience cannot be fully interpreted through efficiency metrics alone.

The NASA-TLX results indicated no statistically significant differences between interfaces for either the overall workload score or the individual workload dimensions. Despite the operational advantages observed in MR, participants reported relatively similar levels of perceived mental demand, effort, temporal demand, frustration, and overall workload across both teleoperation approaches.

One important aspect observed throughout the analyses was the high variability between participant responses, particularly for the NASA-TLX dimensions. Considerable differences were observed in how participants perceived mental demand, effort, and physical interaction during teleoperation tasks. This variability may indicate that subjective workload perception depends strongly on individual characteristics such as interaction strategy, familiarity with robotic systems, and previous experience with immersive technologies.

Although no statistically significant relationships were identified between demographic variables and NASA-TLX results, the combination of high response variability and the relatively small participant sample may have limited the detection of more subtle effects between user profiles and subjective workload perception.

Participant comments nevertheless suggested distinct individual perceptions regarding the immersive interaction. While some participants described the MR interface as intuitive and natural, others reported difficulties related to fine positioning adjust-

ments, system responsiveness, and interaction adaptation. These differences reinforce the idea that subjective workload perception in teleoperation scenarios may be highly participant-dependent.

The SUS results showed no statistically significant differences between the immersive and conventional teleoperation interfaces, indicating comparable usability perceptions between the two approaches despite the operational differences identified throughout the performance analyses. High variability was also observed between individual SUS responses, suggesting that usability perception in teleoperation systems may vary considerably according to individual interaction preferences and adaptation strategies. Nevertheless, the mean SUS scores obtained for both interfaces remained within usability ranges commonly interpreted in the literature as representing good usability (Bangor; Kortum; Miller, 2009).

The UEQ-S results showed the clearest subjective difference between the two teleoperation approaches. While no statistically significant differences were observed for Pragmatic Quality, the immersive interface achieved considerably higher Hedonic Quality scores than the conventional condition, with a large effect size.

The absence of significant differences in Pragmatic Quality is consistent with the SUS results, suggesting that both interfaces were perceived as comparably effective in terms of usability, efficiency, and interaction clarity. In contrast, the Hedonic Quality results indicate that the MR interface provided a more engaging and positively perceived interaction experience.

At the same time, the higher Hedonic Quality scores observed for the immersive interface may also be partially influenced by novelty-related effects commonly associated with immersive technologies, such as the “wow” effect, in which users tend to report more positive hedonic perceptions during interactions with MR systems (McMahan *et al.*, 2012; Arlati *et al.*, 2018; Mondellini *et al.*, 2022; Whitton, 2003). However, the demographic results indicate that several participants already had prior experience with immersive technologies, suggesting that the observed hedonic advantage was not exclusively associated with novelty, but may also reflect characteristics inherent to immersive interaction itself, such as increased spatial involvement and interaction naturalness.

Another relevant aspect involves the higher variability observed in the Hedonic Quality scores of the conventional condition (SD = 2.14), compared with the immersive interface (SD = 0.35). This behavior suggests that participant perceptions regarding the conventional interaction varied considerably, whereas the MR interface produced a more consistently positive hedonic evaluation across users.

Although the immersive interface produced higher Hedonic Quality scores and faster execution times during the manipulation task, the theoretical discussions presented in Section 2.1.3.4.2 indicate that telepresence and immersion should not be

interpreted as direct predictors of operational superiority. While studies such as Ferre *et al.* (2007) suggest that immersive interaction may contribute positively to remote manipulation performance, the literature still does not establish a clear consensus regarding whether increased telepresence consistently leads to superior operational performance across different teleoperation scenarios (Council *et al.*, 1994; Bechtel *et al.*, 2023; Youssef *et al.*, 2023; Boer, 2021). Therefore, the practical benefits of immersive teleoperation appear to depend strongly on the specific context and the characteristics of the task being performed.

7.3 METHODOLOGICAL CONTRIBUTIONS AND RESEARCH GAP

Beyond the quantitative and qualitative results discussed in the previous sections, the comparative literature review suggested recurring methodological tendencies regarding the evaluation of immersive teleoperation systems. Although most studies reported advantages associated with MR interfaces, important limitations were identified in the way usability and user experience were assessed.

7.3.1 Synthesis of Findings from the Comparative Literature

The reviewed studies consistently reported potential advantages associated with immersive teleoperation interfaces, including reductions in workload, improvements in usability, faster task execution, increased safety during manipulation tasks, and higher operator preference. Nevertheless, the comparative analysis also revealed important methodological limitations recurring throughout the literature. Table 18 summarizes the principal findings and limitations identified in the reviewed studies

Table 18 – Main findings and limitations identified in the reviewed studies

Study	Main finding	Main limitation
Pryor *et al.* (2023)	MR showed non-significant trends toward lower workload, perceived difficulty, and task time, while task success and precision remained comparable to the conventional 2D interface.	Observed differences did not reach statistical significance. The absence of standardized usability and hedonic UX assessment, combined with the small sample size, limited conclusions regarding MR superiority.
LeMasurier *et al.* (2024)	Conventional teleoperation achieved higher task performance, while MR resulted in fewer collisions and safer interaction. No significant differences were found in usability, workload, or learning rate.	Evaluation relied on overlapping pragmatic measures (SUS/EoU and NASA-TLX/GTQ) without hedonic UX assessment. For the manipulation task, the predominance of non-significant findings limited the interpretation of user experience.
Whitney *et al.* (2020)	The MR interface enabled faster task execution, higher usability, and lower workload than conventional interface.	Claims that MR was more immersive and intuitive were inferred from performance, NASA-TLX, SUS, and simple Likability ratings without direct UX assessment. It remained unclear whether MR preference resulted from usability, novelty, or emotional engagement.
Condori *et al.* (2025)	The immersive MR interface reduced workload and task execution time while improving usability and task success rates.	Evaluation relied only on NASA-TLX and SUS, while intuitive and natural interaction was inferred from performance and workload results without direct UX assessment.
Yunpeng Su *et al.* (2022)	The MR interface reduced execution time and mental workload while improving perceived usefulness, ease of use, and training efficiency.	Evaluation relied on NASA-TLX and TAM measures only. Claims of natural and engaging interaction were inferred without direct UX assessment, while the absence of SUS limited standardized usability benchmarking and comparison with related studies.
Wu, Zhao, and Li (2025)	The MR interface reduced workload and improved perceived usability compared to traditional control.	Evaluation focused only on workload and functional usability. Claims of intuitive and user-friendly interaction were inferred from NASA-TLX and SUS results without direct UX assessment.
Kanazawa, Sato, and Morita (2024)	MR outperformed the gamepad in execution speed and consistency. Users perceived their performance as better in MR despite higher physical effort.	Evaluation relied only on NASA-TLX, limiting understanding of why MR was perceived as more effective despite higher physical demand and leaving MR preference without a clear experiential explanation.
Jankowski and Grabowski (2015)	MR was perceived as more intuitive, immersive, and comfortable.	Reliance on custom questionnaires and MEC-SPQ limited comparison with standardized instruments. Intuitive and immersive interaction was claimed without direct assessment of these UX dimensions.
Hasan, Yasar, and Iqbal (2026)	No interface was universally superior; prior familiarity reduced perceived workload but did not improve performance.	Evaluation was limited to workload and Likability, restricting interpretation of the divergence between perceived ease and operational efficiency.

Source: Author (2026)

7.3.2 Methodological Patterns Observed in the Literature

Beyond the individual limitations identified in each study, the literature review revealed four recurring methodological patterns associated with immersive teleoperation evaluation. These patterns are summarized in Table 19.

Table 19 – Methodological patterns and recurring limitations identified in the reviewed studies

Study	Pragmatic-only assessment	Limited benchmark	Inferred intuitive/ natural interaction	Statistical inconclusiveness
Pryor *et al.* (2023)	X	X	-	X
LeMasurier *et al.* (2024)	X	-	-	X
Whitney *et al.* (2020)	-	-	X	-
Condori *et al.* (2025)	X	-	X	-
Yunpeng Su *et al.* (2022)	X	X	X	-
Wu, Zhao, and Li (2025)	X	-	X	-
Kanazawa, Sato, and Morita (2024)	X	X	-	-
Jankowski and Grabowski (2015)	-	X	-	-
Hasan, Yasar, and Iqbal (2026)	-	X	X	-
Present Dissertation	-	-	-	-

Source: Author (2026)

The first pattern identified was *Pragmatic-only assessment*. Most studies concentrated primarily on workload, usability, efficiency, or operational performance, while providing limited investigation of broader experiential and hedonic aspects of immersive interaction. Although these approaches are suitable for evaluating functional performance, they offer limited understanding regarding how immersive interfaces influence engagement, stimulation, novelty, or emotional perception during teleoperation activities.

The second recurring pattern corresponds to *Limited benchmarking*. Several studies performed comparative evaluations using subjective measures with restricted UX coverage, frequently relying on custom questionnaires, simplified preference scales,

or predominantly pragmatic assessment approaches, despite the widespread adoption of standardized usability references such as SUS, which is one of the most widely adopted instruments for usability evaluation and benchmarking in interactive systems research (Lewis, James R, 2018).

The third recurring pattern involved *Inferred intuitive/natural interaction*. Several studies explained operational or pragmatic improvements, such as lower workload, higher usability, improved performance, or user preference, through concepts such as intuitive interaction, natural control, immersion, or user-friendliness without directly measuring these characteristics. In these cases, the interpretation of qualitative UX and hedonic advantages remained based primarily on indirect inference from pragmatic results rather than on dedicated UX assessment methods.

Finally, some studies also presented *Statistical inconclusiveness*. In these cases, MR interfaces frequently exhibited favorable numerical trends regarding performance, workload, or user perception, but these tendencies were not consistently supported by statistically significant differences. As a consequence, it remained difficult to determine whether the observed effects reflected real advantages of immersive interaction or simply variability associated with the experimental design, sample size, or evaluation methodology.

7.3.3 Validation of the Research Gap

Regarding the statistical inconclusiveness, in the present study, NASA-TLX and SUS results did not reveal substantial differences between the immersive and conventional interfaces, similarly to the findings reported by (Pryor *et al.*, 2023; LeMasurier *et al.*, 2024). However, UEQ-S identified clearer distinctions related to hedonic and experiential aspects associated with the immersive interaction. This result suggests that evaluations centered primarily on workload and usability may be insufficient to fully capture qualitative differences associated with immersive teleoperation systems.

The results also reinforce the research gap identified in the comparative literature review regarding inferred intuitive and natural interaction. Some reviewed studies associated intuitive, natural, or immersive interaction characteristics with pragmatic results such as usability, workload reduction, or operational performance without directly evaluating broader UX dimensions. However, the present results suggest that these relationships cannot be assumed as necessarily equivalent, since user perception did not fully follow the same tendencies observed in NASA-TLX, SUS, and quantitative performance metrics.

This interpretation is further supported by (Hasan; Yasar; Iqbal, 2026), who argue that “more intuitive” interfaces are not necessarily always preferable, emphasizing the importance of considering task characteristics and operator needs when evaluating teleoperation systems. In the present dissertation, the inclusion of UEQ-S enabled the

identification of subjective differences that would likely remain partially unexplained if the evaluation had relied exclusively on workload, usability, and quantitative performance measures.

Together, these findings suggest that evaluation approaches restricted primarily to pragmatic dimensions such as workload, usability, and operational performance may be insufficient to fully explain subjective differences associated with immersive interaction. The inclusion of broader user experience measures, particularly those capable of assessing hedonic dimensions, can contribute to a more comprehensive interpretation of operator perceptions and help explain differences that may remain undetected when relying exclusively on pragmatic evaluation metrics.

7.3.4 Methodological Contribution of this Dissertation

Beyond the experimental results obtained with the proposed MR teleoperation system, this dissertation also contributes methodologically to the comparative evaluation of robotic arm teleoperation interfaces through the adoption of an integrated and standardized evaluation framework.

As shown in Table 19, the evaluation framework adopted in this dissertation is not characterized by the methodological limitations recurrently observed in the reviewed studies, particularly pragmatic-only assessments, limited benchmarking strategies, indirect interpretations of intuitive interaction, and cases of statistical inconclusiveness. This was achieved through the combined use of quantitative performance metrics and the complementary application of NASA-TLX, SUS, and UEQ-S, enabling the simultaneous evaluation of operational performance, workload, usability, and hedonic user experience dimensions.

Furthermore, the results obtained in this dissertation demonstrated that user experience differences were not always reflected equivalently in workload, usability, or operational performance measures, reinforcing the importance of multidimensional evaluation strategies.

In this context, the main methodological contribution of this dissertation lies in demonstrating the importance of treating workload, usability, operational performance, and user experience as complementary rather than interchangeable dimensions during immersive teleoperation evaluation. This integrated perspective contributes to a broader understanding of operator perception and interaction in MR teleoperation systems while maintaining conceptual alignment with the quality-in-use principles defined in ISO 9241-11:2018 (International Organization for Standardization, 2018).

8 FINAL CONSIDERATIONS

This dissertation presented the development, integration, and comparative evaluation of two teleoperation interfaces applied to the control of a FANUC LR Mate 200iD/7L industrial manipulator: a conventional interface based on MoveIt/RViz and an immersive MR interface implemented using Unity, ROS, and Meta Quest 3. The study investigated differences between both approaches regarding operational performance, workload, usability, learning behavior, and user experience during telemanipulation tasks.

In addition to the experimental evaluation itself, the research also addressed a methodological gap identified in the literature concerning the limited integration of quantitative performance metrics and standardized qualitative assessment instruments in immersive teleoperation studies. To address this gap, the dissertation adopted an integrated evaluation framework combining quantitative performance measures with NASA-TLX, SUS, and UEQ-S.

8.1 MAIN FINDINGS

The experimental results demonstrated that the immersive interface enabled significantly faster task execution than the conventional interface; however, it also increased the number of grasp attempts. Collision results, on the other hand, did not present statistically significant differences between the evaluated conditions.

Regarding learning behavior, no statistically significant differences were observed between interfaces. However, execution-time variability during repeated trials suggests that operational events such as unintended releases and alignment difficulties may have partially influenced the learning analyses.

In addition, the automatic planning-execution workflow introduced short interruptions that negatively affected interaction fluidity during precise positioning tasks in both teleoperation conditions.

The MR platform was perceived by most participants as more intuitive and easier for robot movement execution during most manipulation activities. However, during the insertion stage of the task, where fine positioning precision was required, some participants experienced difficulties, mainly due to the greater dependence on motor coordination and hand movement stability. In contrast, the conventional interface was generally perceived as less intuitive during larger movements, but was preferred by some participants during fine positioning tasks because it allowed segmented axis-by-axis adjustments without depending directly on the operator's motor stability.

Despite these limitations related to robotic manipulation precision, it is important to emphasize that MR teleoperation should not be interpreted as being restricted to the direct use of HMD handheld controllers, since different control approaches and devices may be integrated into both immersive and conventional teleoperation systems. In this

context, the main advantage of MR is more directly associated with the immersive visualization of the robotic workspace and the three-dimensional perception of the remote environment rather than with the specific control device itself. These characteristics may become particularly relevant in teleoperation scenarios where spatial perception and environmental awareness are important during remote interaction, such as high-risk operational environments.

The qualitative results revealed important distinctions between workload, usability, and user experience assessments. NASA-TLX and SUS did not identify substantial differences between the immersive and conventional interfaces, indicating comparable workload and usability perceptions across both conditions. In contrast, UEQ-S identified clearer differences related to hedonic and experiential aspects of interaction, with the immersive interface achieving significantly higher Hedonic Quality scores.

Overall, the UEQ-S results suggest that the main subjective advantage of the immersive teleoperation platform was less related to traditional usability or workload measures and more associated with the experiential and hedonic aspects of immersive interaction.

8.2 RESEARCH OBJECTIVES AND HYPOTHESES

The general objective of this dissertation was achieved through the development and comparative evaluation of immersive and conventional robotic teleoperation interfaces applied to an industrial robotic arm. The proposed MR teleoperation system was successfully integrated with ROS, Unity, and the FANUC manipulator, enabling experimental validation through practical telemanipulation tasks.

The specific objectives related to system development, operational performance evaluation, workload assessment, usability analysis, user experience evaluation, and learning behavior investigation were also achieved through the experimental protocol adopted in this research.

Regarding the research hypotheses, the obtained results partially supported the initial assumptions of the study.

Hypothesis 1 was partially supported, since the immersive interface was frequently perceived by participants as more intuitive and easier to control during teleoperation tasks, particularly during larger and more general robot movements. However, some participants still preferred the conventional interface during fine positioning activities requiring higher movement precision.

Hypothesis 2 was not supported by the obtained results, since no statistically significant differences were observed between the learning behavior of the evaluated interfaces during the experimental repetitions.

Hypothesis 3 was partially supported. Although the immersive interface enabled significantly faster task execution, it also increased the number of grasp attempts, while

collision results did not present statistically significant differences between conditions.

Hypothesis 4 was partially supported. NASA-TLX and SUS did not reveal substantial differences between immersive and conventional teleoperation conditions. However, UEQ-S identified significantly higher Hedonic Quality scores for the immersive interface, indicating more positive subjective perceptions associated with immersive interaction.

8.3 LIMITATIONS

Some limitations should be considered when interpreting the results obtained in this dissertation. The experimental evaluation was conducted under a specific telemanipulation scenario involving the participant sample described in Section 5.3 and a single manipulation task presented in Section 5.4. Although the adopted experimental protocol was adequate for the comparative analyses performed in this study, the obtained results should not be interpreted as universally representative of all teleoperation conditions, operator profiles, or industrial applications.

Another limitation involves the short-term nature of the experimental sessions. The study evaluated participant interaction during relatively brief teleoperation activities, without investigating long-term adaptation effects, prolonged use, or behavioral changes associated with extended interaction periods.

Important limitations were also associated with the developed teleoperation platform itself. The automatic planning-execution workflow introduced interruptions and small delays between user commands and robot motion, negatively affecting interaction fluidity during precise positioning tasks. In addition, some participants experienced difficulties related to the interaction mapping adopted in the immersive interface, particularly involving confusion between the controller commands used for gripper actuation and robot manipulation, which occasionally resulted in unintended object releases during task execution.

Another relevant limitation involves the operational variability observed during the experimental repetitions. In several cases, execution times were influenced by isolated operational events, such as alignment difficulties or unintended object releases during specific trials. Consequently, part of the observed variability may be associated not only with participant adaptation or learning behavior, but also with the inherent operational variability of telemanipulation tasks.

8.4 FUTURE WORK

Future work may investigate the proposed teleoperation system under different manipulation tasks, long term application, and distinct operator profiles in order to

provide a broader evaluation of immersive interaction under different teleoperation conditions and operational demands.

Possible improvements to the developed platform include the implementation of point cloud visualization within the MR environment, potentially improving depth perception and immersion during teleoperation tasks. In addition, dynamic robot speed adjustment mechanisms integrated into the teleoperation interface may allow finer positioning control through proportional adaptation of robot motion according to the precision requirements of the task.

REFERENCES

AI, Letian; KAZANZIDES, Peter; AZIMI, Ehsan. Mixed reality based teleoperation and visualization of surgical robotics. **Healthcare Technology Letters**, Wiley Online Library, v. 11, n. 2-3, p. 179–188, 2024.

ALBERT, Bill; TULLIS, Tom. **Measuring the User Experience: Collecting, Analyzing, and Presenting UX Metrics**. 3. ed. Burlington, MA: Morgan Kaufmann, 2022.

ALROOBAEA, Roobaea; MAYHEW, Pam J. How Many Participants Are Really Enough for Usability Studies? *In:* 2014 Science and Information Conference (SAI). London, UK: IEEE, 2014. p. 48–56.

ANDERSON, Robert J.; SPONG, Mark W. Bilateral Control of Teleoperators with Time Delay. *In:* PROCEEDINGS of the 1988 IEEE International Conference on Systems, Man, and Cybernetics. Beijing, China: IEEE, 1988. v. 1, p. 131–138.

AL-ANSI, Abdullah M; JABOOB, Mohammed; GARAD, Askar; AL-ANSI, Ahmed. Analyzing augmented reality (AR) and virtual reality (VR) recent development in education. **Social Sciences & Humanities Open**, Elsevier, v. 8, n. 1, p. 100532, 2023.

APRAIZ, Ainhoa; LASA, Ganix; MAZMELA, Maitane. Evaluation of user experience in human–robot interaction: a systematic literature review. **International Journal of Social Robotics**, Springer, v. 15, n. 2, p. 187–210, 2023.

ARDUINO SA. **Arduino Uno**. 2024. Available from: `https://docs.arduino.cc/hardware/uno-rev3`. Visited on: 24 Oct. 2025.

ARLATI, Sara; SPOLADORE, Daniele; BALDASSINI, Davide; SACCO, Marco; GRECI, Luca. VirtualCruiseTour: An AR/VR Application to Promote Shore Excursions on Cruise Ships. *In:* AUGMENTED Reality, Virtual Reality, and Computer Graphics. AVR 2018. Cham, Switzerland: Springer, 2018. v. 10851. (Lecture Notes in Computer Science), p. 133–147.

AVGOUSTI, Sotiris; CHRISTOFOROU, Eftychios G; PANAYIDES, Andreas S; VOSKARIDES, Sotos; NOVALES, Cyril; NOUAILLE, Laurence; PATTICHIS, Constantinos S; VIEYRES, Pierre. Medical telerobotic systems: current status and future trends. **Biomedical engineering online**, Springer, v. 15, p. 1–44, 2016.

BANGOR, Aaron; KORTUM, Philip; MILLER, James. Determining what individual SUS scores mean: Adding an adjective rating scale. **Journal of usability studies**, Usability Professionals' Association Bloomingdale, IL, v. 4, n. 3, p. 114–123, 2009.

BECHTEL, Nicolai; WEBER, Bernhard; SEVERIN, Pascal; SANCHO ARAGON, Jaime; VAN BOGAERT, Laurie; PANZIRSCH, Michael. Toward Physically Realistic Vision in Teleoperation: A User Study with Light-Field Head Mounted Display and 6-DoF Head Motion. **Journal of the Society for Information Display**, Wiley, v. 31, n. 12, p. 663–674, 2023.

BERNS, Karsten *et al.* Climbing Robot Inspired by Inchworms: Adaptable for Tubular and Flat Surfaces with Multi-Plane Work Capability*. In:* CLIMBING and Walking Robots. Cham, Switzerland: Springer, 2024. v. 1114. (Lecture Notes in Networks and Systems), p. 211–223.

BOER, Thomas Anton Bob de. **The effects of situated visualized telepresence on task performance in gesture-based teleoperation**. 2021. PhD thesis – MSc thesis). Delft University of Technology. http://resolver.tudelft.nl/uuid:9449df05-ea7f-4adf-8d36-113708fb4bdc.

BROOKE, John *et al.* SUS-A quick and dirty usability scale. **Usability evaluation in industry**, London, England, v. 189, n. 194, p. 4–7, 1996.

BYERS, James C; BITTNER, Arthur C; HILL, Stephen G. Traditional and raw task load index (TLX) correlations: are paired comparisons necessary? *In:* MITAL, Anil (ed.). **Advances in Industrial Ergonomics and Safety I**. New York: Taylor & Francis, 1989. p. 481–485.

CABRERA, Miguel Altamirano; RAKHMATULIN, Viktor; FEDOSEEV, Aleksey; SAUTENKOV, Oleg; ALYOUNES, Oussama; PUCHKOV, Andrei; TSETSERUKOU, Dzmitry. OmniCharger: CNN-Based hand gesture interface to operate an electric car charging robot through teleconference. **ACM Transactions on Human-Robot Interaction**, ACM New York, NY, v. 13, n. 3, p. 1–18, 2024.

CAMBUZAT, Remi; ELISEI, Frédéric; BAILLY, Gérard; SIMONIN, Olivier; SPALANZANI, Anne. Immersive Teleoperation of the Eye Gaze of Social Robots: Assessing Gaze-Contingent Control of Vergence, Yaw and Pitch of Robotic Eyes*. In:* PROCEEDINGS of the 50th International Symposium on Robotics (ISR 2018). Munich, Germany: VDE, 2018. p. 1–8.

CANDELORO, Mauro; VALLE, Eirik; MIYAZAKI, Michel R.; SKJETNE, Roger; LUDVIGSEN, Martin; SØRENSEN, Asgeir J. HMD as a New Tool for Telepresence in Underwater Operations and Closed-Loop Control of ROVs*. In:* OCEANS 2015 MTS/IEEE Washington. Washington, DC, USA: IEEE, 2015. p. 1–8.

CHOI, KyuSang; CHO, JangHo; HAN, JiWoong; CHOI, HyeunSeok; YANG, Gi-Hun. Space Teleoperation Scheme for Korean Lunar Rover*. In:* 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). Jeju, South Korea: IEEE, 2017. p. 706–709.

COHEN, Jacob. **Statistical Power Analysis for the Behavioral Sciences**. 2nd. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

COLEMAN, D.M.; SUCAN, Ioan Alexandru; CHITTA, Sachin; CORRELL, Nikolaus. Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. **ArXiv**, abs/1404.3785, 2014.

COLGATE, J. Edward; WANNASUPHOPRASIT, Witaya; PESHKIN, Michael A. Cobots: Robots for Collaboration with Human Operators*. In:* PROCEEDINGS of the ASME

International Mechanical Engineering Congress and Exposition. Atlanta, GA, USA: American Society of Mechanical Engineers, 1996. v. 58. (Dynamic Systems and Control Division), p. 433–439.

CONDORI, Victor; CASTILLO, Jaime; MAMANI, Alfredo; PARI, Lizardo. EVALUATION OF AN IMMERSIVE VIRTUAL REALITY BASED POSITION CONTROL INTERFACE FOR A SCARA ROBOTIC ARM. **Eastern-European Journal of Enterprise Technologies**, v. 4, 2025.

CORNEJO, Jair; DENEGRI, Eddy; VASQUEZ, Karina; RAMOS, Oscar E. Real-time Joystick Teleoperation of the Sawyer Robot Using a Numerical Approach*. In:* 2018 IEEE ANDESCON. Santiago de Cali, Colombia: IEEE, 2018. p. 1–3.

COUNCIL, National Research *et al.* **Virtual reality: scientific and technological challenges**. Washington, DC: National Academies Press, 1994.

CRUZ ULLOA, Christyan; DOMÍNGUEZ, David; CERRO, Jaime del; BARRIENTOS, Antonio. Analysis of mr–vr tele-operation methods for legged-manipulator robots. **Virtual Reality**, Springer, v. 28, n. 3, p. 131, 2024.

DARVISH, Kourosh; PENCO, Luigi; RAMOS, Joao; CISNEROS, Rafael; PRATT, Jerry; YOSHIDA, Eiichi; IVALDI, Serena; PUCCI, Daniele. Teleoperation of humanoid robots: A survey. **IEEE Transactions on Robotics**, IEEE, v. 39, n. 3, p. 1706–1727, 2023.

DAVIS, Fred D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. **MIS quarterly**, JSTOR, p. 319–340, 1989.

ELEPHANT ROBOTICS. **Mygripper F100**. Available from: `https://shop.elephantrobotics.com/products/mygripper-f100`. Visited on: 29 Aug. 2025.

ELEPHANT ROBOTICS. **Pacote ROS - myGripper F100**. Available from: `https://github.com/elephantrobotics/pro_gripper_ros`. Visited on: 6 Nov. 2025.

ELOR, Aviv *et al.* Catching Jellies in Immersive Virtual Reality: A Comparative Teleoperation Study of ROVs in Underwater Capture Tasks*. In:* PROCEEDINGS of the 27th ACM Symposium on Virtual Reality Software and Technology. Osaka, Japan: Association for Computing Machinery, 2021. p. 1–10.

ELSEVIER. **Scopus Database**. 2026. Available from: `https://www.scopus.com`. Visited on: 2 Apr. 2026.

ERICVOLL. **RobotDynamicsLibrary: uma biblioteca em C# para cinemática e dinâmica de robôs**. 2021. Available from: `https://github.com/EricVoll/RobotDynamicsLibrary?tab=readme-ov-file`. Visited on: 7 Nov. 2025.

ESAKI, Hibiki; SEKIYAMA, Kosuke. Immersive Robot Teleoperation Based on User Gestures in Mixed Reality Space. **Sensors**, MDPI, v. 24, n. 15, p. 5073, 2024.

FAHIMI, Farbod. **Autonomous Robots**. New York, NY: Springer, 2009.

FANUC. **Controlador R-30iB**. Available from: https://www.fanuc.eu/eu-en/accessory/controller/r-30ib-plus-mate-cabinet. Visited on: 29 Aug. 2025.

FANUC. **FANUC VISÃO GERAL**. Available from: https://www.fanucamerica.com/products/robots/series/lr-mate/lr-mate-200id-7l. Visited on: 28 Aug. 2025.

FANUC. **RoboGuide**. Available from: https://www.fanucamerica.com/products/robots/roboguide. Visited on: 9 Sept. 2025.

FANUC. **Technical Specifications - LR Mate 200iD/7L**. Available from: https://www.fanucamerica.com/docs/default-source/robotics-files/lr-mate/lr-mate-200id-7l-data-sheet.pdf. Visited on: 29 Aug. 2025.

FARAJIPARVAR, Parinaz; YING, Hao; PANDYA, Abhilash. A brief survey of telerobotic time delay mitigation. **Frontiers in Robotics and AI**, Frontiers Media SA, v. 7, p. 578805, 2020.

FERRE, Manuel; BUSS, Martin; ARACIL, Rafael; MELCHIORRI, Claudio; BALAGUER, Carlos. Introduction to Advances in Telerobotics. *In:* ADVANCES in Telerobotics. Berlin, Heidelberg: Springer, 2007. v. 31. (Springer Tracts in Advanced Robotics). p. 1–7.

FERRELL, William R.; SHERIDAN, Thomas B. Supervisory control of remote manipulation. **IEEE Spectrum**, v. 4, n. 10, p. 81–88, 1967.

GALARZA, Bryan R; AYALA, Paulina; MANZANO, Santiago; GARCIA, Marcelo V. Virtual reality teleoperation system for mobile robot manipulation. **Robotics**, MDPI, v. 12, n. 6, p. 163, 2023.

GALLIPOLI, Marco; BUONOCORE, Sara; SELVAGGIO, Mario; FONTANELLI, Giuseppe Andrea; GRAZIOSO, Stanislao; DI GIRONIMO, Giuseppe. A virtual reality-based dual-mode robot teleoperation architecture. **Robotica**, Cambridge University Press, v. 42, n. 6, p. 1935–1958, 2024.

GARG, Gaurav; KUTS, Vladimir; ANBARJAFARI, Gholamreza. Digital twin for fanuc robots: Industrial robot programming and simulation using virtual reality. **Sustainability**, MDPI, v. 13, n. 18, p. 10336, 2021.

GIHLEB, Rania; GIUNTELLA, Osea; STELLA, Luca; WANG, Tianyi. Industrial robots, workers' safety, and health. **Labour economics**, Elsevier, v. 78, p. 102205, 2022.

GIL, Antonio Carlos. **Métodos e Técnicas de Pesquisa Social**. 6. ed. São Paulo: Atlas, 2008.

GLOUMEAU, Paul Christopher; PETTIFER, Stephen Robert. Exploring Visual Conditions in Virtual Reality for the Teleoperation of Robots. *In:* PROCEEDINGS of the 30th ACM Symposium on Virtual Reality Software and Technology. Trier, Germany: Association for Computing Machinery, 2024. p. 1–10.

GONÇALVES, Rogério Sales; SOUZA, Frederico Costa; HOMMA, Rafael Zimmermann; SUDBRACK, Daniel Edgardo Tio; TRAUTMANN, Paulo Victor; CLASEN, Bruno Cordeiro. Mobile Robot for Debris Removal from High Voltage Power Lines. *In:* 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE). São Bernardo do Campo, SP, Brazil: IEEE, 2022. p. 1–5.

GOURLEY, David; TOTTY, Brian. **HTTP: The Definitive Guide**. Sebastopol, CA: O'Reilly Media, 2002.

GRIER, Rebecca A. How High Is High? A Meta-Analysis of NASA-TLX Global Workload Scores. *In:* 1. PROCEEDINGS of the Human Factors and Ergonomics Society Annual Meeting. Los Angeles, CA: SAGE Publications, 2015. v. 59, p. 1727–1731.

HAGHJOO, Fatemeh. **Usability Evaluation of Emerging Virtual Reality Technologies in Telerobotics**. 2018. PhD thesis – University of British Columbia.

HART, Sandra G. NASA-Task Load Index (NASA-TLX): 20 Years Later. *In:* 9. PROCEEDINGS of the Human Factors and Ergonomics Society Annual Meeting. Los Angeles, CA: SAGE Publications, 2006. v. 50, p. 904–908.

HART, Sandra G.; STAVELAND, Lowell E. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *In:* HUMAN Mental Workload. Amsterdam: Elsevier, 1988. v. 52. (Advances in Psychology). p. 139–183.

HASAN, Shaid; YASAR, Mohammad Samin; IQBAL, Tariq. A Holistic Evaluation of Teleoperation Interfaces for Robotic Manipulation. **ACM Transactions on Human-Robot Interaction**, ACM New York, NY, v. 15, n. 2, p. 1–42, 2026.

HASSENZAHL, Marc. The effect of perceived hedonic quality on product appealingness. **International Journal of Human-Computer Interaction**, Taylor & Francis, v. 13, n. 4, p. 481–499, 2001.

HERNANDEZ, Matheus Nicolás; MIRANDA, Nathália C. de; DOMINGUES, Brenno H. R.; NEGRI, Doglas; SOUZA, Diego de; SECCO, Ismael; TRABASSO, Luís Gonzaga; SIMONI, Roberto. Influence of User Profiles on Usability Perception in Virtual Reality-Based Robotic Teleoperation. *In:* 2025 27th Symposium on Virtual and Augmented Reality (SVR). Salvador, BA, Brazil: IEEE, 2025. p. 156–164.

HERNANDEZ, Matheus Nicolás; RAMOS, Brenno Henrique; SIMONI, Roberto; NEGRI, Doglas; SOUZA, Diego de; TRABASSO, Luís Gonzaga. Robot Teleoperation

via Virtual Reality: A Unity and ROS Approach. *In:* 2024 3rd International Conference on Automation, Robotics and Computer Engineering (ICARCE). China: IEEE, 2024. p. 26–29.

HERNANDEZ-GOBERTTI, Fernando; KUDYK, Ivan D; LOZANO, Raul; NGUYEN, Giang T; GOMEZ-BARQUERO, David. Immersive Teleoperation via Collaborative Device-Agnostic Interfaces for Smart Haptics: A Study on Operational Efficiency and Cognitive Overflow for Industrial Assistive Applications. **Sensors**, MDPI, v. 25, n. 13, p. 3993, 2025.

HINDERKS, Andreas. Design and evaluation of a short version of the user experience questionnaire (UEQ-S). **International Journal of Interactive Multimedia and Artificial Intelligence**, 2017.

HODRIEN, Andrew; FERNANDO, Terrence. A Review of Post-Study and Post-Task Subjective Questionnaires to Guide Assessment of System Usability. **Journal of Usability Studies**, v. 16, n. 3, 2021.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics 2025 – Industrial Robots**. Available from: `https://ifr.org/downloads/press_docs/PressConference2025_presentation.pdf`. Visited on: 25 Sept. 2025.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics 2025 Report – Industrial Robots: Global Robot Demand in Factories Doubles Over 10 Years**. Frankfurt am Main, Germany: International Federation of Robotics, 2025. Available from: `https://ifr.org/ifr-press-releases/news/global-robot-demand-in-factories-doubles-over-10-years`.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 9241-11:2018 — Ergonomics of human–system interaction — Part 11: Usability: Definitions and concepts**. 2018. Available from: `https://www.iso.org/standard/63500.html`. Visited on: 4 Nov. 2025.

ISLAM, Shafiqul; DIAS, Jorge; SUNDA-MEYA, Anderson. On the Design and Development of Vision-Based Autonomous Mobile Manipulation. *In:* IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society. Toronto, ON, Canada: IEEE, 2021. p. 1–6.

JANKOWSKI, Jarosław; GRABOWSKI, Andrzej. Usability evaluation of vr interface for mobile robot teleoperation. **International Journal of Human-Computer Interaction**, Taylor & Francis, v. 31, n. 12, p. 882–889, 2015.

JOSHI, Ankur; KALE, Saket; CHANDEL, Satish; PAL, D Kumar. Likert scale: Explored and explained. **British journal of applied science & technology**, Sciencedomain International, v. 7, n. 4, p. 396, 2015.

KANAZAWA, Kotaro; SATO, Noritaka; MORITA, Yoshifumi. Usability Evaluation of Human–Robot Interaction in Manipulator Teleoperation via Virtual Reality in Inspection

Tasks. *In:* 2024 13th International Workshop on Robot Motion and Control (RoMoCo). Poznań, Poland: IEEE, 2024. p. 211–216.

KEBRIA, Parham M; ABDI, Hamid; DALVAND, Mohsen Moradi; KHOSRAVI, Abbas; NAHAVANDI, Saeid. Control methods for internet-based teleoperation systems: A review. **IEEE Transactions on Human-Machine Systems**, IEEE, v. 49, n. 1, p. 32–46, 2018.

KEBRIA, Parham M.; KHOSRAVI, Abbas; NAHAVANDI, Saeid; HOMAIFAR, Abdollah; SAIF, Mehrdad. Experimental Comparison Study on Joint and Cartesian Space Control Schemes for a Teleoperation System Under Time-Varying Delay. *In:* 2019 IEEE International Conference on Industrial Technology (ICIT). Melbourne, VIC, Australia: IEEE, 2019. p. 108–113.

KIM, Seungwoon; KIM, Yihan; HA, Jeesoo; JO, Sungho. Mapping System with Virtual Reality for Mobile Robot Teleoperation. *In:* 2018 18th International Conference on Control, Automation and Systems (ICCAS). PyeongChang, South Korea: IEEE, 2018. p. 1541–1541.

KLINE, Paul. **A Handbook of Test Construction: Introduction to Psychometric Design**. London: Routledge, 2015. (Psychology Revivals).

KUJALA, Sari. User involvement: a review of the benefits and challenges. **Behaviour & information technology**, Taylor & Francis, v. 22, n. 1, p. 1–16, 2003.

LAU, Francis; HOLBROOK, Anne. Methods for Comparative Studies. *In:* LAU, Francis; KUZIEMSKY, Craig (eds.). **Handbook of eHealth Evaluation: An Evidence-based Approach**. Victoria, BC, Canada: University of Victoria, 2017. p. 181–198.

LAUGWITZ, Bettina; HELD, Theo; SCHREPP, Martin. Construction and Evaluation of a User Experience Questionnaire. *In:* HOLZINGER, Andreas (ed.). **HCI and Usability for Education and Work**. Berlin, Heidelberg: Springer, 2008. v. 5298. (Lecture Notes in Computer Science), p. 63–76.

LAWRENCE, Dale A. Stability and transparency in bilateral teleoperation. **IEEE transactions on robotics and automation**, IEEE, v. 9, n. 5, p. 624–637, 1993.

LAZAR, Jonathan; FENG, Jinjuan Heidi; HOCHHEISER, Harry. **Research Methods in Human-Computer Interaction**. 2. ed. Cambridge, MA: Morgan Kaufmann, 2017.

LEMASURIER, Gregory *et al.* Comparing a 2D Keyboard and Mouse Interface to Virtual Reality for Human-in-the-Loop Robot Planning for Mobile Manipulation. *In:* 2024 33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). Pasadena, CA, USA: IEEE, 2024. p. 2197–2203.

LEWIS, James R. The system usability scale: past, present, and future. **International Journal of Human–Computer Interaction**, Taylor & Francis, v. 34, n. 7, p. 577–590, 2018.

LIKERT, Rensis. A technique for the measurement of attitudes. **Archives of psychology**, 1932.

LIMA, Amanda Tonel; ROCHA, Filipe Augusto Santos; TORRE, Marcos Paulo; AZPÚRUA, Héctor; FREITAS, Gustavo Medeiros. Teleoperation of an ABB IRB 120 Robotic Manipulator and BarrettHand BH8-282 Using a Geomagic Touch X Haptic Device and ROS. *In:* 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). João Pessoa, PB, Brazil: IEEE, 2018. p. 188–193.

MACEFIELD, Ritch. How to specify the participant group size for usability studies: a practitioner's guide. **Journal of usability studies**, Usability Professionals' Association Bloomingdale, IL, v. 5, n. 1, p. 34–45, 2009.

MARTYNOV, Mikhail; DARUSH, Zhanibek; CABRERA, Miguel Altamirano; KARAF, Sausar; TSETSERUKOU, Dzmitry. MorphoArms: Morphogenetic Teleoperation of Multimanual Robot. *In:* 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE). Auckland, New Zealand: IEEE, 2023. p. 1–6.

MCMAHAN, Ryan P; BOWMAN, Doug A; ZIELINSKI, David J; BRADY, Rachael B. Evaluating display fidelity and interaction fidelity in a virtual reality game. **IEEE transactions on visualization and computer graphics**, IEEE, v. 18, n. 4, p. 626–633, 2012.

META. **Conectar com app Meta Quest Link**. Available from: `https://www.meta.com/help/quest/509273027107091/?srsltid=AfmBOopNAwbmk1cw6MbPuDcnQLepixW3_b3XrmL6mHj9_vTUguA-Osqy`. Visited on: 18 Sept. 2025.

META. **Instalar app Meta Quest Link**. Available from: `https://www.meta.com/help/quest/articles/headsets-and-accessories/oculus-rift-s/install-app-for-link/`. Visited on: 8 Nov. 2024.

META. **Meta Quest 3**. Available from: `https://www.meta.com/quest/quest-3/`. Visited on: 29 Aug. 2025.

MILGRAM, Paul; KISHINO, Fumio. A taxonomy of mixed reality visual displays. **IEICE TRANSACTIONS on Information and Systems**, The Institute of Electronics, Information and Communication Engineers, v. 77, n. 12, p. 1321–1329, 1994.

MONDELLINI, Marta; ARLATI, Sara; GAPEYEVA, Helena; LEES, Kairi; MÄRITZ, Ingrid; PIZZAGALLI, Simone Luca; OTTO, Tauno; SACCO, Marco; TEDER-BRASCHINSKY, Anneli. User experience during an immersive virtual reality-based cognitive task: a comparison between estonian and italian older adults with MCI. **Sensors**, MDPI, v. 22, n. 21, p. 8249, 2022.

MONFORT, Samuel S; GRAYBEAL, John J; HARWOOD, Amanda E; MCKNIGHT, Patrick E; SHAW, Tyler H. A single-item assessment for remaining mental

resources: development and validation of the Gas Tank Questionnaire (GTQ). **Theoretical Issues in Ergonomics Science**, Taylor & Francis, v. 19, n. 5, p. 530–552, 2018.

MORTAZAVI, Ehsan; DOYON-POULIN, Philippe; IMBEAU, Daniel; TARAGHI, Mitra; ROBERT, Jean-Marc. Exploring the landscape of UX subjective evaluation tools and UX dimensions: A Systematic Literature Review (2010–2021). **Interacting with Computers**, Oxford University Press, v. 36, n. 4, p. 255–278, 2024.

MPT; OIT. **Observatório Digital de Saúde e Segurança do Trabalho**. Available from: `https://smartlabbr.org/sst`. Visited on: 23 Oct. 2024.

MUSCOLO, Giovanni Gerardo; MARCHESCHI, Simone; FONTANA, Marco; BERGAMASCO, Massimo. Dynamics modeling of human–machine control interface for underwater teleoperation. **Robotica**, Cambridge University Press, v. 39, n. 4, p. 618–632, 2021.

NANDA, Vibhav; MADDALI, Hanuma Teja; LAZAR, Amanda. Does XR Introduce Experience Asymmetry in an Intergenerational Setting? *In:* PROCEEDINGS of the 24th International ACM SIGACCESS Conference on Computers and Accessibility. Athens, Greece: Association for Computing Machinery, 2022. p. 1–4.

NEGRI, Doglas; FUSINATO, Amanda; FARIA, Felipe; LUZ, Valéria; MOSER, Thiago; SECCO, Ismael; TRABASSO, Luís Gonzaga. Analysis of the Influence of Human Exposure to Risk and ESG as Motivators for the Implementation of Climbing and Mobile Robots. *In:* SYNERGETIC Cooperation Between Robots and Humans. Cham, Switzerland: Springer Nature Switzerland, 2023. (Lecture Notes in Networks and Systems), p. 9–20.

NIEMEYER, Günter; PREUSCHE, Carsten; STRAMIGIOLI, Stefano; LEE, Dongjun. Telerobotics. *In:* SICILIANO, Bruno; KHATIB, Oussama (eds.). **Springer Handbook of Robotics**. 2. ed. Cham, Switzerland: Springer, 2016. p. 1085–1108.

NIEMEYER, Günter; SLOTINE, Jean-Jacques E. Telemanipulation with time delays. **The International Journal of Robotics Research**, SAGE Publications, v. 23, n. 9, p. 873–890, 2004.

OMARALI, Bukeikhan. **Exploring robot teleoperation in virtual reality**. 2023. PhD thesis – QUEEN MARY UNIVERSITY OF LONDON UNIVERSITY OF GENOA.

PARZIALE, Lydia; LIU, Wei; MATTHEWS, Carolyn; ROSSELOT, Nicolas; DAVIS, Chuck; FORRESTER, Jason; BRITT, David T, *et al.* TCP/IP tutorial and technical overview. IBM Redbooks, 2006.

PAVLOVA, Galya. **Software platform for a modular collaborative robot: Simulating a robot in augmented reality**. 2023. PhD thesis – Technische Universität Wien.

PIZZAGALLI, Simone Luca; BONDARENKO, Yevhen; KUTS, Vladimir; O'CONNELL, Eoin; MURRAY, Niall; OTTO, Tauno. ROS-based Augmented and Virtual

Reality Path Planning Interface for Industrial Robotic Arms: A Preliminary Assessment. *In:* PROCEEDINGS of the 2023 6th International Conference on Robot Systems and Applications. Stockholm, Sweden: Association for Computing Machinery, 2023. p. 38–48.

POLLÁK, Martin; KOČIŠKO, Marek; PAULIŠIN, Dušan; BARON, Petr. Measurement of unidirectional pose accuracy and repeatability of the collaborative robot UR5. **Advances in Mechanical Engineering**, SAGE Publications Sage UK: London, England, v. 12, n. 12, p. 1687814020972893, 2020.

PRYOR, Will; WANG, Liam J.; CHATTERJEE, Arko; VAGVOLGYI, Balazs P.; DEGUET, Anton; LEONARD, Simon; WHITCOMB, Louis L.; KAZANZIDES, Peter. A Virtual Reality Planning Environment for High-Risk, High-Latency Teleoperation. *In:* 2023 IEEE International Conference on Robotics and Automation (ICRA). London, United Kingdom: IEEE, 2023. p. 11619–11625.

R CORE TEAM. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2026. Available from: `https://www.R-project.org/`.

RASTEGARPANAH, Alireza; CONTRERAS, Cesar Alan; STOLKIN, Rustam. Semi-autonomous Robotic Disassembly Enhanced by Mixed Reality. *In:* SONG, Aiguo; DING, Zhengtao; NAYYAR, Anand; SONG, Ran (eds.). **RobCE '24: Proceedings of the 2024 4th International Conference on Robotics and Control Engineering**. Edinburgh, United Kingdom: Association for Computing Machinery, 2024. p. 7–13.

RAUSCHNABEL, Philipp A; FELIX, Reto; HINSCH, Chris; SHAHAB, Hamza; ALT, Florian. What is XR? Towards a framework for augmented and virtual reality. **Computers in human behavior**, Elsevier, v. 133, p. 107289, 2022.

ROS WIKI. **ROS Introduction – ROS Wiki**. Available from: `http://wiki.ros.org/ROS/Introduction`. Visited on: 21 Aug. 2025.

ROS WIKI. **ROS Noetic – ROS Wiki**. Available from: `https://wiki.ros.org/noetic`. Visited on: 2 Sept. 2025.

ROS WIKI. **Rviz – ROS Wiki**. Available from: `https://wiki.ros.org/rviz`. Visited on: 9 Sept. 2025.

ROS WIKI. **Tutoriais FANUC-ROS – ROS Wiki**. Available from: `https://wiki.ros.org/fanuc/Tutorials/`. Visited on: 10 Sept. 2025.

ROS WIKI. **Tutoriais: Configuração Drivers FANUC-ROS – ROS Wiki**. Available from: `https://wiki.ros.org/fanuc/Tutorials/hydro/Configuration`. Visited on: 10 Sept. 2025.

ROS WIKI. **Tutoriais: Instalação Drivers FANUC-ROS – ROS Wiki**. Available from: `https:`

//wiki.ros.org/fanuc/Tutorials/hydro/Installation#Copying_the_Binaries. Visited on: 10 Sept. 2025.

ROS WIKI. **URDF – ROS Wiki**. Available from: https://wiki.ros.org/urdf. Visited on: 21 Aug. 2025.

ROS-INDUSTRIAL. **ROS-INDUSTRIAL - FANUC GitHub**. Available from: https://github.com/ros-industrial. Visited on: 9 Sept. 2025.

ROS-INDUSTRIAL. **ROS-INDUSTRIAL - FANUC-Driver GitHub**. Available from: https://github.com/ros-industrial/fanuc/tree/noetic-devel/fanuc_driver. Visited on: 10 Sept. 2025.

ROS-INDUSTRIAL. **ROS-Industrial Visão Geral**. Available from: https://rosindustrial.org/. Visited on: 8 Nov. 2024.

SAURO, Jeff; LEWIS, James R. **Quantifying the User Experience: Practical Statistics for User Research**. 2. ed. Cambridge, MA: Morgan Kaufmann, 2016.

SCALERA, Lorenzo; SERIANI, Stefano; GALLINA, Paolo; DI LUCA, Massimiliano; GASPARETTO, Alessandro. Experimental Evaluation of Vibrotactile Training Mappings for Dual-Joystick Directional Guidance. *In:* PRATTICHIZZO, Domenico; SHINODA, Hiroyuki; TAN, Hong Z.; RUFFALDI, Emanuele; FRISOLI, Antonio (eds.). **Haptics: Science, Technology, and Applications**. Cham, Switzerland: Springer, 2018. v. 10894. (Lecture Notes in Computer Science), p. 575–586.

SHAO, Shuyu; ZHOU, Qianxiang; LIU, Zhongqi. Mental workload characteristics of manipulator teleoperators with different spatial cognitive abilities. **International journal of advanced robotic systems**, SAGE Publications Sage UK: London, England, v. 16, n. 6, p. 1729881419888042, 2019.

SHAPIRO, Samuel Sanford; WILK, Martin B. An analysis of variance test for normality (complete samples). **Biometrika**, Oxford University Press, v. 52, n. 3-4, p. 591–611, 1965.

SHARP CORPORATION. **TOF10120 Time-of-Flight Ranging Sensor — Datasheet**. Available from: https://www.mikrocontroller.net/attachment/618619/TOF10120___-1.pdf. Visited on: 24 Oct. 2025.

SHERIDAN, Thomas B. Telerobotics. **Automatica**, Elsevier, v. 25, n. 4, p. 487–507, 1989.

SHUKLA, Amit; KARKI, Hamad. Application of robotics in onshore oil and gas industry—A review Part I. **Robotics and Autonomous Systems**, Elsevier, v. 75, p. 490–507, 2016.

SICILIANO, Bruno; KHATIB, Oussama. Robotics and the Handbook. *In:* SICILIANO, Bruno; KHATIB, Oussama (eds.). **Springer Handbook of Robotics**. 2. ed. Cham, Switzerland: Springer, 2016. p. 1–6.

SICILIANO, Bruno; SCIAVICCO, Lorenzo; VILLANI, Luigi; ORIOLO, Giuseppe. **Robotics: Modelling, Planning and Control**. 1st. London: Springer, 2009.

SIEGWART, Roland; NOURBAKHSH, Illah Reza; SCARAMUZZA, Davide. **Introduction to Autonomous Mobile Robots**. 2. ed. Cambridge, MA: The MIT Press, 2011.

SKARBEZ, Richard; SMITH, Missie; WHITTON, Mary C. Revisiting Milgram and Kishino's reality-virtuality continuum. **Frontiers in Virtual Reality**, Frontiers Media SA, v. 2, p. 647997, 2021.

SOLANES, J Ernesto; MUÑOZ, Adolfo; GRACIA, Luis; TORNERO, Josep. Virtual reality-based interface for advanced assisted mobile robot teleoperation. **Applied Sciences**, MDPI, v. 12, n. 12, p. 6071, 2022.

STOTKO, Patrick; KRUMPEN, Stefan; SCHWARZ, Max; LENZ, Christian; BEHNKE, Sven; KLEIN, Reinhard; WEINMANN, Michael. A VR System for Immersive Teleoperation and Live Exploration with a Mobile Robot. *In:* 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Macau, China: IEEE, 2019. p. 3630–3637.

SU, Yun-Peng; CHEN, Xiao-Qi; ZHOU, Cong; PEARSON, Lui Holder; PRETTY, Christopher G; CHASE, J Geoffrey. Integrating Virtual, Mixed, and Augmented Reality into Remote Robotic Applications: A Brief Review of Extended Reality-Enhanced Robotic Systems for Intuitive Telemanipulation and Telemanufacturing Tasks in Hazardous Conditions. **Applied Sciences**, MDPI, v. 13, n. 22, p. 12129, 2023.

SU, Yun-Peng; CHEN, Xiao-Qi; ZHOU, Cong; PEARSON, Lui Holder; PRETTY, Christopher G; CHASE, J Geoffrey. Integrating virtual, mixed, and augmented reality into remote robotic applications: A brief review of extended reality-enhanced Robotic systems for Intuitive Telemanipulation and Telemanufacturing tasks in Hazardous conditions. **Applied Sciences**, MDPI, v. 13, n. 22, p. 12129, 2023.

SU, Yunpeng; CHEN, Xiaoqi; ZHOU, Tony; PRETTY, Christopher; CHASE, Geoffrey. Mixed reality-integrated 3D/2D vision mapping for intuitive teleoperation of mobile manipulator. **Robotics and Computer-Integrated Manufacturing**, Elsevier, v. 77, p. 102332, 2022.

SU, Yunpeng; LLOYD, Leo; CHEN, Xiaoqi; CHASE, J Geoffrey. Latency mitigation using applied HMMs for mixed reality-enhanced intuitive teleoperation in intelligent robotic welding. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 126, n. 5, p. 2233–2248, 2023.

SUN, Da; KISELEV, Andrey; LIAO, Qianfang; STOYANOV, Todor; LOUTFI, Amy. A new mixed-reality-based teleoperation system for telepresence and maneuverability enhancement. **IEEE Transactions on Human-Machine Systems**, IEEE, v. 50, n. 1, p. 55–67, 2020.

TAVAKKOLI, Alireza; WILSON, Brandon; BOUNDS, Matthew. An Immersive Virtual Environment for Teleoperation of Remote Robotic Agents for Everyday Applications in Prohibitive Environments*. In:* 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). Atlanta, GA, USA: IEEE, 2020. p. 371–375.

TOM STÖVEKEN. **mjpg-streamer**. 2021. Available from: `https://github.com/jacksonliam/mjpg-streamer?tab=readme-ov-file`. Visited on: 12 Nov. 2025.

TRIANTAFYLLIDIS, Eleftherios; MCGREAVY, Christopher; GU, Jiacheng; LI, Zhibin. Study of multimodal interfaces and the improvements on teleoperation. **IEEE Access**, IEEE, v. 8, p. 78213–78227, 2020.

UDEKWE, Daniel; SEYYEDHASANI, Hasan. Human robot interaction for agricultural Tele-Operation, using virtual Reality: A feasibility study. **Computers and Electronics in Agriculture**, Elsevier, v. 228, p. 109702, 2025.

UNITY TECHNOLOGIES. **Logo Unity**. Available from: `https://unity.com/es/legal/branding-trademarks`. Visited on: 2 Sept. 2025.

UNITY TECHNOLOGIES. **OpenXR Plugin**. Available from: `https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.15/manual/index.html`. Visited on: 2 Sept. 2025.

UNITY TECHNOLOGIES. **ROS-TCP-CONNECTOR**. Available from: `https://github.com/Unity-Technologies/ROS-TCP-Connector`. Visited on: 22 Aug. 2025.

UNITY TECHNOLOGIES. **ROS-TCP-Endpoint**. Available from: `https://github.com/Unity-Technologies/ROS-TCP-Endpoint`. Visited on: 8 Sept. 2025.

UNITY TECHNOLOGIES. **Unity Real-Time Development Platform**. Available from: `https://unity.com/`. Visited on: 22 Aug. 2025.

UNITY TECHNOLOGIES. **URDF Importer**. Available from: `https://github.com/Unity-Technologies/URDF-Importer`. Visited on: 2 Sept. 2025.

UNITY TECHNOLOGIES. **XR Interaction Toolkit**. Available from: `https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html`. Visited on: 2 Sept. 2025.

VAIRAVASAMY, Sasidharan; HAMIN, M. J.; AHMED, Syed; SEENU, N.; DEAN, Ramya M. M. Simulation and Real Time of VR Controlled Robotic Manipulator Using ROS. *In:* 2022 IEEE Bombay Section Signature Conference (IBSSC). Mumbai, India: IEEE, 2022. p. 1–6.

VILLANI, Valeria; PINI, Fabio; LEALI, Francesco; SECCHI, Cristian. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. **Mechatronics**, Elsevier, v. 55, p. 248–266, 2018.

WAMPLER, Charles W. Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 16, n. 1, p. 93–101, 1986.

WANG, Xiangyu. **Mixed Reality and Human–Robot Interaction**. Berlin, Heidelberg: Springer, 2011. v. 47. (Springer Tracts in Advanced Robotics).

WHITNEY, David; ROSEN, Eric; PHILLIPS, Elizabeth; KONIDARIS, George; TELLEX, Stefanie. Comparing Robot Grasping Teleoperation Across Desktop and Virtual Reality with ROS Reality. *In:* AMATO, Nancy M.; HAGER, Gregory D.; THOMAS, Seth; TORRES-TORRITI, Miguel (eds.). **Robotics Research**. Cham, Switzerland: Springer, 2020. v. 10. (Springer Proceedings in Advanced Robotics), p. 335–350.

WHITNEY, David; ROSEN, Eric; ULLMAN, Daniel; PHILLIPS, Elizabeth; TELLEX, Stefanie. ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots. *In:* 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid, Spain: IEEE, 2018. p. 1–9.

WHITTON, Mary C. Making virtual environments compelling. **Communications of the ACM**, ACM New York, NY, USA, v. 46, n. 7, p. 40–47, 2003.

WONSICK, Murphy; PADIR, Taskin. A systematic review of virtual reality interfaces for controlling and interacting with robots. **Applied Sciences**, MDPI, v. 10, n. 24, p. 9051, 2020.

WU, Yan; ZHAO, Bin; LI, Qi. The Teleoperation of Robot Arms by Interacting with an Object's Digital Twin in a Mixed Reality Environment. **Applied Sciences (2076-3417)**, v. 15, n. 7, 2025.

XIE, Yiping; HOU, Xilong; WANG, Shuangyi. Design of a novel haptic joystick for the teleoperation of continuum-mechanism-based medical robots. **Robotics**, MDPI, v. 12, n. 2, p. 52, 2023.

YIM, Lai Sum; VO, Quang T. N.; HUANG, Ching-I; WANG, Chi-Ruei; MCQUEARY, Wren; WANG, Hsueh-Cheng; HUANG, Haikun; YU, Lap-Fai. WFH-VR: Teleoperating a Robot Arm to Set a Dining Table Across the Globe via Virtual Reality. *In:* 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Kyoto, Japan: IEEE, 2022. p. 4927–4934.

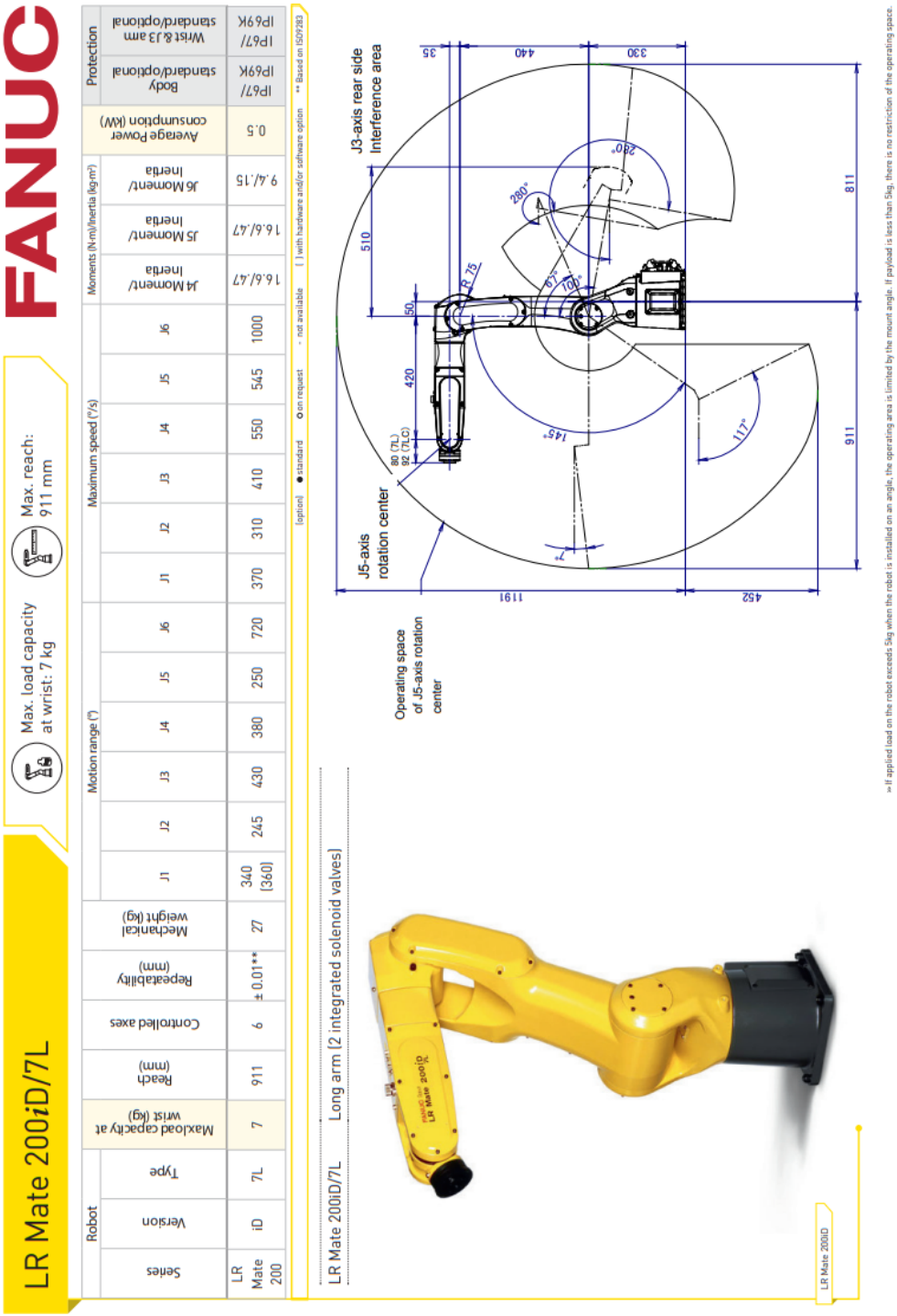
YOUSSEF, Karim; SAID, Sherif; AL KORK, Samer; BEYROUTHY, Taha. Telepresence in the recent literature with a focus on robotic platforms, applications and challenges. **Robotics**, MDPI, v. 12, n. 4, p. 111, 2023.

ZANATTA, Francesco; GIARDINI, Anna; PIEROBON, Antonia; D'ADDARIO, Marco; STECA, Patrizia. A systematic review on the usability of robotic and virtual reality devices in neuromotor rehabilitation: patients' and healthcare professionals' perspective. **BMC health services research**, Springer, v. 22, n. 1, p. 523, 2022.

ZHANG, Qinglei; LIU, Qinghao; DUAN, Jianguo; QIN, Jiyun. Research on teleoperated virtual reality human–robot five-dimensional collaboration system. **Biomimetics**, MDPI, v. 8, n. 8, p. 605, 2023.

APPENDIX A – TECHNICAL SPECIFICATIONS OF THE FANUC LR MATE 200ID/7L

Figure A.1 – Technical Specifications of the FANUC LR Mate 200iD/7L



Source: (FANUC, 2025d).

APPENDIX B – TECHNICAL SPECIFICATIONS OF THE MYGRIPPER F100

Figure B.1 – Technical Specifications of the MyGripper F100

Name	myGripper F100
Material	PC, PBT
Manufacturing Process	Injection molding
Stroke (Diameter)	0-100 mm (default fingertip spacing)
Rated Payload	500 g, multi-level adjustable
Repeated Positioning Accuracy	±0.5mm
Lifespan	300,000 opening/closing cycles
Drive Type	Electric
Transmission Method	Gear + linkage
Dimensions	156×106×61 mm
Weight	340 g
Operating Voltage	24V
Interface Model	M8-8PIN
Mounting Method	Screw-mounted
Operating Environment Requirements	Normal temperature and pressure
Control Interface	5x5 LED matrix
Communication Frame Rate	MODBUS-RTU based on RS485 bus, IO control, Button control
Robot Compatibility	**myCobot 320 Series** **myCobot Pro 630** **Mercury A/B/X Series** **Other general-purpose robots**

Source: (Elephant Robotics, 2025a).

APPENDIX C – INTEGRATION BETWEEN SYSTEMS

This appendix presents the complete technical detailing of the integrations between the teleoperation system components, complementing the summarized description presented in Section 4.1.1. The configuration procedures, installation steps, parameter settings, package versions, as well as the stages required to establish communication between the HMD, Unity, ROS, and the FANUC robot controller are described.

HMD – UNITY

Communication between the Meta Quest 3 HMD and Unity is established through the Meta Quest Link software, connected to the computer via a USB 3.0 cable, ensuring high data transfer rates and low latency. This connection allows the tracking data from the HMD and controllers to be continuously received by Unity, maintaining real-time updates of the operator's position and orientation within the virtual environment.

Additionally, Unity processes and sends graphical information back to the HMD, ensuring that the user visualizes the virtual environment in a synchronized manner. Thus, the connection acts as a bidirectional channel between the device and the graphics engine, enabling continuous and immersive interaction within the application.

UNITY – ROS

The integration between Unity and ROS was enabled through the ROS–TCP Connector package (Unity Technologies, 2025c), which establishes bidirectional communication between the two systems using the TCP/IP protocol. In this architecture, Unity acts as a client, sending and receiving messages through sockets, while ROS operates as a server, responsible for interpreting and publishing the corresponding topics. For message exchange to occur correctly, both computers, the one running Unity and the one running ROS, must be connected to the same local network, with IP addresses and ports configured consistently.

In Unity, the ROS–TCP Connector was installed through the Unity Package Manager. Version 0.7.0 was used, due to its compatibility with the Unity version adopted in this work. After installation, the ROS messages required by the application were configured for publication and subscription, allowing communication between Unity and ROS according to the requirements of the teleoperation system.

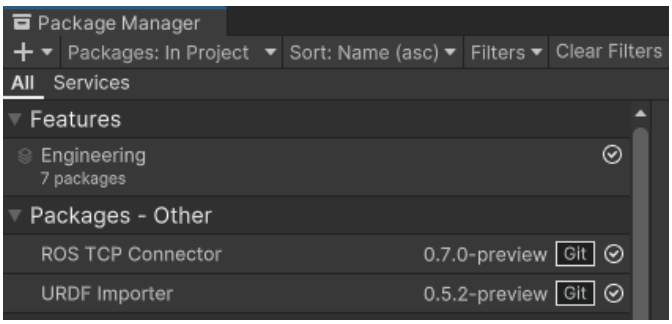
On the ROS side, the complementary ROS–TCP Endpoint package (Unity Technologies, 2025d) was used, responsible for providing the nodes that establish the connection with Unity. This component translates ROS topics, services, and actions into formats understandable by the graphics engine, allowing the operator's mixed reality movements to be converted into robotic commands and, simultaneously, enabling data

from the physical robot, such as joint states or sensor information, to be returned to Unity in real time.

Another fundamental resource used was the URDF Importer (Unity Technologies, 2025f), also installed through the Unity Package Manager. The adopted version was 0.5.2, which allows directly importing into Unity the robot model described in URDF format, the ROS standard for representing the kinematic and geometric structure of robots. The importer automatically generates the robot's three-dimensional representation, preserving the hierarchy of links, joints, and defined degrees of freedom. In this way, it ensures that the Unity simulation corresponds to the mechanical structure of the physical robot, guaranteeing visual coherence and synchronization with the real system.

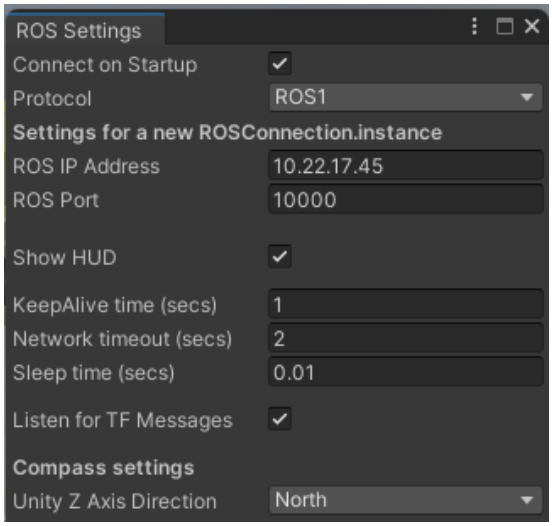
Figure C.1 presents the ROS–TCP Connector and URDF Importer packages installed in the Unity environment, and Figure C.2 shows the ROS–TCP Connector package settings used for the ROS connection.

Figure C.1 – Packages installed in Unity



Source: Author (2026).

Figure C.2 – ROS–TCP Connector configuration



Source: Author (2026).

Complementing this architecture, MoveIt was employed in the ROS environment to generate feasible trajectories while considering kinematic constraints, joint limits, and collision avoidance, with the resulting robot states being published through ROS topics and transmitted to Unity, enabling continuous updates of the virtual robot models according to the current planning and execution states. This mechanism provided visual feedback to the operator, maintaining consistency between the virtual representation and the physical robot throughout the teleoperation process.

Within this architecture, Unity was responsible for immersive visualization and user interaction, while ROS handled robot control, trajectory planning, and communication with the physical system, with the integration between both environments being established through the ROS–TCP Connector, ROS–TCP Endpoint, and the URDF Importer.

ROS – FANUC

For the connection between ROS and the FANUC robot, it was necessary to configure both environments separately: the ROS environment, responsible for managing communication, and the robot controller environment, responsible for executing the received commands.

ROS Setup

To configure the ROS environment in order to enable the connection with the FANUC robot controller, the official ROS-Industrial packages and repositories available on Github (ROS-INDUSTRIAL, 2025a) were used. This repository covers a wide range

of FANUC robot models and their respective configurations, including the FANUC LR Mate 200iD/7L robot used in this developed application.

Among all packages in this repository, three of them played a prominent role in the application development:

- *fanuc_lrmate200id_support*
- *fanuc_lrmate200id7l_moveit_config*
- *fanuc_lrmate200id_moveit_plugins*

The *fanuc_lrmate200id_support* package gathers the fundamental files for robot description, including the definition of joint names, the URDF model with its configurations (such as positions and movement limits), and the three-dimensional models in STL format, used for visualization and collision analysis. In addition, it contains the kinematic descriptions and geometric parameters required for ROS to correctly recognize the manipulator structure, serving as the basis for the remaining configuration packages.

Complementarily, the *fanuc_lrmate200id7l_moveit_config* package provides the complete MoveIt environment configuration for the LR Mate 200iD/7L model. This package integrates the definitions of joints, links, and operating limits, in addition to the configurations that enable the initialization of the planning system and its integration with RViz. In this way, both simulation in a virtual environment and trajectory execution on the physical robot become possible.

Finally, the *fanuc_lrmate200id_moveit_plugins* package exclusively concentrates the implementation of the IKFast *plugin*, responsible for the analytical calculation of the manipulator inverse kinematics.

Together, these three packages constitute the basis for controlling the FANUC LR Mate 200iD/7L robot in ROS, covering everything from geometric modeling to practical integration with the industrial controller.

FANUC Setup

To configure the FANUC controller in order to allow connection with ROS-Industrial, the official ROS tutorials (ROS Wiki, 2025d) were used.

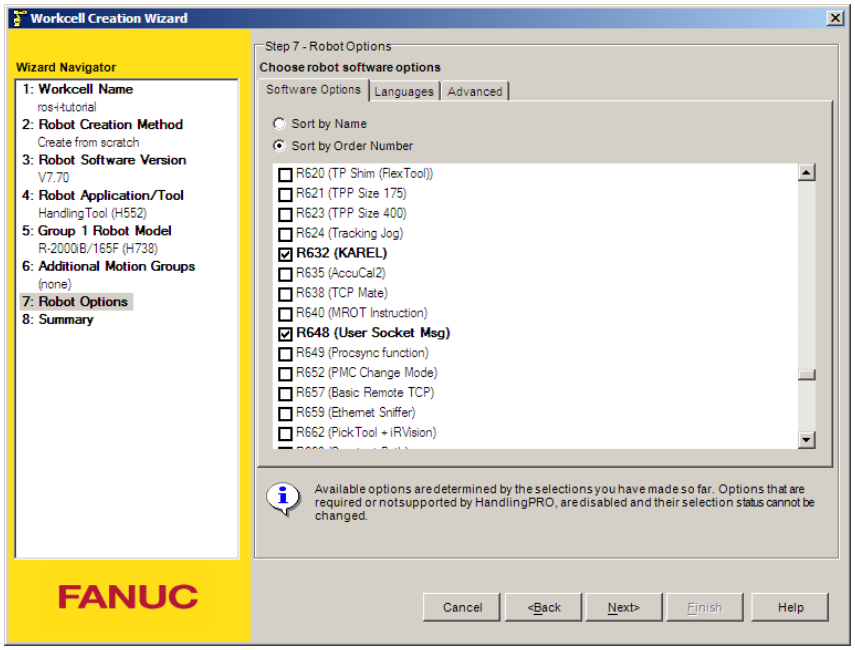
Two installation approaches are available: the *binary install*, which uses precompiled binary files and makes the process faster and simpler, and the *source installation*, which consists of compiling files from source code using the Roboguide software. In the present work, source installation was chosen due to the incompatibility between the ROS version used (3.2.3) and the precompiled binaries.

For controller installation and configuration, some prerequisites must be satisfied. First, the controller must include certain additional software options, which are not included with the robot by default and must be purchased or activated separately,

such as options R632 (KAREL) and R648 (User Socket Messaging). In addition, it is necessary to have a copy of the *fanuc_driver* package (ROS-INDUSTRIAL, 2025b), version 0.2 or higher, as well as the Fanuc Roboguide software installed on a Windows computer or virtual machine, since Roboguide is not compatible with Linux systems.

With the prerequisites satisfied, the first installation stage consisted of creating a new *workcell* in Roboguide, using the software wizard and selecting the correct robot type and required software options (Figure C.3). This workcell served as the basis for file compilation before transfer to the controller.

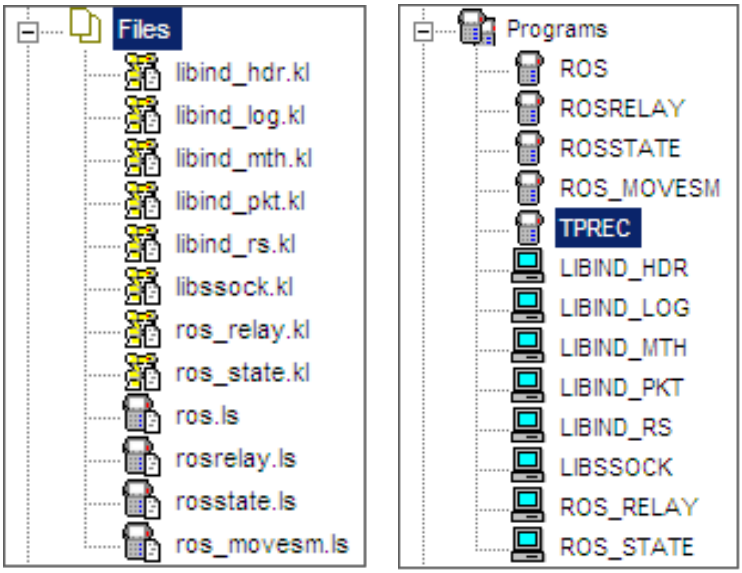
Figure C.3 – Workcell configuration in Roboguide



Source: (ROS Wiki, 2025f).

Next, the KAREL and TPE source files from the *fanuc_driver* package were imported into the workcell (Figure C.4a) and then compiled in Roboguide (Figure C.4b), thus generating the binary files required for transfer and execution on the robot controller.

Figure C.4 – Files imported and compiled in Roboguide

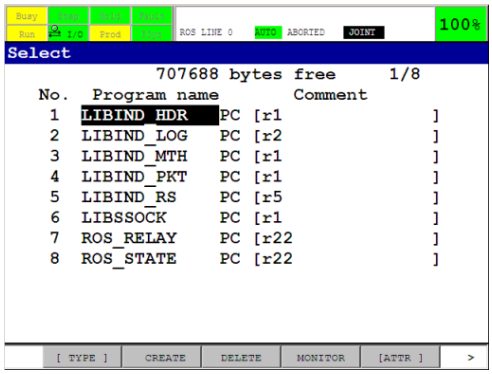


(a) Imported files. (b) Compiled files.

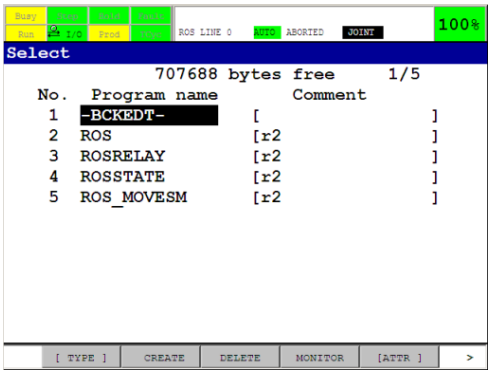
Source: (ROS Wiki, 2025f).

Finally, the compiled binaries were transferred to the controller using an external storage device (USB flash drive). Installation validation was performed directly on the *Teach Pendant*, through the *Program Select* window, where it was possible to confirm the presence and availability of the KAREL programs (Figure C.5a) and TPE programs (Figure C.5b). These programs constitute the interface between the controller and ROS, ensuring that communication is active and functional.

Figure C.5 – List of installed programs on the controller verified in the *Teach Pendant*



(a) KAREL programs present in the controller.



(b) TPE programs present in the controller.

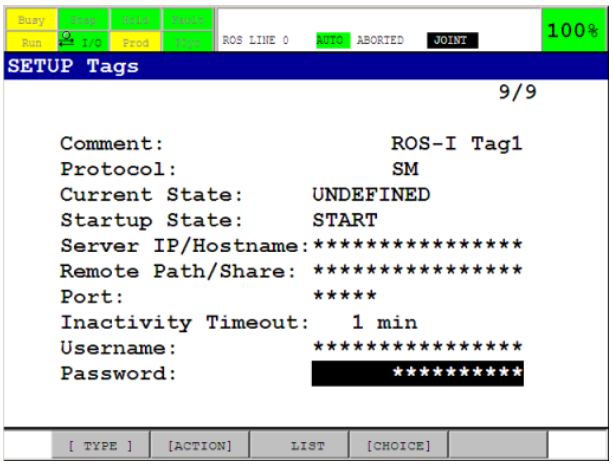
Source: (ROS Wiki, 2025f).

After completing the installation process (ROS Wiki, 2025f), the ROS-Industrial *driver* configuration process (ROS Wiki, 2025e) was initiated on the robot controller. This stage aims to ensure that the installed KAREL and TPE programs are properly

parameterized, allowing correct communication between the controller and the ROS environment.

The configuration involves adjustments at different controller levels. Initially, it is necessary to define the *Server Tags*, which are used by the *ros_state* and *ros_relay* programs to establish connections through *User Socket Messaging*. Figure C.6 presents an example of a *Server Tag* configuration in the controller.

Figure C.6 – Example of a *Server Tag* configuration in the controller



Source: (ROS Wiki, 2025e).

In the context of FANUC controllers, *Server Tags* function as identifiers associated with the *User Socket Messaging* feature, creating network communication channels between the robot and external applications such as ROS. Integer registers act as global numerical variables, useful for movement parameterization or information exchange, while positional registers directly store manipulator spatial points, either in joint or Cartesian coordinates. The *flags*, in turn, have a boolean nature (enabled/disabled) and are used for signaling and synchronization between programs. Together, these elements constitute the basis for ROS-Industrial to interact in a structured manner with the controller, ensuring state transmission and trajectory execution.

Another fundamental aspect consists of adjusting the maximum number of concurrent tasks, since the controller, by default, has a reduced limit of simultaneous processes, which could prevent the initialization of ROS-Industrial programs.

Next, the specific configurations of the KAREL programs are performed. Figures C.7 and C.8 show, respectively, the parameterization of the *ros_relay* and *ros_state* programs, where communication parameters such as TCP ports, *server tags*, speed, and update rate are defined.

Figure C.7 – Default configuration of the *ros_relay* program

Name	Type	Default	Unit	Description
checked	BOOLEAN	False	-	Configuration has been completed by user
f_msm_rdy	INTEGER	1	-	movesm i'face: 'ready/ack' signal flag
f_msm_drdy	INTEGER	2	-	movesm i'face: 'data ready' signal flag
loop_hz	INTEGER	42	Hz	Main loop update rate
move_cnt	INTEGER	50	%	CNT to set with each joint motion instruction
move_speed	INTEGER	20	%	Joint speed to set for all trajectory points
pr_move	INTEGER	1	-	movesm i'face: position register for next trajectory point
r_move_spd	INTEGER	1	-	movesm i'face: integer register for motion speed
r_move_cnt	INTEGER	2	-	movesm i'face: integer register for CNT value
s_tcp_nr	INTEGER	11000	-	TCP port to listen on
s_tag_nr	INTEGER	4	-	Index of the Server Tag to use
um_clear	BOOLEAN	True	-	Clear user menu on start

Source: (ROS Wiki, 2025e).

Figure C.8 – Default configuration of the *ros_state* program

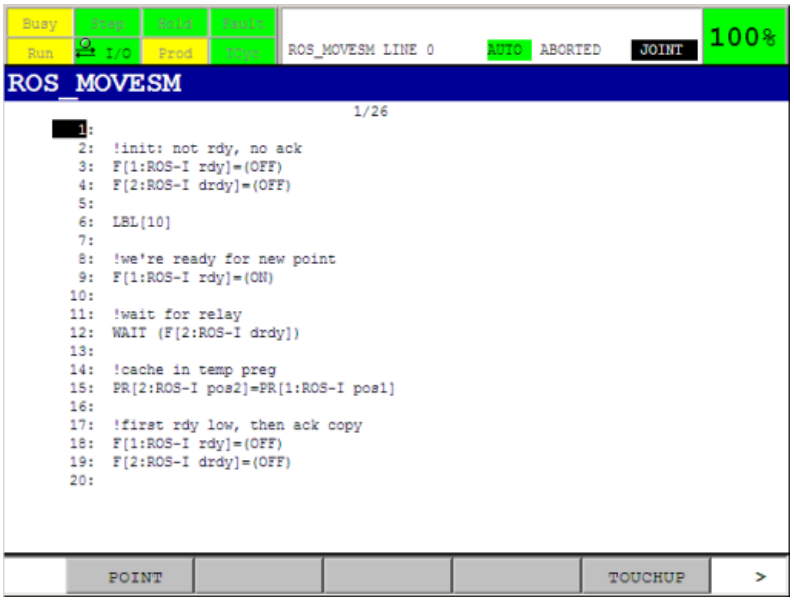
Name	Type	Default	Unit	Description
checked	BOOLEAN	False	-	Configuration has been completed by user
loop_hz	INTEGER	42	Hz	Main loop update rate
sloop_div	INTEGER	10	-	Divider for robot_status reporter loop
s_tcp_nr	INTEGER	11002	-	TCP port to listen on
s_tag_nr	INTEGER	3	-	Index of the Server Tag to use
um_clear	BOOLEAN	True	-	Clear user menu on start

Source: (ROS Wiki, 2025e).

After completing the configurations of the *ros_relay* and *ros_state* programs, it is necessary to change the state of the checked line from *false* to *true*.

Finally, the TPE program *ros_movesm* must be manually edited to align the register and *flag* references with the previously defined values. Figure C.9 presents the configured program in the *Teach Pendant*.

Figure C.9 – Configured *ros_movesm* program in the *Teach Pendant*



Source: (ROS Wiki, 2025e).

With the configuration completed, the controller became capable of operating together with ROS. The detailed robot initialization procedure, including the necessary steps to place it into functional mode and establish communication with ROS, is described in Appendix D.

APPENDIX D – FANUC ROBOT INITIALIZATION PROCEDURE FOR ROS CONNECTION

This appendix presents the step-by-step procedure required to power on the FANUC LR Mate 200iD/7L robot and place it in operational condition, ready for connection with ROS.

STEP-BY-STEP PROCEDURE

1. **Powering on the robot and initial commands**
    - Verify that the robot is connected to the main power supply.
    - Turn on the control panel and wait for the system initialization.
    - Check whether the emergency buttons are pressed; if so, rotate them to unlock.
    - Select the *Auto* mode using the selector switch on both the controller and the Teach Pendant (TP).
    - To reset alarms: while pressing the *Deadman switch*, press “Shift” and then “Reset”.
2. **Robot configuration for ROS operation**
    - On the TP, press “Menu” → “6 SETUP” → “0 - - NEXT - -” → “9 Host Comm” and press “Enter” (Figure D.1).

Figure D.1 – Teach Pendant screen with access to the Host Comm menu



Source: Author (2026).

- On the TP, press "| SHOW |" → "3 Servers" (Figure D.2).

Figure D.2 – Server configuration on the Teach Pendant



Source: Author (2026).

- Select tag "S5", press "[ACTION]" → "1 DEFINE" → "Enter" (Figure D.3a). After that, press "[ACTION]" → "3 START" → "Enter" (Figure D.3b). The TP screen should indicate that the "State" of tag "S5" is "[STARTED]". Return to the initial screen using the button indicated in Figure D.4.

Figure D.3 – Definition and startup of tag "S5" on the Teach Pendant



(a) Tag S5-Define.



(b) Tag S5-Start.

Source: Author (2026).

Figure D.4 – Verification of tag “S5” on the Teach Pendant



Source: Author (2026).

- On the initial screen, press “SELECT”, select the “ROS” program, and press “ENTER” (Figure D.5a). The screen should display the corresponding information (Figure D.5b).

Figure D.5 – Selection and execution of the “ROS” program on the Teach Pendant



(a) Selecting the “ROS” program.



(b) “ROS” program running.

Source: Author (2026).

- To finalize the configuration, press the green button on the controller; the indicator light should remain on (Figure D.6).

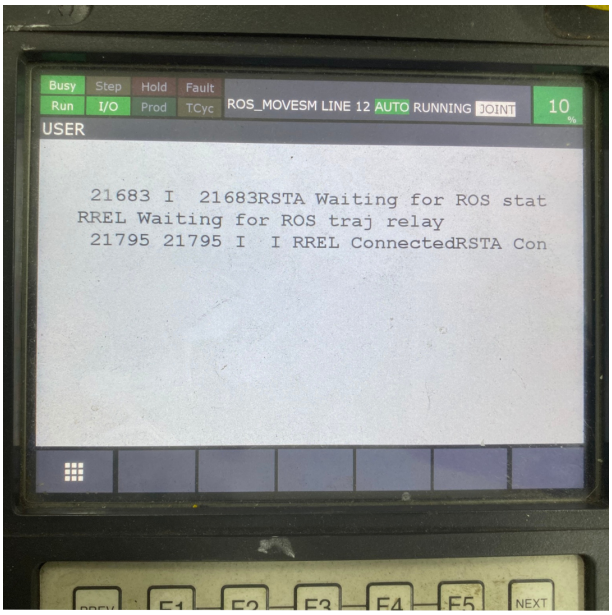
Figure D.6 – Green controller button indicating active operation



Source: Author (2026).

- Confirm that the TP screen shows the “ROS” program running (Figure D.7), indicating that the robot is ready for ROS connection.

Figure D.7 – Screen indicating ROS program operation



Source: Author (2026).

3. **ROS communication**
   - Connect the robot controller network cable to the local network used by ROS.
   - Test the connection with ROS (Appendix E).

APPENDIX E – ROS ENVIRONMENT INITIALIZATION PROCEDURE

After completing the FANUC robot configuration procedures (Appendix D), the slave system configuration in the ROS environment begins. This procedure is divided into two operation modes: conventional mode and immersive mode. Although both use the same infrastructure, each mode requires a distinct initialization sequence, detailed in the following sections.

IMMERSIVE MODE (UNITY + ROS + FANUC)

In immersive mode, 11 terminals are required, each responsible for a specific part of the communication between Unity, ROS, FANUC, gripper, and sensors. The execution order presented below must be strictly followed, as the system was structured to operate in this manner.

Terminal 1: Unity–ROS Connection

This terminal starts the *ROS TCP Endpoint*, responsible for establishing the communication bridge between Unity and ROS. Without this node, Unity cannot publish or receive messages.

```
roslaunch ros_tcp_endpoint endpoint.launch
```

Terminal 2: Gripper Control (Unity $\rightarrow$ ROS)

Launches the `.launch` file that enables the integration between the virtual gripper control in Unity and ROS. This node sends opening and closing commands to MoveIt and to the hardware.

```
roslaunch pro_gripper_f100 force_gripper_slider_unity.launch
```

Terminal 3: Connection with the Physical Gripper Hardware

This terminal establishes direct communication with the physical gripper microcontroller. The port may be `/dev/ttyACM0` or `/dev/ttyACM1`, depending on the order in which the devices were connected. It is recommended to verify the available entries beforehand.

```
rosrun pro_gripper_f100 force_gripper_slider.py _port:=/dev/ttyACM0 _baud:=115200
```

Terminal 4: Connection with the FANUC Robot

This command initializes MoveIt and connects ROS to the physical controller of the FANUC LR Mate 200iD/7L. The parameter `robot_ip:=10.0.0.40` corresponds to the IP address configured in the robot controller.

```
roslaunch fanuc_lrmate200id7l_moveit_config moveit_planning_execution.launch sim:=
    false robot_ip:=10.0.0.40
```

Terminal 5: HMD Controller Bridge

Node responsible for mapping the HMD (Meta Quest 3) controller buttons to actions executed in MoveIt, such as trajectory confirmation, automatic execution, and fine control adjustments.

```
rosrun button_p_e fanuc_moveit_bridge.py
```

Terminal 6: Unity $\rightarrow$ ROS Gizmo Synchronization

This node continuously updates the *interactive marker* in RViz according to the gizmo position manipulated in Unity. It is essential for keeping the target pose synchronized between both environments.

```
rosrun unity_moveit_bridge unity_interactive_marker_node.py
```

Terminal 7: Automatic Mode Activation

Script responsible for enabling the automatic execution mode in MoveIt. It must be executed inside the correct package folder, as it uses relative references.

```
cd ~/catkin_ws/src/button_p_e/scripts
python3 fanuc_moveit_go.py
```

Terminals 8 and 9: Camera Initialization

The cameras responsible for capturing the physical environment must each be started in their own terminal.

First, verify the available devices:

```
ls /dev/video*
```

Camera 1 (e.g., /dev/video3 on port 8080):

```
cd ~/mjpg-streamer/mjpg-streamer-experimental
v4l2-ctl -d /dev/video3 --set-ctrl=power_line_frequency=2 &&  ./mjpg_streamer -i "
    input_uvc.so -d /dev/video3 -r 1280x720 -f 30" -o "output_http.so -w ./www -p 8080
    "
```

Camera 2 (e.g., /dev/video1 on port 8081):

```
cd ~/mjpg-streamer/mjpg-streamer-experimental
v4l2-ctl -d /dev/video1 --set-ctrl=power_line_frequency=2 && ./mjpg_streamer -i "
    input_uvc.so -d /dev/video1 -r 1920x1080 -f 30" -o "output_http.so -w ./www -p
    8081"
```

Terminal 10: Arduino Initialization (TOF Sensor)

Here, only the Arduino IDE is opened, required to upload the sensor firmware when necessary.

```
~/Applications/ArduinoIDE/arduino-ide.AppImage
```

Terminal 11: TOF Sensor ROS Publishing

This node uses `rosserial` to publish to ROS the values read by the TOF sensor connected via USB.

```
rosrun rosserial_python serial_node.py _port:=/dev/ttyACM1
```

Terminal 12: TOF Sensor Marker in RViz

Publishes the visual marker of the sensor in RViz to assist in interpreting the measured distance.

```
roslaunch my_visualizers tof_tool0_marker.launch
```

With all 11 terminals running without errors, the immersive-mode teleoperation system will be fully operational.

CONVENTIONAL MODE (RVIZ + MOVEIT)

In conventional mode, the operator uses RViz and MoveIt directly, without interaction with Unity. Four terminals are required, as described below.

Terminal 1: Gripper Control

```
roslaunch pro_gripper_f100 force_gripper_slider_alone.launch
```

Terminal 2: Connection with the Physical Gripper

```
rosrun pro_gripper_f100 force_gripper_slider.py _port:=/dev/ttyACM0 _baud:=115200
```

Terminal 3: Connection with the FANUC Robot

```
roslaunch fanuc_lrmate200id7l_moveit_config moveit_planning_execution.launch sim:=
    false robot_ip:=10.0.0.40
```

Terminal 4: Interactive Marker in RViz

```
cd ~/catkin_ws/src/unity_moveit_integration/scripts
rosrun unity_moveit_bridge unity_interactive_marker_node.py
```

The procedures related to the cameras and the TOF sensor are identical to those used in immersive mode. In RViz, to display the sensor marker, select: *Add* $\rightarrow$ *By topic* $\rightarrow$ *Marker*.

With all conventional mode terminals active, the system will be ready for operation using RViz and MoveIt.

APPENDIX F – UNITY ENVIRONMENT INITIALIZATION PROCEDURE

This appendix presents the procedure required to start the Unity environment, establish the connection with the Meta Quest headset, and integrate the immersive teleoperation system with the master computer and the ROS environment. The following steps must be performed after the complete ROS initialization, as described in Appendix E.

1. PHYSICAL CONNECTION AND META QUEST LINK ACTIVATION

The first step consists of connecting the Meta Quest headset to the master computer using a USB 3.0 cable. Communication via *Quest Link* is required to stream the Unity environment to the HMD (Meta, 2025a).

1. Connect a USB 3.0 cable to a USB 3.0 port on the computer.
2. Connect the other end to the Meta Quest.
3. Put on the headset and press the *Meta* button to open the universal menu.
4. Access the quick settings menu by selecting the clock in the left corner.
5. Select the *Link* option and choose the connected computer.
6. Confirm by selecting *Launch*.

After this step, the headset will be ready to receive the Unity execution.

2. OPENING THE UNITY PROJECT

1. Open the *Unity Hub*.
2. Locate the **FANUC_MR** project.
3. Wait for the loading of the assets, scripts, and virtual environment hierarchies.

3. ROS CONNECTOR CONFIGURATION IN UNITY

The *ROSConnector* component is responsible for establishing communication between Unity and ROS. To configure it, follow the steps below:

1. On the slave computer (Ubuntu), obtain the IP address by executing the command:

```
hostname -I
```

2. In Unity, select the `ROSConnector` object in the hierarchy.

3. In the *Inspector*, enter the obtained IP address in the *ROS IP Address* field, as previously illustrated in Figure C.2.

4. MONITORING CAMERA CONFIGURATION IN UNITY

Unity receives the video streams from the physical environment cameras through MJPEG-Streamer running on the ROS computer. Therefore, the URL configured in Unity must exactly match the port defined in the command used in ROS.

The URL structure is:

```
http://<ROS-IP>:<port>/?action=snapshot
```

The IP address is the same one defined in the ROSConnector configuration. The port must be taken directly from the command used to start MJPEG-Streamer in ROS. Below is a real example of a command used in the system:

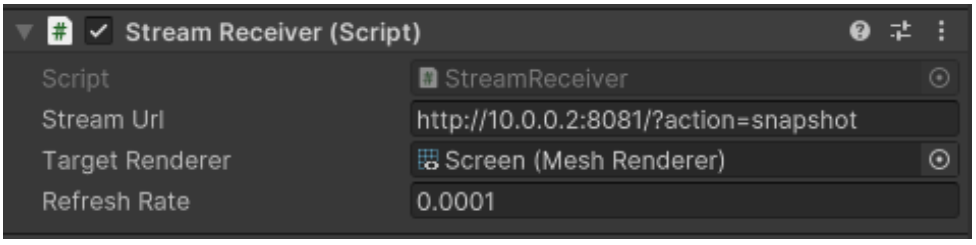
```
cd ~/mjpg-streamer/mjpg-streamer-experimental
v4l2-ctl -d /dev/video2 --set-ctrl=power_line_frequency=2
./mjpg_streamer -i "input_uvc.so -d /dev/video2 -r 1920x1080 -f 30" -o "output_http.so
    -w ./www -p 8080"
```

In this case, the configured port is **8080**. If the ROS computer IP address is **10.0.0.2**, the correct URL to be configured in Unity will be:

```
http://10.0.0.2:8080/?action=snapshot
```

If there is a second camera running on another port (for example, 8081), Unity must use this second port in a second capture instance.

Figure F.1 – Camera configuration in the Unity “Stream Receiver” script



Source: Author (2026).

5. UNITY ENVIRONMENT EXECUTION

After completing the ROSConnector and camera configurations:

1. Click the Play button at the top of the Unity window.
2. The immersive environment will be loaded into the Meta Quest through Link.
3. All functionalities will operate in real time, including:
    - gizmo synchronization with MoveIt,
    - camera visualization,
    - interpretation of HMD buttons,
    - transmission of the target pose to ROS.

The MR teleoperation system will then be fully operational.