Robotic Teleoperation: A Cross-Cultural Study of User Experience and Performance

Florent P. Audonnet

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow

December 2024

Abstract

This thesis explores the performance and user experience of robotic teleoperation systems, focusing on the impact of various factors such as control methods, robot types, and levels of immersion. Robotic teleoperation, which allows human operators to control robots directly by moving some part of their body, is a critical technology in fields ranging from industrial automation to medical robotics. However, the cognitive and physical demands placed on operators, especially non-experts, remain under-explored. This research addresses these gaps by developing a modular, plug-and-play framework for direct teleoperation and conducting extensive user studies across different hardware configurations and cultural contexts.

The core contributions of this thesis include a systematic comparison of simulation software for robotic arm manipulation using ROS2, the development of a novel teleoperation framework called TELESIM, and an investigation into the effects of immersion through a mixed reality system named IMMERTWIN. The systematic comparison, achieved through a benchmark of various ROS2-compatible simulation platforms evaluates their suitability for teleoperation tasks, resource utilization, and their potential to serve as digital twins. This evaluation provides critical insights into the strengths and limitations of current simulation tools in supporting real-time robotic operations. Using the insights gained from our evaluation, TELESIM, a modular and plug-and-play framework, was designed to enable seamless control of different robotic systems with minimal setup time. It leverages digital twin technology to offer real-time feedback and control, allowing operators to interact with the robot to execute tasks in the real world. This framework was tested across multiple robotic platforms (e.g., UR3, Baxter, UR5e), control methods (e.g., VR controllers) and countries (e.g., UK, Japan) offering a flexible solution for diverse teleoperation scenarios. Additionally, IMMERTWIN, a mixed reality system, was developed to explore how varying levels of immersion affect operator performance and cognitive load. By integrating real-world data into virtual environments, IMMERTWIN enhances user interaction with robotic systems, providing a more intuitive and immersive teleoperation experience. The research also delves into external factors such as user expertise and trust in robots, examining how these elements influence teleoperation performance across different cultural contexts.

Key findings demonstrate that robot type, controller configuration, and immersion level significantly affect task performance and operator workload. For instance, the study revealed that different robots such as the Universal Robot 3 or Baxter, exhibit varying levels of precision and ease of control, which directly impacts task completion times and error rates. Quantitative results showed the Red Design (Baxter with VR controllers) achieved a 77.42% cube placement rate compared to only 46.29% for the Blue Design (UR3 with SenseGlove). Task completion rates also varied dramatically, with 85% of participants using VR controllers successfully completing tower-building tasks within the allocated time, compared to only 46% of those using SenseGlove controllers. Immersive systems like IMMERTWIN, which integrate real-world data into virtual environments, were shown to reduce cognitive load by providing more natural interaction cues and better situational awareness. NASA-TLX assessments demonstrated that IMMERTWIN reduced mental demand scores by 23% compared to non-immersive systems. Additionally, the research highlights the importance of external factors such as user expertise and trust in robots. Non-expert users, who participated in a large-scale international study conducted in both the UK and Japan (n=74), reported varying levels of comfort and trust depending on their cultural background and prior experience with robotic systems. The study found that the Japanese participants generally exhibited higher levels of trust toward robots compared to their UK counterparts, with Japanese participants scoring 41.94 on the Negative Attitude Towards Robot Scale compared to 49.43 for UK participants, representing a statistically significant difference of 15%. The Yellow Design (UR5e with VR controllers) produced the lowest NASA-TLX mental demand scores (8.94 out of 21), while the Blue Design resulted in the highest physical demand (11.65) and frustration levels (10.76). Overall, the thesis provides valuable insights into optimizing teleoperation systems for non-expert users by balancing technological complexity with human-centric design principles, ultimately paving the way for more accessible and efficient human-robot collaboration.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Gerardo Aragon-Camarasa, for his invaluable guidance, insightful feedback, and unwavering support throughout my research journey over the past three years. His expertise and knowledge have been instrumental in shaping my research direction, and I am particularly grateful for the academic freedom he granted me, allowing me to explore and pursue my own ideas while ensuring that I remained focused and on track. I am also grateful for his help in teaching me the art of writing high-quality research papers.

I also want to thanks all the members of the Computer Vision and Autonomous Systems group, and especially the member past and present of my G141 office: Dr Ali AlQallaf, Dr Piotr Ozimek, Dr Li Duan, Dr Ozan Bahadir and the not yet graduated: Lipeng Zhuang, Paul McHard, Oliver Summerell and finally my collegue and flatmate Yingdong Ru. They have been my sounding board, providing valuable insights and started discussions that have greatly contributed to my research. Their presence and camaraderie have made going to the office every day a pleasure, creating a stimulating environment that has been essential to my success. Two other members of the CVAS group needs a specific thanks, Dr Nicolas Pugeault and Dr Jonathan Grizou, who served on my annual progress review board. Their challenging questions and insightful feedback have been instrumental in pushing me to think more critically and deeply about my research choices.

I would like to extend my heartfelt thanks to my friends and, in particular, to all the members of the Francophone Society Committee. Their companionship and support have been invaluable in helping me maintain a healthy work-life balance throughout my Ph.D. journey. They have consistently reminded me of the importance of taking breaks and having fun, which has been essential for maintaining my mental well-being and ultimately enabling me to produce better results in my research.

Last but not least, I want to express my deepest gratitude to my family for their unwavering belief in me, even during the most challenging times of my thesis. Their constant support, encouragement, and unconditional love have been the bedrock upon which I have built my success. I am forever grateful for their presence in my life and for the immeasurable role they have played in helping me achieve this milestone.

List of Publications

The following is a list of papers I have written or contributed to during my PhD. Contributed paper are marked with ∗:

- Published Work

  1. Florent P. Audonnet, Andrew Hamilton, and Gerardo Aragon-Camarasa. A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pages 755–762, November 2022. doi: 10.23919/ICCAS55662.2022.10003832. ISSN: 2642-3901
  2. Florent P Audonnet, Jonathan Grizou, Andrew Hamilton, and Gerardo Aragon-Camarasa. TELESIM: A Modular and Plug-and-Play Framework for Robotic Arm Teleoperation using a Digital Twin. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17770–17777, Yokohama, Japan, May 2024. IEEE. ISBN 9798350384574. doi: 10.1109/ICRA57147.2024.10610935. URL https://ieeexplore.ieee.org/document/10610935/

- Work Under Review

  3. Florent P Audonnet, Andrew Hamilton, Yukiyasu Domae, Ixchel G Ramirez-Alpizar, and Gerardo Aragon-Camarasa. Breaking Down the Barriers: Investigating Non-Expert User Experiences in Robotic Teleoperation in UK and Japan. *arXiv preprint arXiv:2410.18727*, page 13, 2024. doi: https://doi.org/10.48550/arXiv.2410.18727. URL https://arxiv.org/abs/2410.18727
     Submitted to Transactions on Robotics
  4. Florent P. Audonnet, Ixchel G. Ramirez-Alpizar, and Gerardo Aragon-Camarasa. IMMERTWIN: A Mixed Reality Framework for Enhanced Robotic Arm Teleoperation, September 2024. URL http://arxiv.org/abs/2409.08964. arXiv:2409.08964
     Submitted to ICRA 2025
  5. ∗ Lipeng Zhuang, Shiyu Fan, Yingdong Ru, Florent Audonnet, Paul Henderson, and Gerardo Aragon-Camarasa. Flat'n'Fold: A Diverse Multi-Modal Dataset for Garment Perception and Manipulation, September 2024. URL http://arxiv.org/abs/2409

.18297. arXiv:2409.18297
Submitted to ICRA 2025

In this alternative format thesis [1], [2], [3] and [4] have been reproduced as Chapter 2, Chapter 3, Chapter 4, and Chapter 5, respectively. These have been included in the thesis with the same format as the rest of the thesis, and the numbering of pages, references, etc., are in sequence with the rest of the thesis. All papers presented in this thesis have been contributed to at least 90% by myself, and the work was carried out during my time as a PhD student at the University of Glasgow.

Chapter 1

Introduction

Teleoperation, a method of controlling robots or machines remotely, enables human operators to manipulate robotic systems from a distance. This capability proves essential for tasks in environments that may be hazardous, inaccessible, or unsuitable for direct human presence, including military applications, space exploration, and hazardous material handling. In this thesis, we examine robotic arm teleoperation through motion mapping, where the robot mirrors the operator's arm movements.

1.1 Motivation

The ANA Avatar XPRIZE challenge (2018-2022) offered a $10 Million prize pool for developing robotic avatar systems capable of real-time human presence transportation to remote locations [9]. This challenge stimulated extensive research into teleoperation systems, where operators' bodily movements control specific robot actions, leading to numerous innovative approaches for robotic avatar control [10, 11].

While teleoperation presents significant challenges for non-expert operators [12, 13], it remains a critical component of Industry 5.0 [14]. Industry 5.0, the fifth industrial revolution, emphasizes ecological and social considerations, promoting human-centred industrial environments [15, 16, 17, 18]. The European Union identifies six fundamental enabling technologies [14] constituting Industry 5.0's technological foundation, as illustrated in Figure 1.1. These integrated technologies establish an industrial paradigm prioritizing human factors and sustainability, advancing beyond Industry 4.0's efficiency-focused approach. Two technologies demonstrate particular relevance to this research:

- Digital Twins and Real-time Simulation enabling system monitoring
- Human-Machine interaction solutions adapting to worker requirements

Digital Twins, as conceptualized by Grieves [19], establish bidirectional communication between physical and virtual systems. This system enhances teleoperation safety by eliminating

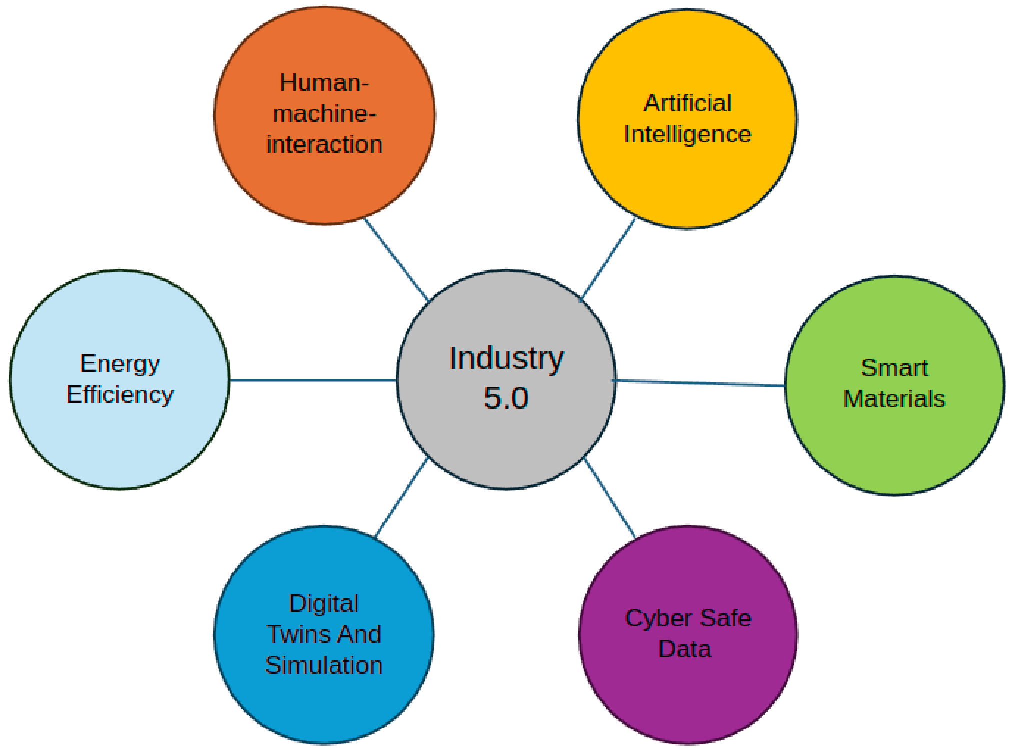


Figure 1.1: Industry 5.0 Enabling Technologies

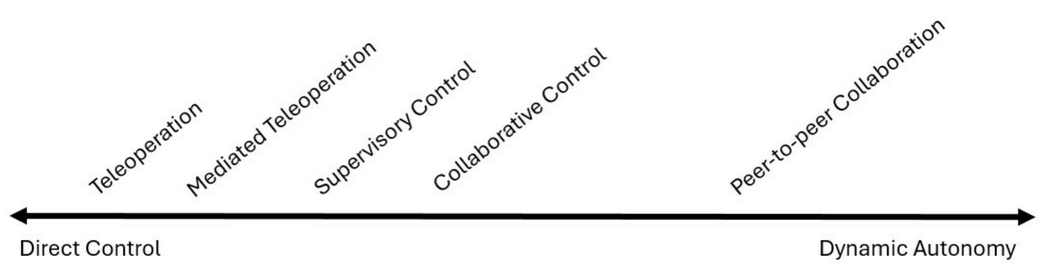


Figure 1.2: Levels of Autonomy, from direct teleoperation (left) to fully autonomous (right). Image reproduced from [1]

the requirement for physical proximity to robots and facilitates operation testing in simulation without risking physical infrastructure [20]. Their ability to mimic real environments makes them a perfect candidate for Industry 5.0.

Teleoperation falls under the umbrella of *Human-Machine Interaction Solutions* as they allow humans to control robots directly. Despite its significance, research in pure teleoperation remains limited, as most investigators focus on developing shared autonomy robots. Shared Autonomy encompasses systems that combine teleoperation with automated actions, defined through levels of autonomy by Goodrich and Schultz [1], as illustrated in Figure 1.2.

While shared autonomy offers potential reductions in operator workload while enhancing performance, the absence of a unified baseline leaves the impact of specific components, such as control methods and robot types, on teleoperation performance undetermined, resulting in researchers employing diverse control methodologies [11, 21, 22].

1.2 Thesis Statement

This PhD thesis investigates the multifaceted implications of direct teleoperation across diverse robotic platforms, control methodologies, and immersion levels, focusing on user performance metrics and the associated user's workload such as mental and physical. While teleoperation has emerged as a promising paradigm for seamless human-robot interaction, the intricate interplay between its fundamental components (control methods, robot typology, and immersion depth) remains largely unexplored. Furthermore, the absence of standardized protocols for implementing direct teleoperation has led to a fragmented research landscape, with individual studies often developing bespoke methodologies.

This PhD thesis seeks to address the critical gap in the literature surrounding direct teleoperation systems and to elucidate the complex interrelationships between human factors and hardware variables that influence task performance and operator workload. To achieve these objectives, this study proposes the development of a novel, modular, and interoperable framework for direct teleoperation. The efficacy and impact of various factors are evaluated through a series of comprehensive user studies, employing non-expert participants to operate an array of robotic manipulators with minimal prior training. By adopting this approach, this thesis aims to contribute significant insights to the field of human-robot interaction and advance our understanding of the optimal design and implementation of teleoperation systems.

1.3 Challenges

1.3.1 Simulation Software

The evolution of robotic simulation software benchmarking began in 2003, with initial methodologies focusing on basic capabilities and resource usage measurements. Early approaches primarily evaluated simulators based on qualitative and quantitative metrics, including OS compatibility, programming language support, and RAM usage during robot control [23]. As ROS [24] emerged as the standard framework for robotics, the focus shifted to comparing simulators' ROS integration capabilities and their effectiveness in creating Digital Twins [25].

Recent research has advanced to more sophisticated benchmarking approaches, emphasizing performance metrics crucial for modern applications such as Digital Twinning and machine learning. Studies have expanded to measure real-time factors, CPU usage, and memory consumption, with particular attention to the impact of GUI rendering on simulation performance [26]. Various researchers have conducted specialized comparisons focusing on specific robotics applications, such as robotic manipulation and humanoid scenarios [27] [28]. Simulation software has become increasingly important as a safe testing environment for operators and a platform for reinforcement learning approaches in autonomous robot training [20].

The research gap becomes apparent in several dimensions. First, while current methodologies have progressed from basic feature comparisons to performance measurements, there remains a lack of standardized, comprehensive benchmarking frameworks to evaluate simulators across diverse robotics applications. Second, with the impending end-of-life of ROS1 in 2025 and the transition to ROS2, there is a critical need for performance analysis of ROS2-compatible simulation software, as compatibility was limited at the start of this research. Finally, there is a pressing need to evaluate simulation software's capability to function as digital twins, meeting Industry 5.0 requirements for real-time systems monitoring and supporting teleoperation baselines for autonomous robot task learning.

1.3.2 Teleoperation

As previously noted in Section 1.1, direct teleoperation systems lack standardized performance metrics to evaluate operator strain [29, 30], particularly during precise millimetre-scale adjustments. While medical applications employ scaled-down user movements [31, 32], the absence of comprehensive benchmarks makes it difficult to assess whether this approach is optimal for tasks requiring extensive arm movements.

The limitations of autonomous systems are well-documented. Most autonomous systems are either hardcoded or learning-based. Hardcoded systems fail in unexpected conditions, while learning-based approaches lack reliability [33, 34]. Hardcoded systems, such as industrial assembly lines, demonstrate rigid computational architectures with minimal adaptive capacity, rendering them susceptible to failure when confronted with unexpected environmental perturbations. Learning-based approaches, on the other hand, have their performance evaluated based on specific datasets with specific conditions, and their performance outside this scope remains an active area of research [35, 36, 37].

As outlined in Figure 1.2, shared autonomy balances human oversight with automated precision [22, 38]. However, the relationship between operator expertise and system effectiveness remains inadequately quantified, particularly in industrial applications where skilled operators manage broad movements while autonomous systems handle precise adjustments [17].

While shared autonomy systems rely fundamentally on direct teleoperation [39], current literature does not systematically evaluate how different control interfaces affect performance. Researchers have explored various control methods such as joysticks [40], keyboards [41], touchscreens [42], and VR controllers [43, 11], but without comparatively analysing their relative influence on operator cognitive load and performance across different robot morphologies.

The emergence of Virtual Reality in teleoperation [44] further complicates performance evaluation. Mixed Reality introduces new variables through digital overlays and real-world data integration [45]. While point cloud visualization shows promise in reducing cognitive strain [46], the field lacks comprehensive metrics to compare performance across different immersion levels and visualization methods.

Cultural factors represent another critical gap, with only Bartneck et al. [47] examining trust variations across countries. This limited understanding of cultural influences on operator trust and interaction patterns hinders the global deployment of teleoperation systems. The absence of standardized metrics further complicates cross-cultural comparisons of operator performance and system acceptance.

1.4 Research Questions

This thesis will attempt to answer the following scientific questions:

Chapter 2

RQ_1 Are current simulation software for robotics suitable for acting as a digital twin or digital clone?

Chapter 3 and 4

RQ_2How do variations in User Interface Device (UID), robotic arm configurations, and gripper designs influence user performance and cognitive workload during teleoperation tasks?

Chapter 4

RQ_3 To what extent does prior user experience with robotics and virtual reality affect task performance and adaptability in teleoperation scenarios?

RQ_4 How do cultural differences between operators from different regions (e.g., UK and Japan) impact workload and performance in robotic teleoperation?

RQ_5 What is the relationship between operator trust in robots and their performance during teleoperation tasks?

Chapter 5

RQ_6 Does Virtual Reality Immersion help reduce the mental and cognitive load of the user during teleoperation, particularly across different cultural contexts?

1.5 Research approach

This thesis begins by investigating whether current simulation software can serve as effective digital twins (**RQ$_1$**). This question is addressed through a systematic comparison of ROS2-compatible simulation platforms in Chapter 2. The benchmarking effort evaluates their suitability for manipulations tasks and resource utilization. Results revealed that while Webots demonstrated high stability and repeatability, Isaac Sim emerged as the most versatile platform due to its integration with machine learning frameworks and advanced motion planning capabilities.

Building on this foundation, Chapter 3 introduces TELESIM, a modular plug-and-play framework designed to facilitate seamless teleoperation across multiple robotic platforms. By integrating digital twin technology, TELESIM provides near real-time feedback and control, enabling operators to interact intuitively with robotic systems. The framework's adaptability is validated through user studies involving different robots (e.g., UR3 and Baxter) and control modalities (e.g., VR controllers and SenseGlove). These studies address **RQ$_2$**, which explores how variations in User Interface Devices (UIDs), robot configurations, and gripper designs influence user performance and cognitive workload.

To expand the scope of evaluation, Chapter 4 examines the impact of cultural differences on teleoperation performance (**RQ$_4$**) and investigates the relationship between prior user experience (**RQ$_3$**) and task adaptability. A large-scale international study involving participants from the UK and Japan reveals significant insights into how cultural factors shape trust in robots (**RQ$_5$**) and user interaction patterns. Contrary to existing literature, Japanese participants exhibited higher trust levels toward robots compared to their UK counterparts. These findings challenge preconceived notions about cultural attitudes toward robotics and highlight the importance of context-specific evaluations.

Chapter 5 extends TELESIM by incorporating immersive virtual reality capabilities through a system named IMMERTWIN. This mixed-reality framework addresses **RQ$_6$**, which investigates whether virtual reality immersion can reduce cognitive load during teleoperation tasks. By integrating real-world data into virtual environments, IMMERTWIN enhances user interaction with robotic systems, providing a more intuitive and immersive teleoperation experience. Experimental results demonstrate that varying levels of immersion significantly affect operator performance and cognitive workload.

The structure of the thesis reflects a logical progression from foundational research to advanced applications. Chapter 1 introduces the motivation, challenges, and research gaps that underpin this work. Chapter 2 establishes a baseline for simulation software evaluation, while Chapter 3 develops and validates the TELESIM framework through user studies. Chapter 4 expands this validation to include cross-cultural comparisons, providing a broader understanding of user interactions with teleoperation systems. Chapter 5 integrates immersive technologies into TELESIM, offering novel insights into enhancing operator experience through mixed reality. Finally, Chapter 6 synthesizes these findings into a comprehensive conclusion that outlines future research directions.

1.6 Contributions

This thesis aims to address the limitations described in Section 1.5 and answers the research questions defined in Section 1.4 by creating a modular framework for direct teleoperation that can be used to establish a baseline by recording the performance of an operator while performing a direct teleoperation task and measuring the cognitive and physical load perceived by the user during this task.

The main contributions of this thesis are:

- Development of a methodological framework for systematic evaluation and comparative analysis of robotic simulation platforms, with specific implementation in ROS2-based environments (Chapter 2).
- Development of baseline performance metrics for direct teleoperation through systematic analysis of user interaction across multiple robotic platforms and control interfaces (Chapter 3).
- Quantitative investigation of teleoperation performance factors through a large-scale international study ($n = 73$), examining the influence of robot types, end-effector design, and control methodologies on operator performance (Chapter 4).
- Analysis of correlations between operator experience, robot trust metrics, and teleoperation performance outcomes (Chapter 4).
- Integration of immersive control capabilities within the modular framework, enabling enhanced operator interaction within the digital twin environment (Chapter 5).
- Experimental evaluation of immersion levels and their impact on teleoperation performance metrics and operator's workload (Chapter 5).

And the following minor contributions:

- Development of a modular, plug-and-play teleoperation framework using digital twin technology for robotic manipulator control, emphasizing system adaptability (Chapter 3).

Chapter 2

A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2

This section (Section 2 reproduces the author's version of the accepted, peer review manuscript presented at the 22nd International Conference on Control, Automation and Systems (ICCAS) as Florent P. Audonnet, Andrew Hamilton, and Gerardo Aragon-Camarasa. A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2. In 2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pages 755–762, November 2022. doi: 10.23919/ICCAS55662.2022.10003832. ISSN: 2642-3901 and does not violate the copyright of the publisher*

The totality of the work presented below was undertaken by myself, with my supervisor Dr Gerardo Aragon-Camarasa and Andrew Hamilton providing edits and corrections

The version of this paper presented in this thesis has been altered to reflect the changes suggested by the examining committee

2.1 Chapter Summary

This chapter comprehensively evaluates robotics simulation software compatible with ROS version 2 (ROS2), a widely used meta-operating system. This investigation addresses a significant literature gap, as previous comparisons focused exclusively on ROS version 1 (ROS1) implementations. We devised a rigorous methodology to systematically benchmark simulation software under similar parameters, tasks, and scenarios. We also evaluated their capability for long-term operations, task success, repeatability, and resource usage. Our analysis guided the selection of NVIDIA Isaac Sim [48], a simulation platform featuring Ray Tracing capabilities. Although Isaac Sim neither demonstrated superior performance nor minimal resource utilization, its integration with Deep Learning and Reinforcement Learning frameworks proved decisive for ensuring our framework's future compatibility. Furthermore, its proprietary Riemann Motion Plan (RMP) [49] offers real-time planning capabilities, reducing dependence on MoveIt! [50], robotics' predominant motion planner. This independence enhances our framework's modularity and contributes to a more comprehensive direct teleoperation baseline, particularly significant as most current direct autonomy frameworks rely on MoveIt!, as noted in Section 1.3.

However, this work was completed in 2021 and since then, technology evolved and the focus of this thesis changed. As such some of the insight and design choices are no longer applicable in their current states. An in depth discussion has been added in Section 2.8

2.2 Abstract

Simulation software is a powerful tool for robotics research, allowing the virtual representation of the real world. However, with the rise of the Robot Operating System (ROS), there are new simulation software packages that have not been compared within the literature. This paper proposes a systematic review of simulation software compatible with ROS version 2. The focus is research in robotics arm manipulation as it represents the most often used robotic application in industry and their future applicability to digital twins. For this, we thus benchmark simulation software under similar parameters, tasks and scenarios and evaluate them in terms of their capability for long-term operations, success at completing a task, repeatability and resource usage. We find that there is no best simulation software overall. Still, two simulation packages (Ignition and Webots) have higher stability than others while, in terms of resource usage, PyBullet and Coppeliasim consume less than their competitors.

2.3 Introduction

With the advent of deep learning technologies, current research efforts have been focused on teaching robots how to perform various tasks autonomously. However, a data-driven approach is required to acquire and process the vast amount of data to effectively teach a robot how to perform an unfeasible task using a real robotic testbed. For this, robot simulation software [51, 52, 53, 54, 55] have been used to overcome the shortcomings of data-hungry AI approaches and to allow the developer to obtain a constant environment [56]. The world can be controlled in a simulated environment, including aspects that would be impractical in reality. There is also no risk of damaging the robot or human operators, and simulations allow time control, increasing the data collection speed.

Simulations are the gateway for Digital Twins, a high-fidelity representation of the physical world [57], and can allow manufacturing to increase production and flexibility of supply chains. Therefore, digital twinning consists of interconnecting a simulation software to a real autonomous robotic system to reduce the implementation time of the manufacturing process when changing a production line. A recent example of a digital twin solution for a robotic arm can be

Name	Logo	Physics Engine	Headless Support	Open Source	ROS2 Support	ML Support
Gazebo		Bullet, DART, ODE, Simbody	Full	Yes	Yes	External
Ignition	IGNITION	DART	Full	Yes	Yes	External
Webots		ODE	Partial	Yes	Yes	External
Isaac Sim		PhysX	Full	No	Yes	Integrated
Unity		Havok, PhysX, RaiSim	Full	No	No	External
PyBullet		Bullet	Full	Yes	No	External
CoppeliaSim (Vrep)		Bullet, Newton, ODE, Vortex Dynamics	Full	No	Yes	External
Mujoco	MuJoCo	Mujoco	Full	Yes	No	External

Figure 2.1: Overview of the Simulation Software and Their Capabilities

found in [58] where the authors used ROS (Robot Operating System) [59] to achieve seamless operation between the real and digital world. However, simulation software is imperfect because its physics engines do not accurately represent the real world. Furthermore, simulations allow for perfect data capture with no noise, which has powered research in deep learning approaches for robotics.

In this paper, we propose to carry out a systematic benchmark of current simulation software (Figure 2.1) to investigate their performance and suitability to perform different robotic manipulation tasks using the ROS2 (Robot Operating System version 2). ROS has become the de facto meta-operating system for modern robotic systems. We choose ROS2 because it supports a wide array of devices (e.g. micro-controllers), enabling the Internet of Things (IoT) integration. The latter is the main requirement for developing a working digital twin system. ROS2 can also be used to bridge the gap between AI-enabled robots and real-world robot control. We choose robotic arms in this paper as they are prevalent in automated manufacturing operations.

We consider two tasks for the robot arm to perform. The first task is about picking and placing an object which is a common operation in the industry. The second task consists of throwing a cube into a pyramid. We chose this throwing task as we aim to test the accuracy and repeatabil-

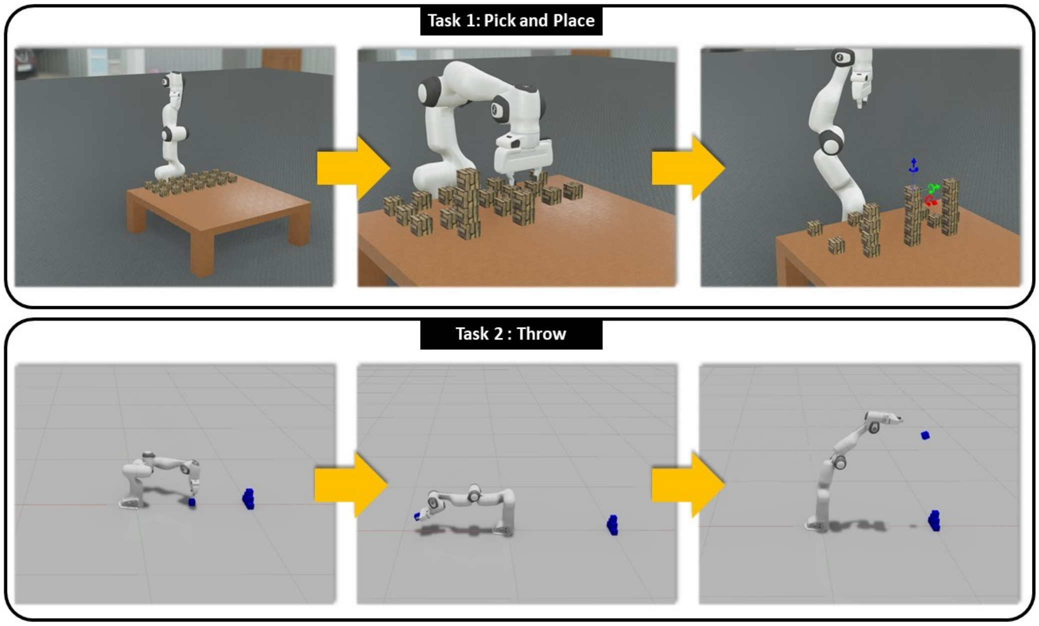


Figure 2.2: Simulation Tasks Progression over Time. Task 1 (top), is a Pick and Place task where the goal is to stack 3 columns of 5 cubes. Task 2 (bottom) is a Throw task where the goal is to collapse a pyramid of 6 cubes by throwing a cube at it.

ity of the simulation software to decide its potential suitability for building digital twins. Figure 2.2 shows an overview of the tasks. We record the resource usage of each simulation considered in this paper while performing each task in both a headless and a graphical version. Our contributions include proposing a systematic comparison of state-of-the-art robotic arm simulation software using ROS2, the state-of-the-art version of the Robot Operating System. Furthermore, we develop an experimental methodology to evaluate robot simulation software for long-term operations and their success at completing a task. We also devised an experimental validation system to evaluate the stability of robot simulation software and its capacity to repeat a given task. In the context of this paper stability refers to the ability of the simulation software to perform the task without crash.

2.4 Background

Benchmarking robotic simulation software can trace its origins to Oreback, and Christensen [23]. They were the first to propose a methodology to test robotic simulations. Their approach consisted of summarising the capabilities of three simulators, considering quantitative values such as supported OS or programming languages and qualitative opinions such as the learning curve or the difficulty of installation. They also recorded the amount of RAM used to control a real robot using the simulation software. Kramer and Scheutz [60] extended [23] and developed a comprehensive testing suite for open-source robot simulation software. They devised a set of criteria based on the software-development process. They created a feature score based on different properties such as usability, supported features (path planning, voice recognition, etc.) and faults handling. They performed a low-level task implementation on a real robot and recorded the resource usage. However, the task is scarcely described and was only repeated three times.

Before ROS [59], some roboticists used the simulation software as a middleware to send data and commands to the robot, e.g. [60]. Staranowicz and Mariottini [25] provided the first comparison of simulation software that used ROS as the communication framework to control a robot. They compared the properties of three open source and commercial simulations. They then demonstrated the capabilities of Gazebo [61], a popular simulation software with a real robot, effectively creating a Digital Twin. However, they neither recorded the resources usage nor tried a different simulator for the real-world task. Their work was then extended by Nogueira [62], who compared two simulators and their integration with ROS, the ease of building the world and the CPU usage.

Pitonakova *et al.* [26] adopted the methodology in [62]. They compared three simulators and then ran extensive tests to record each simulator's performance on tasks involving multiple robotic arms. They recorded memory, CPU usage, and the real-time factor, meaning the speed at which the simulation runs. This is vital for Digital Twining. It is also vital for machine learning, as the faster the simulation runs without compromising the physics, the faster the training of a machine learning model would be. They performed each test with and without a Graphical User Interface (GUI) and then compared the impact of rendering the simulation to a screen. Ayala *et al.* [27] and Korber *et al.* [28] followed the idea of recording resources usage during the running of the experiment. After recapitulating the properties of each simulator, they coded tasks and recorded memory and CPU usage. Korber *et al.* compared four simulation software on robotic manipulation tasks while Ayala *et al.* only compared three for humanoid robot scenarios.

In this paper, we initially consider eight robot simulation software but narrow our benchmark to five that support ROS2, including two simulation software that have not been considered in the literature. We also propose to implement pick and place and throwing tasks to investigate the advantages and limitations of each simulation software, their performance and, ultimately, their suitability for Digital Twins.

2.5 Materials and Methods

To evaluate and compare robotic simulation software, we develop our methodology and experiments guided by the following research questions:

RQ1.1 How does simulation software compare in terms of supporting long-term operations while still succeeding at completing a given task?

RQ1.2 How repeatable is the simulation software under the same scene and task constraints?

RQ1.3 Which simulation software would be more suitable for machine learning research in terms of resource usage and idle time?

2.5.1 Simulation Software

The above research questions inform our choice of the simulation software investigated in this paper, as shown in Figure 2.1. Not all simulation software has support for ROS2. For this paper, we have attempted to implement our ROS2 bridge but with limited success due to the rapid development cycle of ROS2. For completeness, we describe our experience while implementing the ROS2 bridge for the simulations we do not use in this paper. Unity's existing bridge is limited as it does not support asynchronous communications, which are the underlying communication paradigm in ROS2. Mujoco conflicts with ROS2 because the ROS2 multithreaded controller is incompatible with Mujoco's single threading nature. Finally, we had to drop Gazebo because development efforts have turned to Ignition, and there is currently an implementation error in the controller code, causing our robot to move erratically[1].

We also consider simulations that feature a headless mode. This is because a headless mode is critical in a machine learning context (ref. **RQ1.3**). Therefore, we analyse the GUI's impact on resource usage. The robot simulation software examined in this paper are:

1) Ignition [51] is a set of open source software libraries which are arranged as multiple modular plugins written in Ruby and C++. Open Robotics has developed them since 2019. It has a similar communication principle to ROS2. We chose this simulator as it is the successor of Gazebo.

2)Webots [52] has been developed since 1998 by Cyberbotics, a spin-off company from EPFL (Swiss Federal Institute of Technology Lausanne). It supports a wide range of sensors and robot controllers out of the box, is well documented and includes several examples files. Figure 2.1 has partial headless support because it only disables the simulation rendering. There is still a GUI visible. We considered it because it is one of the oldest simulation software still actively developing.

3) Isaac Sim [53] is a recent, Linux-only, simulation environment developed by Nvidia which runs on the PhysX engine and can be programmed in Python or C. By default, it integrates machine learning capabilities and has in-built plugins to generate synthetic data for domain adaptation and transfer learning. The latter is possible because of its ray tracing capabilities, allowing for a visual simulation as close to reality. While it can be run in headless mode, this is not possible while using their ROS2 extension since there is an issue with the ROS2 plugin not loading when launched from a python script instead of from the terminal.

4) PyBullet [54] is a Python-based simulation environment based on the Bullet physics engine, which has been in development since 2016. It is popular for machine learning research

[1]https://github.com/ros-simulation/gazebo_ros2_control/issues/73

as it is lightweight and easy to use. For this paper, we implemented a ROS2-compatible plugin since there is no official ROS2 support.

5) Coppeliasim [55], previously known as V-REP, is a partially closed source simulation environment developed since 2010. It can be programmed directly in Lua or external controllers in 6 other languages. We decided to include it in our research as it has been compared in previous simulation software reviews, e.g. [62, 26, 63, 27, 64].

2.5.2 Data Capturing

For data capturing and recording, we adopt the metrics mentioned in Section 2.4, namely, the processor usage (CPU), the memory usage (RAM) and the execution time (ref. **RQ1.1 & 3**). We also record task-specific data, such as the number of cubes placed or moved (ref. **RQ1.2 & 3**). The execution time is not mentioned in the literature but was added as a metric for machine learning, in which the running time can have major impact, as training a model involves several iterations of the simulation. A delay of tens of seconds for one iteration can turn into long training sessions. To accurately record each simulation, we start recording 5 seconds before the simulation starts and ends the recording 60 seconds after the task has ended. This is to ensure we record the start and end of all the processes and ensure that all process have been terminated. We record processes related to the simulation while discarding the rest, such as OS-specific processes.

2.5.3 Robotic Tasks

We consider two tasks, each divided into two sub-tasks, to evaluate each simulator considered here. A sub-task is repeated 20 times to reduce the variance during data recording and to obtain an accurate statistical characterisation of a simulation. In practice, we found that more than 20 repetitions do not result in a statistically significant difference using the Student's T-Test. The two tasks and their rationale are summarised in Table 2.1. The task execution logic is the same for all simulations. We must note that we use the default simulation parameters to set up these tasks. This is to remove bias while implementing the tasks and avoid tuning simulator-specific parameters in order to obtain an objective evaluation for each simulation software. The Pick and Place task has been chosen as a proxy for assembly tasks that requires multiple actions of picking and placing objects. Furthermore the multiple towers evaluates the abilities of the robot to manipulate in a precise manner while taking care of avoiding the environment. The Throw task was chosen to evaluate the repeatability of the physics engine and it's accuracy, despite the fact that there is no real world task that would make use of such an evaluation

Table 2.1: Sub-Task overview and Design rationale

Name	Design	Features Tested	Data Recorded and Rationale
Task 1-A: Pick, and Place	A robotic arm randomly takes 5 cm cubes from a table with 21 cubes arranged in a 7×3 grid. The task is to stack them into 3 towers of 5 cubes, as seen in Figure 2.2. We consider 3 stacks to leave cubes on the table and to allow for more diversity in each repetition. We set the limit to 5 stacked cubes due to the table's height and the robot's capabilities.	Friction, Gravity, Inertia	This experiment . addresses **RQ1.2** which analyses the numbers of cubes correctly placed. It will also test the stability and suitability of the simulation for long operations, as recorded by the number of cubes still in place at the end (ref. **RQ1.1**). The idea of stacking cubes to analyse performance is motivated from [28]
Task 1-B: Pick and Place Headless	We use the same setup as Task 1-A but without a GUI. This was chosen as in a machine learning settings experiments need to be restarted multiple times and often run on a server with no screen (ref. **RQ1.3**).		
Task 2-A: Throwing	A robotic arm will pick up a cube and throw it towards a pyramid of 6 cubes. The arm goes as far back as mechanically possible and performs a throwing motion towards the pyramid in front of it. Figure 2.2 shows the trajectory taken by the robot during this task. The cube is released at 50% of the trajectory. The pyramid is placed such that a successful throw at full power will collapse it.	Friction, Gravity, Inertia, Kinetic	This task benchmarks the accuracy and repeatability of the simulation software and addresses **RQ1.2**. The latter is carried out by recording the number of cubes displaced from their original position. This idea has been inspired by a contributor to Ignition[2] demonstrating how to interface ROS2 and Ignition.
Task 2-B: Throwing Headless	We follow the same design as Task 2-A, except without a GUI (ref. **RQ1.3**).		

2.5.4 Robot Control

There are 3 methods to control a robot using ROS2: the joint controller, the joint trajectory follower and the general purpose ROS controller. The joint controller sets the position of the joints to a given joint angle using hardware-specific interfaces of a robot. This is the simplest method as it provides no feedback to the controller regarding whether the robot performed the change in position successfully. The joint trajectory follower uses a ROS action client/server combination in which the client sends the joint position for a given trajectory. Then, the server continuously sends the current value of the joints as a feedback mechanism until the trajectory is completed. In this situation the server is the robot's interface. This method works well in practice and we have implemented it for Coppeliasim, PyBullet and Isaac Sim. For the Ignition and Webots, we use the general purpose ROS controller (`ros_control`) [65], which is not implemented for the other simulations. It provides a wrapper for the joint trajectory follower described above and different methods of control such as a velocity, effort or gripper controller.

2.6 Experiments

2.6.1 Methodology

We use a docker container with Nvidia Ubuntu 20.04 `cudagl` image for all simulators except for Isaac Sim that cannot access the base graphics driver API when using docker. Isaac Sim is thus executed in the base system from where we run all experiments. The docker overhead is ignored as only processes related to the tasks will be recorded . ROS2 Foxy has been installed, along with simulator-specific packages. Docker has been used to easily provide an image with all the necessary packages installed without conflict between different simulations. It also allows for reproducibility of these experiments by providing the same setup every time. The base system runs an Intel I7 10700 with 32GB of RAM and an Nvidia GeForce RTX 3060 with Ubuntu 20.04. We used `psutil`[3] which is a python package that records the CPU and RAM usage. Each process was monitored at 10 Hz to minimise the resource impact. We used the recommended time step for each simulator, and we fixed all simulators to run in real-time. We use the Franka Panda robot and its model and configuration files provided by MoveIt 2 [66]. The project repository can be found at https://github.com/09ubberboy90/ros2_sim_comp.

2.6.2 Implementation

The implementation comprises four components as shown in Figure 2.3 and as noted below.

[2] https://github.com/AndrejOrsula/ign_moveit2

[2] https://github.com/AndrejOrsula/ign_moveit2

[3] https://pypi.org/project/psutil/

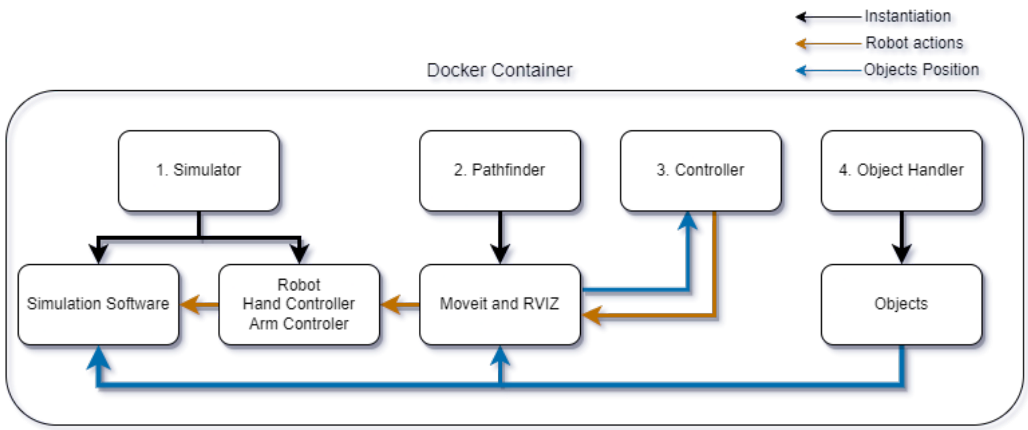


Figure 2.3: Implementation Diagram

1 *Simulator:* launches the simulation software and spawns the robot along with the gripper and arm controllers.

2 *Pathfinder:* launches Rviz (a ROS data visualisation software) and MoveIt 2.

3 *Controller:* chooses which object to grab from the list of collision objects advertised by MoveIt 2, shown in Figure 2.3 as blue arrows. Then, the pick and place or throwing task is executed accordingly.

4 *The Object-Handler:* spawns the objects and publishes their position to the planning scene of MoveIt 2 at 2 Hz. We choose 2 Hz because the scene's rate of change in real time does not deviate considerably. Higher rates consume more CPU usage, which impacts the performance of all simulations in this paper.

In our implementation, both the arm and the gripper are controlled using a position controller. We must note that the gripper controller available in ROS1 has not yet been ported to ROS2. The latter causes issues while grasping objects in the simulations (except Webots) as the amount of force is constant with no feedback. To mitigate this issue, we command the gripper to close its gripper beyond the optimal closing distance to ensure that the object is grasped correctly. Webots does not have this issue because it implements a PID controller for each simulated motor by default. These 4 components are launched using a single ROS2 launch file with ad hoc delays to ensure everything is started, as Parts 3 and 4 do not work unless part 2 is launched. For each simulation software, we are using the default physics time step. The physics engine is also the default, except for Coppeliasim, in which we use the *Newton physics engine* because the other supported physics engines cause the gripper to fail to grip the cube.

2.6.3 Experiment 1

Table 2.2 shows the result of task 1, which addresses **RQ1.1 & RQ1.2**. The reason the task times out (failure in Table 2.2) is because the ROS controller fails to start, or in the case of

Table 2.2: Task 1: Percentage of Cubes Placed

Name	Failure (%)	Cubes Placed (%)			Cubes at the end of the task (%)		
		Min	Mean	Max	Min	Mean	Max
Ignition	0	60	91	100	47	82	100
Ignition GUI	0	80	94	100	67	88	100
Isaac Sim GUI	10	47	89	100	20	65	100
Pybullet	0	7	18	47	7	15	47
Pybullet GUI	0	7	11	40	7	11	40
Coppeliasim	30	67	92	100	47	79	100
Coppeliasim GUI	15	53	91	100	47	76	100
Webots	5	53	88	100	47	79	93
Webots GUI	5	73	91	100	40	80	100

Coppeliasim, the client refuses to connect to the simulation server. The rest of the metrics only focus on successful attempts.

Ignition and PyBullet did not have timeouts; however, PyBullet performs significantly worse at stacking 5 towers than the other simulators as 15% of the cubes on average (i.e 3 cubes) were correctly positioned at the end of the simulation, and, therefore, the robot does not encounter scenarios where it collapses towers. Ignition and Webots are the best performing simulations for executing the task of stacking cubes and keeping the cubes in place. Coppeliasim and Isaac Sim, are carrying out the task well at placing the cube in the right place but, tend to have situations where the robot collapses the towers. Furthermore, while Coppeliasim achieves 92% success of placing cubes, we can observe that it often times out, and reduces its overall success. We can also observe in Table 2.2 that there is no statistically significant difference between headless and GUI modes, using the Student T-Test. These results suggest that Ignition (headless and GUI) succeeds at completing the task more frequently using the default parameters (ref. **RQ1.1**) and has less variation over different attempts (ref. **RQ1.2**).

Table 2.3 shows that PyBullet headless consumes fewer resources overall, while Isaac Sim, is the most memory-intensive simulation as it consumes 10 times more RAM than the next simulator (Webots GUI). This is in line with the current trend of Pybullet being used in machine learning research (ref. **RQ1.3**). It is worth noting that Coppeliasim uses fewer resources with a GUI than headless. We speculate that this is because it was initially designed as a GUI application, with headless support only added at a later date, thus having received less development focus.

Figure 2.4 shows the spread of the start time and end time for each simulation (ref. **RQ1.3**). As mentioned in Section 2.6.2, Isaac Sim has to be started manually, thus the time that takes the simulation to start is not captured in the plot. Ignition takes the most time to load because it requires an additional movement to the start position. Webots finish the earliest with little

Table 2.3: Task 1: Resources usage

Name	CPU (%)	RAM (MB)
Ignition	202 ± 94	686 ± 313
Ignition GUI	205 ± 51	775 ± 159
Isaac Sim GUI	134 ± 25	10070 ± 1885
Pybullet	117 ± 41	**663 ± 224**
Pybullet GUI	140 ± 28	919 ± 168
Coppeliasim	135 ± 42	860 ± 262
Coppeliasim GUI	**100 ± 43**	850 ± 379
Webots	144 ± 72	1191 ± 550
Webots GUI	162 ± 61	1322 ± 410

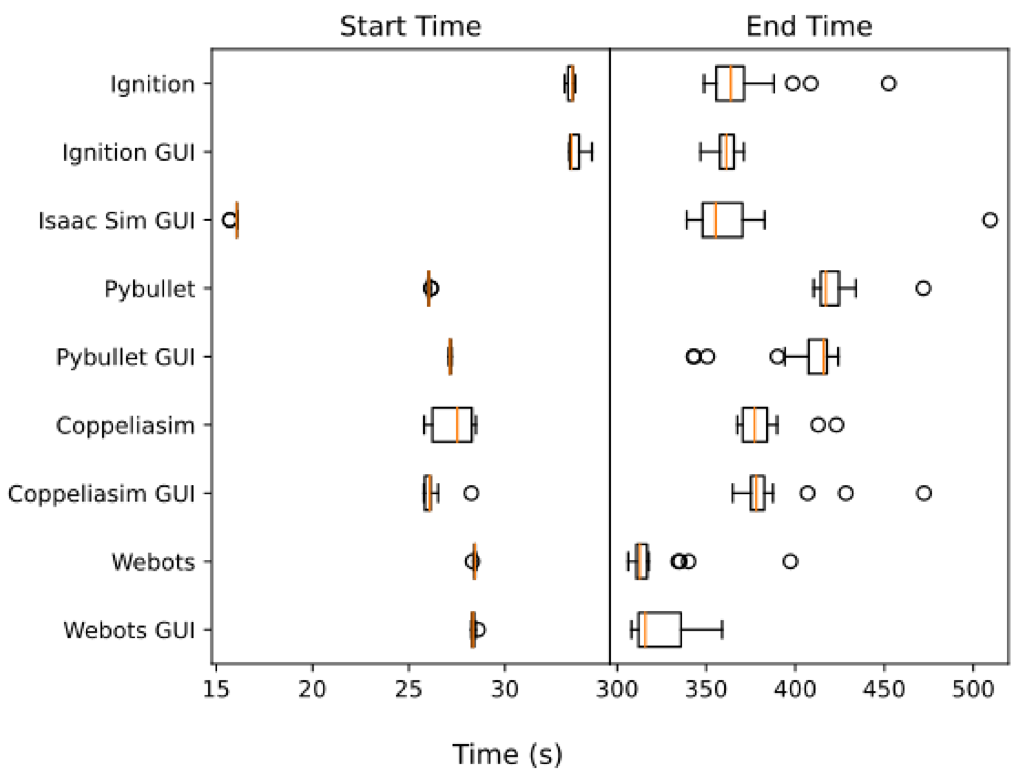


Figure 2.4: Task 1: Mean task start and end time

variation. Combined with its relatively high success rate from Table 2.2, Webots appears to be ideal for prototyping robotic tasks and for machine learning experiments due to its relatively high success rate from Table 2.2 and finishing the task and simulation early with low variation. PyBullet, on the other hand, takes the most time and combined with its high failure rate (with the default parameters), it may not be suitable for machine learning as it would take more time to run a single experiment. Similarly, further parameter tuning would be required in order to obtain a stable simulation that succeeds at completing the task.

Table 2.4: Task 2: Percentage of Cubes Placed

Name	Failure (%)	Cubes Moved (%)		
		Min	Mean	Max
Ignition	10	0	0	0
Ignition GUI	0	0	0	0
Isaac Sim GUI	0	0	18	83
Pybullet	0	0	0	0
Pybullet GUI	0	0	0	0
Coppeliasim	32	0	9	50
Coppeliasim GUI	20	0	0	0
Webots	5	0	11	50
Webots GUI	15	0	21	50

2.6.4 Experiment 2

As shown in Table 2.4, which focuses on **RQ1.1 & 3**, only Webots throws consistently. Isaac Sim consistently manages to throw the cube but fails to hit the pyramid as the motion behaviour is not deterministic. We speculate that this is because we did not tune the simulation parameters and used the default values. Coppeliasim and PyBullet manage to hit the pyramid, but the behaviour is rare as the few times the arm manages to successfully perform the throwing motion, the throw is not always at the same distance nor perfectly aligned. Coppeliasim has a high timing out rate (failure column in Table 2.4) due to the reasons mentioned in Section 2.6.3. Finally, for Ignition, the success at hitting the pyramid is zero. We observe that, in most cases, the cube falls in transit, especially when the arm is as far back as possible and starts to move at full speed for the throwing motion. At this point, the robot and the cube have the highest moment of inertia, and if the friction between the cube and the gripper is not enough, the cube falls. We must note that we fix the friction parameter to explore the default capabilities for each simulator. We also notice that there are instances when the robot manages to throw the cube but does not hit the pyramid. This is because the gripper controller had a delay in opening its gripper, changing the thrown cube landing spot.

Table 2.5 shows similar results to task 1. Coppeliasim uses the lowest amount of CPU while Ignition uses less memory. The CPU usage for all simulations observes less variation. This could be due to the simplicity of the world and the short time of execution. As mentioned in Section 2.6.3, Coppeliasim still uses fewer resources with a GUI than headless. Figure 2.5 shows similar start and end times for all simulations, observing lower variations compared to task 1. The reason for this is the relatively short time of execution and the low amount of path planning that can fail and delay the execution. For this scenario, considering only the time of execution will not have an impact on the choice for a machine learning approach as the difference between execution is minimal. If resource usage is important, then Coppeliasim should be considered for

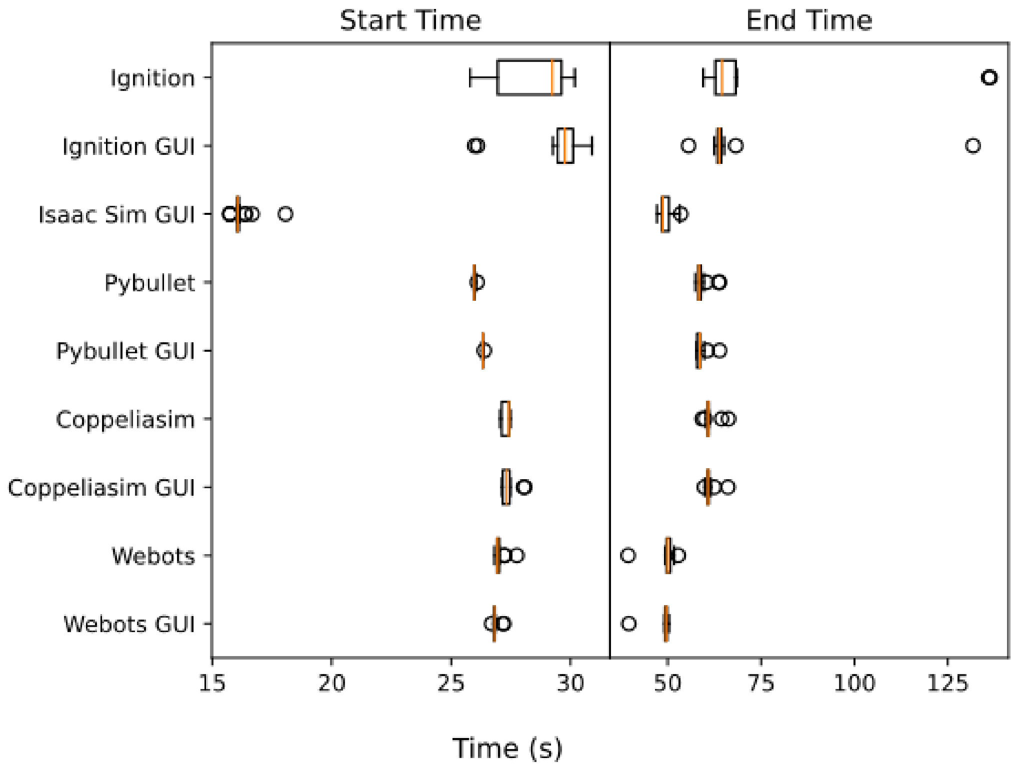


Figure 2.5: Task 2: Mean task start and end time

Table 2.5: Task 2: Resources usage

Name	CPU (%)	RAM (MB)
Ignition	152 ± 99	**524 ± 340**
Ignition GUI	137 ± 103	574 ± 371
Isaac Sim GUI	148 ± 48	11123 ± 1865
Pybullet	139 ± 32	730 ± 174
Pybullet GUI	136 ± 31	828 ± 205
Coppeliasim	115 ± 37	742 ± 185
Coppeliasim GUI	**98 ± 28**	838 ± 214
Webots	141 ± 74	1275 ± 261
Webots GUI	134 ± 72	1207 ± 198

machine learning tasks. Otherwise, a more successful simulation should be considered such as Webots.

2.7 Conclusion and Future Work

In this paper, we have investigated current robot simulation software performance and their suitability to perform two different robotic manipulation tasks. We have also developed a methodology to systematically benchmark robot simulations under similar parameters, tasks and scenarios. Based on our experimental results, Webots appears to be the more suitable for long-term operations while still succeeding at completing a given task (ref. **RQ1.1**) and be able to replicate the same simulation conditions across attempts (ref. **RQ1.2**). Webots would only be suitable for machine learning if the execution time and resources are not a requirement while training machine learning models (ref. **RQ1.3**). Ignition, while comparable to Webots, is more suited to answer **RQ1.1 & RQ1.3**. **RQ1.2** is only satisfied if the task is slow moving and constant. We must note that Ignition is still in development and some of the challenges we encountered while implementing both tasks and carrying out our experiments may be mitigated in the future. Coppeliasim and PyBullet have less impact in terms of resource usage and are the most suited to answer **RQ1.3**. That is, Coppeliasim provides better stability for task success at the cost of timing out more often. Finally, Isaac Sim only satisfies **RQ1.1**, as the simulated scene was not repeatable across attempts.

From our review and experimental results, we found that current robot simulation software could not be used to develop a digital twin. This is because the simulators considered in this paper cannot maintain a repeatable simulated scene over time. We hypothesise that a continuous feedback mechanism is needed between the simulation and reality similar to [67] in order to maintain an accurate representation of the real environment. While this paper focused on benchmarking robot simulation software, future work consists of optimising each simulator to minimise failure rates and maximise task completion, and benchmark them accordingly. Additionally, the Unreal Engine plugin for ROS2 has recently seen more development and could potentially replace Unity in our original plan. We also aim to specifically benchmark each simulation in a machine learning context, such as in [68] with the view to developing a digital twin that can take advantage of a simulated environment to deploy AI solutions for autonomous robotic systems.

Acknowledgment

This research has been supported by EPSRC DTP No. 2605103 and NVIDIA Corporation for the donation of the Titan XP GPU.

2.8 Discussion

This paper was written in 2021, and since then, technology has evolved significantly, presenting new opportunities for refinement and innovation. While the experimental design and methodology described in this chapter provided valuable preliminary insights, they also highlight areas where future research could build upon and improve these findings.

The experimental design could be enhanced by incorporating tasks such as "peg-in-the-hole," which is widely recognized as a robust proxy for assembly tasks in real-world scenarios. This would allow for a more comprehensive evaluation of simulation software capabilities and their applicability to practical use cases.

Additionally, while our experiments utilized a fixed starting position for the robot and predefined pick-and-place actions, future studies could introduce randomized starting positions and dynamic pick-and-drop locations. This approach aligns better with the modular and adaptive requirements of Industry 5.0, creating a deeper understanding of how simulation software can support diverse operational scenarios[16].

Our decision to keep default parameters likely influenced the results, as factors such as timestep, sampling periods, and physics parameters significantly impact simulation stability [69]. For example, the grasping issue noted in Section 2.6.2, where the gripper was commanded to close beyond its optimal distance, could be mitigated through parameter tuning tailored to each simulator's unique characteristics.

Despite these limitations, this paper serves as an important stepping stone for future research. The insights gained here can inform more rigorous experimental designs that leverage statistical tests to produce meaningful and actionable results. By addressing these areas of improvement, subsequent studies can generate deeper insights into the capabilities of simulation software.

Finally, while Isaac Sim did not perform optimally in certain tasks such as throwing, its additional features, including built-in machine learning support, real-time motion planning based on Riemannian Motion Policies [70], and photorealistic rendering, make it an excellent choice for advancing teleoperation frameworks. These features proved pivotal in developing the modular teleoperation system described in Chapter 3, demonstrating Isaac Sim's potential to drive innovation despite initial performance limitations. The update after this experiments has reduced the amount of RAM usage by 2 and further enhancement were added each following updates.

Chapter 3

TELESIM: A Modular and Plug-and-Play Framework for Robotic Arm Teleoperation using a Digital Twin

This section (Section 3 reproduces the author's version of the accepted, peer review manuscript presented at the 2024 IEEE International Conference on Robotics and Automation (ICRA) as Florent P Audonnet, Jonathan Grizou, Andrew Hamilton, and Gerardo Aragon-Camarasa. TELESIM: A Modular and Plug-and-Play Framework for Robotic Arm Teleoperation using a Digital Twin. In 2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17770–17777, Yokohama, Japan, May 2024. IEEE. ISBN 9798350384574. doi: 10.1109/ ICRA57147.2024.10610935. URL Link and does not violate the copyright of the publisher*

The totality of the work presented below was undertaken by myself, with my supervisor Dr Gerardo Aragon-Camarasa providing edits and corrections. Andrew Hamilton helped provide materials for our research, specifically the senseglove. Jonathan Grizou helped refined the user questionnaire

The version of this paper presented in this thesis has been altered to reflect the changes suggested by the examining committee

3.1 Chapter Summary

This chapter outlines the development and evaluation process of our initial framework, TELESIM, designed to assess user performance in direct teleoperation utilizing the simulation software selected in Chapter 2. By implementing TELESIM alongside literature-informed user questionnaires, we addressed the research gap identified in Section 1.3, particularly regarding the scarcity of user evaluations in direct teleoperation research. Our experimental protocol involved participants operating two distinct robots through various User Interface Device (UID), employing direct visual feedback, as illustrated in Figure 3.1. The framework demonstrated robust modularity and plug-and-play capabilities, successfully accommodating multiple robots and UIDs. However, our initial investigation remained geographically constrained to a UK population, necessitating broader demographic sampling for result generalization.

The work presented in this chapter is explained in greater details in Chapter 4, specifically, the hardware setup and experimental design due to the publication page limit

3.2 Abstract

Teleoperating robotic arms can be a challenging task for non-experts, particularly when using complex control devices or interfaces. To address the limitations and challenges of existing teleoperation frameworks, such as cognitive strain, control complexity, robot compatibility, and user evaluation, we propose TELESIM, a modular and plug-and-play framework that enables direct teleoperation of any robotic arm using a digital twin as the interface between users and the robotic system. Due to TELESIM's modular design, it is possible to control the digital twin using any device that outputs a 3D pose, such as a virtual reality controller or a finger-mapping user input device(UID). To evaluate the efficacy and user-friendliness of TELESIM, we conducted a user study with 37 participants. The study involved a simple pick-and-place task, which was performed using two different robots equipped with two different control modalities. Our experimental results show that most users were able to succeed by building at least a tower of 3 cubes in 10 minutes, with only 5 minutes of training beforehand, regardless of the control modality or robot used, demonstrating the usability and user-friendliness of TELESIM.

3.3 Introduction

Robot teleoperation is difficult for non-experts [12, 13]. Recently, the ANA Avatar XPRIZE Challenge [9] set a series of challenging tasks to test the limits of teleoperation. The best systems that completed the challenges were rewarded with a prize pool of $10 million. At its core, the challenge involves the direct teleoperation of a robot with minimal latency and the capacity to experience the environment from the robot's perspective. However, direct teleoperation still places a heavy physical and mental strain on the user, as Pettinger *et al.* [30] reported that a user performing a pick and place task was faster and had fewer errors while reporting the task was more accessible when shared autonomy systems were enabled. Hence, researchers from HCI, medicine, robotics, and others have explored different means of control for teleoperation

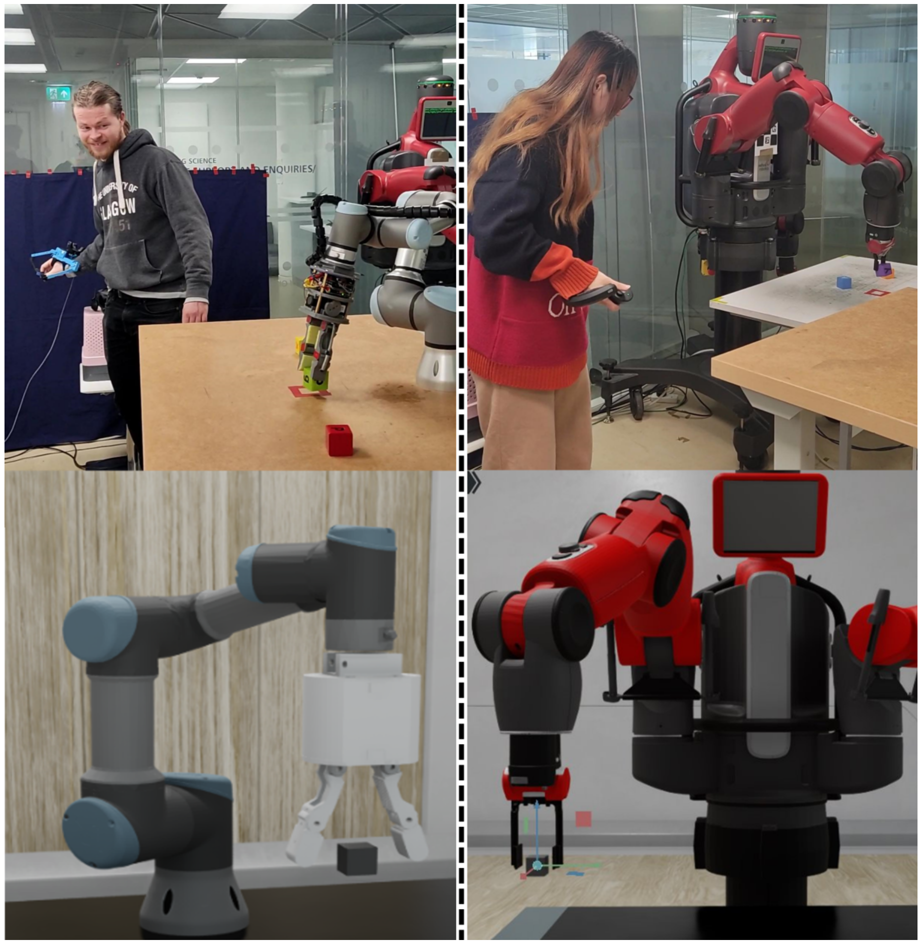


Figure 3.1: Our modular and plug-and-play TELESIM framework is being used to control a UR3 Robot (top-left) and a Baxter Robot (top-right) and its digital twin (bottom-right).

to address these limitations. While most research efforts focus on a physical control device such as a Virtual Reality Controller [11, 32], a Joystick [21, 71], or phone [22], others decided to use cameras to track the whole body [72, 29], or just the gaze [73]. There has yet to be an overall consensus on the most appropriate type of control for direct teleoperation with specific applications requiring specific implementations.

In this paper, we develop a modular and plug-and-play direct teleoperation framework called TELESIM that non-experts can use without specialised training using off-the-shelf Virtual Reality (VR) technologies. Specifically, our objective is to allow for the direct teleoperation of any robotic arm using a digital twin as the interface between the user and the robotic system. According to Semeraro (2021), a digital twin can be defined in various ways. However, in the context of this paper, a digital twin is defined as a virtual representation or model that engages with a physical system throughout its life cycle. We then demonstrate TELESIM's user-friendliness using a user study and the users' success rate at completing the task using two different types of control and grasping systems. Specifically, we use a virtual reality controller and a finger mapping user input device(UID) mounted on two robotic manipulators using different grasping

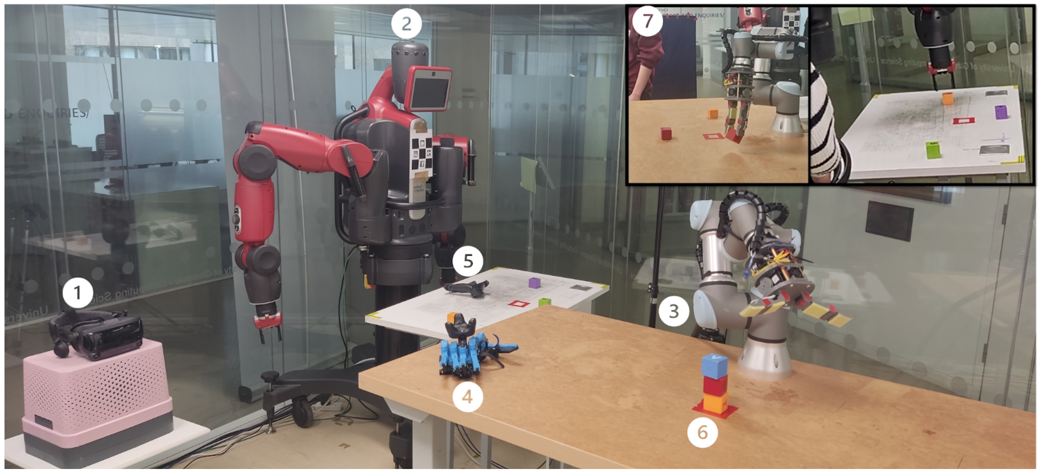


Figure 3.2: Overview of the experimental setup. The Steam Index VR Headset [2] is marked as (1) on the far left, which acts as the world's frame. The Baxter robot on the left (2) is controlled by the Steam Index VR controller (5). In front of it, the UR3 is on the right (3), with the Yale OpenHand T42 gripper [3], controlled by the Senseglove and HTC Vive tracker (4) on the left side of the brown table. Additionally, in the upper right corner (7), a view of the starting setup of the task, which consists of 3 cubes in a triangle pattern (described in Section 3.6), while on the brown table, the cubes are arranged in the goal configuration (6).

systems. We compare their performance to study whether additional degrees of freedom in the control scheme enhance performance while performing a simple task. Our contributions are:

- A modular and plug-and-play framework for teleoperation for any robotic arm using a digital twin.
- An experimental validation for testing the framework's performance through a simple non-expert task.
- A rigorous evaluation involving 37 participants demonstrating the user-friendliness of TELESIM.

3.4 Background

Direct teleoperation is considered a stepping stone for shared autonomy [39]. This is because direct teleoperation causes significant cognitive strain on the user [30], and the user may not be capable of millimetre-scale adjustment to the position of the robot end effector. While in medicine, the user's movement is scaled down to allow for more precision [31, 32], it may not be suitable for all types of manipulation tasks as some require significant arm movements to move an object from one place to another.

Hence, researchers have explored different UID to reduce the cognitive strain while giving the highest precision. For instance, low degree-of-freedom UIDs such as a keyboard [41], a joystick [74, 21], a touchscreen [42], or a gamepad [75] have brought an improved level of control [41] to address the user's mental strain. However, with the advent of VR technologies, researchers have investigated whether these technologies are appropriate for direct teleoperation. For example, they have proposed using a VR controller such as [43, 11] or a phone [76]. While others have investigated the use of motion mapping of the user's body [72, 77] or only gaze control [73]. However, for the former, the added mobility generates a higher cognitive load [30], and mapping motions to robot movements is challenging due to differences in kinematics chains between robot arms and users [78].

Recently, Gottardi *et al.* [38] have investigated combining multiple control systems, such as a VR controller and a tracking band on the upper arm, to track the user's movements. Rakita *et al.* [32] also compared various UIDs: a stylus, a touchscreen, and a VR controller. These were then integrated into a custom inverse kinematics solver that adjusted the tolerance level when matching the end-effector pose to that of the user. The authors showed that users preferred the VR controller as they were more successful at completing pick-and-place tasks, such as picking up bottles or plates. To mitigate the limitation of direct teleoperation, researchers have focused on how much shared autonomy improved the success of a given task. For this, research works have aimed at comparing direct teleoperation with respect to an assisted version to analyse the impact of shared autonomy on task success. For example, Chen *et al.* [72] created a system in which the operator, using a joystick, manipulated the robot's end effector to an object, and then the robot could either grasp the object autonomously or assist the user in fine-tuning the robot position for a more optimal grasp. Later, [29, 30, 38] built on [72] where the user teleoperated the robots directly to a planned position but allowed the robot to perform the grasp automatically or, in the case of [30], turn a valve handle. Lin *et al.* and Gottardi *et al.* conducted a user survey and confirmed that users preferred the shared autonomy approach, as it reduced complexity and mental strain. Furthermore, [38] observed results similar to [71], who hypothesised that users preferred to give up control if it meant increasing the task completion rate. However, Javdani *et al.* [71, 40] have falsified this hypothesis using a system similar to [29, 30, 38]. The authors concluded that users preferred to lose control if it meant an increase in a task's success rate only for a more complex task, while, for simple tasks, users still preferred to have more control. These works have focused on one robotic system and conducted their experimental survey on a small user base (between 8 and 23 participants). Furthermore, they focused on different autonomy levels and not on different UIDs. Finally, they used either MoveIt! or their planning interface for controlling the robot.

Although this paper focuses on TELESIM as a framework, our evaluation also addresses four main limitations of previous work: (1) researchers have only used one robot per study, (2) with either MoveIt! or their own implementation of path planning, (3) using either one or more tracking methods and (4) testing their system with user studies with a small user base, which does not represent a statistically significant sample. It also addresses a gap discussed by Rea &

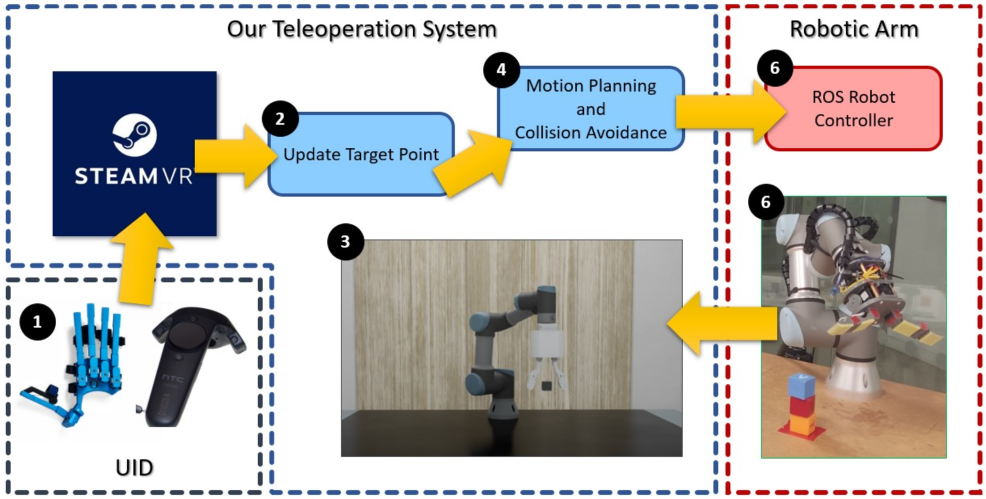


Figure 3.3: Overview of TELESIM, our direct teleoperation framework. The User Input Device (UID) (in the black dotted line) (1) can be any system that outputs a 3D pose, while our framework is depicted in the blue dotted line. It accepts the pose given by (1) to update the 3D pose of a cube in the digital twin. The robot then calculates a path to this cube in real time while avoiding collision with the world (4). Finally, TELESIM can be plugged into any robotic system (6) via a ROS2 robot controller (5), as shown in the red dotted line.

Seo [12], which states that there needs to be more non-expert evaluation of robotic teleoperation for general tasks such as picking and placing common objects. Therefore, to advance the state-of-the-art in robotic teleoperation, we investigate the performance of different control modalities for direct teleoperation using a VR controller and finger mapping as different ways of interacting with the gripper while relying on VR tracking. Similarly, we evaluate our framework on two different robots, a Rethink Robotics Baxter and a Universal Robotic 3, which demonstrates the modularity of our system, while MoveIt! and other systems needed to be fine-tuned to work [79]. We ask 37 non-expert participants to carry out a simple pick-and-place task. Additionally, by using two different control modalities, our goal is to bridge the gap between physical UIDs [43, 11] and complete body mapping [72, 29] by investigating the performance of hand and finger tracking through a SenseGlove.

3.5 Direct Teleoperation Framework

TELESIM (Fig. 3.3) consists of 3 components. The User Input Device(UID) (1) can be any control system that outputs a 3D pose. In this paper, the UID is either a VR controller or a SenseGlove with an HTC Vive Tracker. Our framework (2) calculates a motion plan to the pose given by (1) and sends it through ROS2 (Robot Operating System) [80] to any robotic arm that supports ROS control (3). In this paper, we use either a Baxter robot or a UR3 robot. The following subsections detail the design and implementation of each component.

3.5.1 Hardware Setup

The hardware used in this paper consists of a SteamVR [81] tracking a Steam Index controller and HTC Vive Tracker, as shown in Figs. 3.2 points 5 and 4, respectively. The Baxter robot (point 2 in Fig. 3.2) is controlled using the Steam Index VR controller, and the gripper can be closed by pressing the main trigger on the Steam Index VR controller. These conditions will henceforth be referred to as *Red Design*. The UR3 robot (point 3 in Fig. 3.2) is controlled using a Senseglove development kit, which allows for mapping of all individual finger movements, with an HTC Vive Tracker mounted on top of the middle finger of a T42 gripper from the Yale OpenHand project [3]. These conditions will henceforth be referred to as *Blue Design*. These UIDs are represented as point 1 in Fig. 3.3.

The T42 gripper is an underactuated 2-finger gripper that we modified to have a 1 degree of freedom per finger by fixing the 2nd joint of each finger, counting from the base. The interface with the Senseglove is thus restricted to movements of the user's thumb and index finger, and these are mapped to a single degree of freedom to result in the opening and closing of the gripper. We adopted this approach to compare the Steam Index VR controller and the Senseglove fairly. However, it still allowed us to test the increase in complexity of adding an extra degree of freedom for the user to control. We also developed a custom control loop for the gripper using two separate Arduino UNOs, as they only have one serial port per board. Joint positions are converted from degrees to motor commands and then sent to the first Arduino through a serial interface. Then, commands are relayed to the second Arduino through I2C and transmitted to the two MX-28 Dynamixels [82]. The second Arduino is also responsible for reading the motors' current position and load and transmitting it back to the computer. Additionally, we restricted the amount of force the user can transmit to the gripper, as without it, it leads to breakage of the controlling string or exceeding the amount of resistance allowed by the motor. We decided not to implement haptic feedback as it would give the users an advantage of sensing whether an object is grasped. Thus, this will result in an unfair comparison between the VR controller and the SenseGlove. Therefore, we leave haptic feedback for future work.

The VR headset (1 in Fig. 3.2) acts as the origin of both robots, giving the user an easy reference point for teleoperation. The SteamVR outputs the UID position in a 3D space with respect to the headset. The origin is thus transformed into the user's resting hand position when initiated. This method of tracking the position is preferred by multiple researchers [30, 83], as well as many of the participants in the ANA Avatar XPRIZE Challenge [11, 10]. The Senseglove is also used by the winning team [10], but to control a Schunk robotic hand that replicates a human hand.

3.5.2 Digital Twin

The position in the 3D space from the SteamVR is transmitted through ROS2 to a full digital twin created in NVIDIA Isaac Sim [48]. This flow of information can be seen in Fig. 3.3 point 2. Isaac Sim, a recent ray-tracing simulation software, is used to calculate the robot motion plan using RMPFlow [49], which is a motion generation based on Riemannian Motion Policies [70] (Fig. 3.3 point 3). Isaac Sim was chosen as it is the most realistic simulation software compatible with ROS2 [5].

Isaac Sim takes in a URDF (Unified Robotics Description Format) of a robot for visualisation along with a robot description file, describing the joints that can be actuated by the motion planner and the robot's collision as spheres, as Isaac Sim uses them for collision checking. This file can be created with an included extension (Lula Robot Description Editor [84]). The ability to add new robots is a significant part of what makes TELESIM modular, along with ROS2.

We decided to use ROS2 as it is the most used framework for controlling various robots, making our framework plug-and-play. Views of the UR3 and the Baxter robot from Isaac Sim are shown in 3.1. The grey square in between the gripper (Fig. 3.3 point 4) is the point that is controlled by the teleoperation system and indicates where Isaac Sim must find a path. The fact that Isaac Sim acts as a complete digital twin allows the robot to avoid collision with the world around it and damage itself. Additionally, both robots have systems that allow them to work alongside humans. Our system is capable, with minimal configuration, of handling the restrictions placed by their need to be safe around humans. Finally, Isaac Sim transmits the position of each joint to the real robot through ROS2 and passes this information to the robotic system, as shown in Fig. 3.3 points 5 and 6.

3.5.3 Robot Control

ROS control [65] provides a wrapper that facilitates the interface of different robots. However, each robot needs to be adapted to work with specific hardware. For this paper, we implemented the Universal Robot ROS2 Control package to transfer the joint states from Isaac Sim to the robots (Fig. 3.3 point 5). Specifically, for the UR3, the size of the gripper and the safety regulations of the laboratory reduced the available workspace for the robot. We handled these limitations by adding safety planes and limitations on the range of motion of the real robot. For Baxter, we used a ROS1 to ROS2 bidirectional bridge, as Baxter only works with ROS1 internally. The pipeline is thus converting Isaac Sim's joint states into Baxter messages in ROS2 and then they are sent through the bridge as ROS1 messages to the robot. The robot outputs its current state, which is converted to ROS2 through the same bridge.

Our framework introduces a slight amount of lag (100-150 ms) between the user movement and the robot movement, partly due to the path planning step, the time taken by the actuator to move the robot arm to the desired position and mostly due to safety restrictions that caps the maximum speed of our robots. The faster the user moves from one position to another, the higher the delay as the arm tries to catch up. The user easily accounts for this by making small and slow movements, as confirmed by Schwarz and Behnkle [85]. Although we acknowledge lag is present in our system, Schwarz and Behnkle [85] found that minor delays (up to 200ms) do not impact performance. None of our users mentioned lag as an issue in our experiments (see Section 3.7). We recorded the amount of lag by following the same principle used by Rakita *et al.* [32], which recorded the scene using a 100Hz camera starting with both the robot and human at a complete stop. We recorded the difference between the starting frame of the movement by the operator and the starting frame of movement for the robot, using multiple manual keypoints on the user's hand and the robotic gripper. This approach has some subjectivity due to identifying which frame the movement started.

3.6 Experiments

3.6.1 Methodology

As described in Section 3.5.1, the user controls a robot by teleoperating it using either a VR controller or the SenseGlove. For this experiment, we consider a simple task where the user has to pick up three cubes on a table. The user must then bring them individually to the centre of the table to complete a tower. Once the tower is completed, the cubes are returned to their original position, and the user is asked to repeat the task as many times as possible within 10 minutes.

The task definition consists of three cubes positioned at each vertex of an isosceles triangle and at similar distances from the robots' base on each robot's table (Fig. 3.2 point 7). We have placed markers of the vertices of this isosceles triangle on each robot's operating table for repeatability of the task between attempts and among users. The front cube at the top of the triangle is positioned such that it is at the maximum reach limit of each robot's overhead rotation. This means that the end-effector's z-axis is perpendicular to the table, as shown in Figure 3.4, which makes it difficult for the user to pick up the cube from overhead. Thus, the user has to add some rotation in the x- or y-axis of the gripper to pick the cube successfully. The right cube is the furthest away from the user in both robots and adds a degree of difficulty due to the user's viewpoint, but still within reach. Finally, the left cube is placed such that the user has to move their body. That is, for the Red Design, the location where the user needs to pick up the left cube is approximately at the waist of the user, while for the the Blue Design, the location is on the other side of (farthest from) the headset. The cube positions were chosen to let users be spatially aware of their position and its relationship to the robot. Finally, the location where the users need to stack the three cubes is easily accessible by the robot, and this position is marked by a red square in red tape, two cm larger than the cube. This position can be seen in Fig. 3.2 point 6, with the cubes stacked as required. For a better view of the setup, we refer the reader

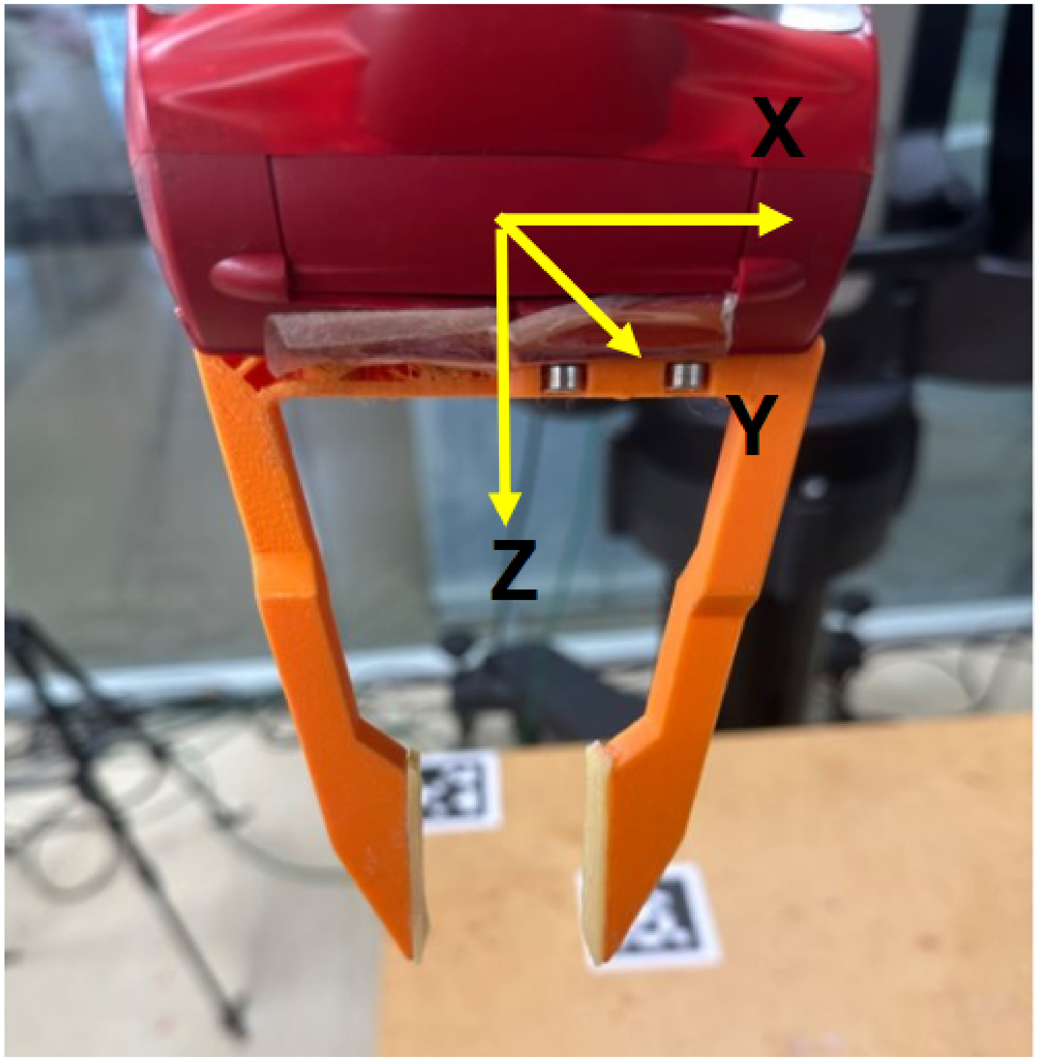


Figure 3.4: Baxter Gripper Frame

to the video attached[1]. To teleoperate the Red Design, the users need to stand with their back to the VR headset, while for teleoperating the Blue Design, users need to stand with the headset on their right and the Baxter robot behind them. This difference in position is due to the space constraint of the room in which we ran our experiments. The operating room can be seen in Fig. 3.2, with the headset on the left of the picture shown as point 1.

3.6.2 User Survey

In our experiments, we asked 37 participants (29 male and 8 female) from various backgrounds aged 19 to 51 (mean: 25.32, standard deviation: 6.26) to teleoperate both robots and stack three cubes without a monetary reward. This experiment was approved by the University of Glasgow Ethics Committee (Application Number 300220026). We did not intend for our experiment to be influenced by the participants' gender, but we have included this information for completeness.

[1]https://cvas-ug.github.io/telesim

Furthermore, our study had more participants than previous research, e.g. Javdani *et al.* [40], which only had 23 participants. Lastly, we observed that we had reached the saturation point, where the inclusion of each new participant did not significantly alter the mean and standard deviation. Participants reported having, on a 5-point Likert scale (going from "No Experience" with a score of 1 to "Experienced" with a score of 5), a 2.97 mean experience with Virtual Reality with a standard deviation of 1.2. They also reported having a mean experience of 2.76 with a robot with a standard deviation of 1.24. These questions are all relative to the user's understanding.

Each participant completed the short questionnaire described above at the beginning of the experiment. After being asked to position their back to the VR headset, an explanation was given on how to control the robot, emphasising that all of their hand movements and rotation will be mapped one-to-one to the robot. They were instructed to try to grasp a cube from both sides. They had 5 minutes to get used to the control without a specific task objective. Most of the participants picked up and placed a cube during this time.

After 5 minutes, the participants performed the task of stacking the 3 cubes in the given location without any restriction on the cubes' pose and order. Users were asked to stack cubes as many times as possible in 10 minutes. Once a tower has been completed, we pause time and reset the cubes to their initial configuration. Users' actions were recorded, such as the time taken for each tower and for individual actions for each pick, place, and drop (i.e. failures).

After 10 minutes, users were allowed to take a break while answering the Single-Ease Question (SEQ) [86]. SEQ is a 7-point Likert-scale question asking, "How difficult was the task?". It was chosen instead of other metrics such as the System Usability Scale (SUS) [87] as Hodrien and Fernando [86] have argued that it is a good end-of-task metric. Then, they were asked to repeat the same experiment but with the UR3 robot. While this could lead to some learning bias, the fact that every participant performed the experiment this way, we expect the bias to be inconsequential. Additionally, the results for the the Blue Design (UR3 with T42 Gripper controlled by the Senseglove) described in Section 3.7 do not show any improvement compared to the Red Design (Baxter controlled with the Vive VR controller. This leads to the experiments taking a total time of $2 \times (5+10)$ min, so 30 minutes plus approximately 5 minutes of questionnaires.

3.7 Evaluation

Fig. 3.5 shows that 85% of the participants can build at least one tower in 10 minutes using the Red Design and the VR controller. However, there is a steady decline for each of the following towers, with only 5% of the users able to build 8 towers. This is in direct comparison to the the Blue Design as shown in Fig. 3.6, with slightly less than 50% of the population failing to build one tower and 5% managing to build 4, half as many towers for the Red Design.

The box plots in Fig. 3.5 and 3.6 show the median and interquartile range of the time taken

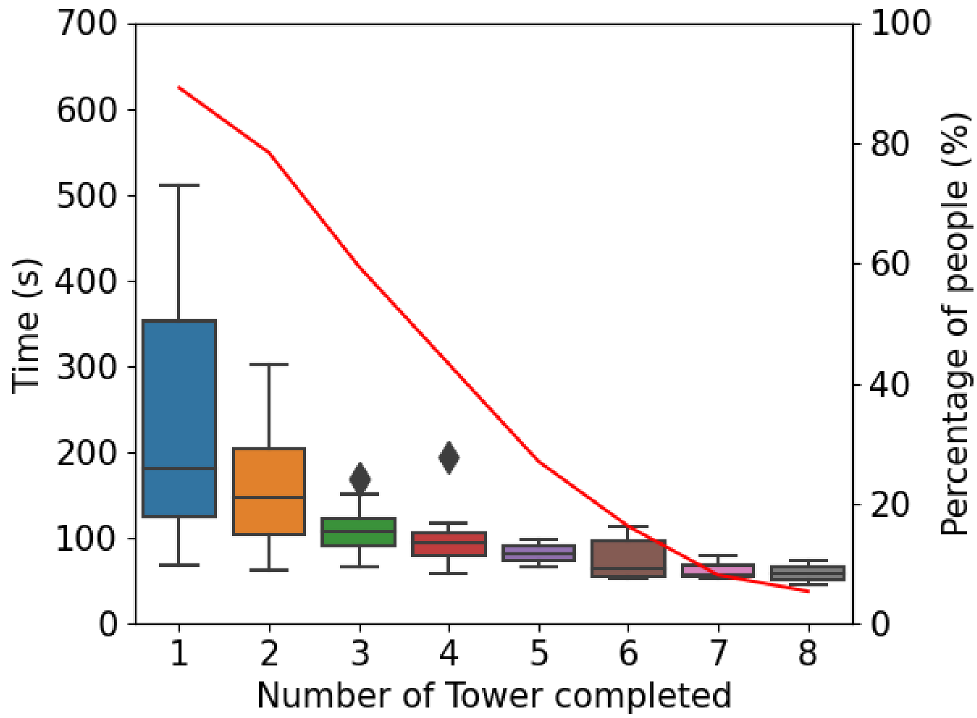


Figure 3.5: Average Time Taken and Population percentage for each tower completed for the the Red Design. The right axis corresponds to the red line, while the left corresponds to the box plot.

by users to complete the towers. In particular, the first tower for both robots took most of the task duration because some participants could not build one tower. This time completion trend shows TELESIM's user-friendliness as, from the 4th tower for the Red Design (Fig. 3.5), building one tower took less than 100 seconds on average. Similarly, for the the Blue Design, the time taken to complete each tower follows a trend similar to that of the Red Design.

Table 3.1 shows the additional statistics collected during the experiment, such as the percentage of times the user dropped a cube that caused the tower to collapse. Table 3.1a indicates that for the Red Design, 75% of the picking actions resulted in a correctly placed cube that did not collapse due to incorrect placement or the user inadvertently moved the robot in the tower's path. Similarly, in Table 3.1b, 46% of all the picking actions resulted in a correct place. The difference in the number of towers built, shown in Fig. 3.6, can be explained by a greater difficulty in placing the cube. Specifically, our results indicate that there is a significant difference in the difficulty of placing a cube ($P < 0.01$), as well as the amount of time a cube is dropped ($P < 0.01$). The null hypothesis was respectively, *There are no difference in the placing/dropping of a cube between the Red and Blue Design*. All statistical test were performed using the Welch T-Test, as it performs similarly whether or not the variance are equal, unlike the Student T-Test. More information on the distribution of the data is available in Figure 4.8 in Chapter 4. These similarities are logical as the placing and dropping rates are complementary since these are the only

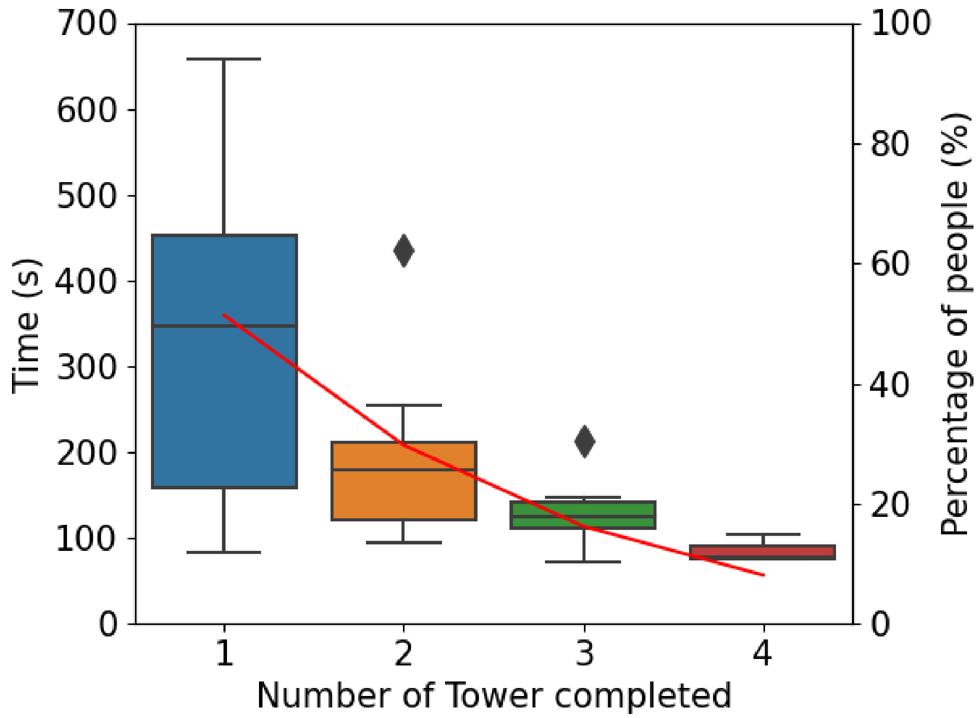


Figure 3.6: Average Time Taken and Population percentage for each tower completed for the the Blue Design. The right axis corresponds to the red line, while the left corresponds to the box plot.

two outcomes after picking up a cube. The differences can be explained by the difference in the two designs, especially the dropping rate, which is most likely linked to the weakness of the gripper, such as the limitation of the grip strength of the closed finger, to prevent the cable from breaking, and the limited range of motions since we observed that some users let the cube fall while the gripper was closed. However, successful users moved slowly to prevent unnecessary movement, thus reducing the risk of dropping. It is difficult to relate these results with the literature as previous research has not performed similar experiments. The collapse rate in both Table 3.1a and Table 3.1b is similar, and the statistical analysis shows that the difference is not significant ($P > 0.05$) and seems to indicates that the type of robot does not influence the difficulty in operating the robot around obstacles. However, more evaluation is needed to isolate the effect of the robot, UIDs and gripper type, which are currently linked.

Results of the Single Ease Question (SEQ) asked at the end of each task, in which a higher score means that TELESIM is easy to use, indicate that participants who performed better gave a higher SEQ score. Specifically, the Red Design obtained a mean of 3.32 with a standard deviation of 1.27, while the Blue Design obtained a mean of 2.19 with a standard deviation of 1.14. Additionally, the mean difference between the Red Design and the Blue Design corresponds with fewer towers built for the the Blue Design.

Table 3.1: Additional Statistics Collected

(a) Red Design

	Min	Mean ± Std	Max
Placing Rate	25.00%	77.42% ± 15.54%	100.00
Dropping Rate	3.70%	23.83% ± 14.08%	66.67
Collapse Rate	5.56%	18.44% ± 11.66%	57.14
Still in Place Rate	24.31%	75.21% ± 15.20%	95.92

(b) Blue Design

	Min	Mean ± Std	Max
Placing Rate	12.50%	46.29% ± 17.97%	86.67
Dropping Rate	13.33%	53.37% ± 18.63%	87.50
Collapse Rate	4.76%	22.25% ± 12.46%	50.00
Still in Place Rate	14.88%	46.93% ± 16.75%	84.44

3.8 Conclusion and Future Work

In this paper, we have investigated the performance of TELESIM by conducting a medium-scale user survey with 37 participants who were asked to build towers of three cubes by teleoperating robots. We tested TELESIM's modularity on two different robots with two different control modalities. Our experimental results show that TELESIM is modular, plug-and-play and user-friendly, as not only were we able to deploy it on two robots with different modalities, but most users were able to succeed by building at least once a tower of three cubes, with only five minutes of training, regardless of the control modality, robot used and hardware limitations. Our experimental study seems to indicate that the types of robot does not have any impact on the performance of teleoperation. However, due to a high level of coupling between the UID, the robot type and the gripper type, more research is needed. We thus bridged the gap pointed out by Rea & Seo [12], where they state that there is a lack of non-expert evaluation of robotic teleoperation for general tasks such as picking and placing common objects. TELESIM is available on GitHub[2], allowing developers to perform teleoperation on their robots with minimal setup time.

While TELESIM demonstrates significant potential as a modular and user-friendly framework for robotic teleoperation, this study acknowledges a critical limitation: the inherent coupling between the gripper type, robot morphology, and UIDs. Specifically, the experimental setup heavily intertwines these three factors, making it challenging to isolate their individual

[2]https://cvas-ug.github.io/telesim

contributions to user performance and workload. For instance, the SenseGlove (used with the UR3 robot) provides fine-grained control of finger movements but is paired with a less robust gripper design and a robot with limited reach. Conversely, the VR controller (used with the Baxter robot) offers simpler control but is paired with a more capable robotic platform. These combinations confound the analysis, as performance differences may stem from hardware capabilities rather than the UID itself.

To address this limitation in future work, experimental designs should decouple these variables by systematically testing each gripper type, robot morphology, and UIDs independently. For example, using a single robot platform with interchangeable grippers and UID could help isolate the effects of each component. While this study provides valuable insights into the usability of TELESIM across diverse setups, its findings should be interpreted with caution due to the intertwined nature of hardware and UIDs.

Our underlying motivation for choosing direct teleoperation in this paper is to establish a baseline for further research on shared autonomy, which could combine human intuition and a high-level overview of a task while giving freedom to the robot to perform, for example, accurate picking and placing objects. Additionally, we plan to remove the constraint of having the VR headset behind the user and allow them to wear the headset to operate either in VR in the digital twin view of Isaac Sim or Augmented Reality by allowing the user to move around the robot and have different viewpoints while manipulating, thus enhancing the precision of the teleoperation. Finally, we want to explore the relationship between previous experience with VR and robotic technology and success at completing the tasks to identify the level of competence required to teleoperate robots successfully.

Chapter 4

Breaking Down the Barriers: Investigating Non-Expert User Experiences in Robotic Teleoperation in UK and Japan

This section (Section 4 reproduces the author's version of the peer review manuscript submitted to the IEEE Transaction on Robotics Journal as Florent P Audonnet, Andrew Hamilton, Yukiyasu Domae, Ixchel G Ramirez-Alpizar, and Gerardo Aragon-Camarasa. Breaking Down the Barriers: Investigating Non-Expert User Experiences in Robotic Teleoperation in UK and Japan. arXiv preprint arXiv:2410.18727*, page 13, 2024. doi: https://doi.org/10.48550/arXiv.2410.18727. URL https://arxiv.org/abs/2410.18727 and does not violate the copyright of the publisher*

The totality of the work presented below was undertaken by myself, with my supervisor Dr Gerardo Aragon-Camarasa providing edits and corrections. Ixchel G. Ramirez-Alpizar was responsible for organising my exchange to the Advance Institute of Science and Technology in Japan and being my contact while I was working there. Andrew Hamilton is responsible for providing the senseglove and Yukiyasu Domae is responsible for some of the funding in Japan

The version of this paper presented in this thesis has been altered to reflect the changes suggested by the examining committee

4.1 Chapter Summary

This chapter details research conducted during an internship at the Advanced Institute of Science and Technology (AIST) in Tokyo. We deployed TELESIM on a different robot with a different gripper configuration, extending beyond the methodology described in Chapter 3. This deployment validated our framework's adaptability, demonstrating TELESIM's capability for deployment by non-experts with minimal preparation. Replicating our previous user survey with the new robotic system revealed significant insights. Our core findings remained consistent despite substantially improved task success rates attributed to superior hardware. Teleoperation continues to represent a mental and physical challenge for non-expert users, with hardware configurations playing a crucial role in mitigating cognitive stress. Furthermore, our trust evaluation produced unexpected results, challenging previous findings by Bartneck et al. [47] that the British users trusted robots more than the Japanese participants. In our findings, Japanese participants demonstrated notably higher trust levels toward the robotic arm than their British counterparts, challenging existing literature on cultural perceptions of robotic systems.

4.2 Abstract

Robots are being created each year with the goal of integrating them into our daily lives. As such, there is an interest in research in evaluating the trust of humans toward robots. In addition, teleoperating robotic arms can be challenging for non-experts. To reduce the strain put on the user, we created TELESIM, a modular and plug-and-play framework that enables direct teleoperation of any robotic arm using a digital twin as the interface between users and the robotic system. We evaluated our framework using a user survey with three robots and User Interface Device (UID) and recorded the user's workload and performance at completing a tower stacking task. However, an analysis of the strain on the user and their ability to trust robots was omitted. This paper addresses these omissions by presenting the additional results of our user survey of 37 participants carried out in United Kingdom. In addition, we present the results of an additional user survey, under similar conditions performed in Japan, with the goal of addressing the limitations of our previous approach, by interfacing a VR controller with a UR5e, henceforth referred to as Yellow Design. Our experimental results show that the Yellow Design has more towers built. Additionally, the Yellow Design gives the least amount of cognitive stress, while the combination of Senseglove and UR3 provides the user with the highest physical strain and causes the user to feel more frustrated. Finally, the Japanese participants seem more trusting of robots than the British participants.

4.3 Introduction

The advent of Industry 4.0 has fundamentally transformed the manufacturing landscape which has marked a transition from conventional programmable robots to sophisticated, data-driven systems [88, 89]. This paradigm shift, coupled with advancements in digital twin technology, has enabled the development of virtual replicas of smart factories to enhance operational efficiency and decision-making processes. As we progress into the era of Industry 5.0, the focus is now on fostering an ecologically and socially responsible industry that prioritizes human-centric values [18]. This evolution needs the emergence of machines that extend beyond mere digital replicas. That is, these machines have to evolve into collaborative partners that support humans in navigating complex tasks.

Central to the framework of Industry 5.0 is teleoperation systems, which facilitate remote human-machine collaboration [18]. However, extended teleoperation sessions present challenges, notably leading to physical and mental fatigue among operators [30]. Moreover, the requisite level of expertise for effective teleoperation still needs to be more adequately defined [4]. This highlights the need for further research to establish standardized protocols and training programs to enhance teleoperation practitioners' efficiency and well-being. Therefore, this paper investigates the influence of robotic hardware and participant demographics on performance metrics and user workload in the context of direct teleoperation.

Our previous work introduced TELESIM, a framework for intuitive robotic arm teleoperation [4]. While our initial study with 36 participants demonstrated that non-experts could effectively teleoperate robotic arms, it did not fully address the physical and cognitive strains experienced by users. This critical aspect warrants further investigation to ensure user safety and well-being in teleoperation scenarios [30].

To address these gaps and expand on our previous findings, this paper rigorously explores teleoperation's impact on users' mental and physical health and their relationship with robots during task performance. For this, we conducted a large-scale, international user survey across Japan and the United Kingdom, involving 74 participants from diverse backgrounds. The study used three robots with varying ranges and speeds: the Universal Robot 3, Universal Robot 5e, and Rethink Robotics Baxter. Participants performed a standardized 3-cube tower stacking task for 10 minutes, which allowed us to compare robot types and user demographics directly. Our contributions are thus the following:

1 A large-scale, cross-cultural comparison of teleoperation performance and user experience between Japan and the United Kingdom (UK), involving 74 participants from diverse backgrounds.

2 An in-depth analysis of the relationship between teleoperation system capabilities (e.g., reach, precision, User Interface Device (UID)) and teleoperation performance derived from experiments with three robotic arms: Baxter, UR3, and UR5e.

3 A comprehensive evaluation of user workload during teleoperation using the NASA-TLX questionnaire. We found that the difference in hardware impacts the mental workload and that the user's frustration is not linked to the performance of the teleoperation.

4 An investigation into the differences in trust towards robots between the Japanese and the British participants, challenging previous findings that the British users trusted robots more than the Japanese participants [47].

The rest of this paper is organized as follows. In Section 4.4, we explore the current state of the art and the research gap. Specifically, in Section 4.4.1, we review existing teleoperation frameworks and the methodologies other researchers have employed to evaluate them. Then,

Figure 4.1: Our framework TELESIM is being used to control a UR3 Robot (top-left, top-center) and a Baxter Robot (top-right) in the United Kingdom and a UR5e robot in Japan (bottom)

in Section 4.4.2, we explore ways the user's workload during teleoperation has been assessed. Finally, in Section 4.4.3, we inspect how trust has been recorded and evaluated in teleoperation. Afterwards, in Section 4.5, we present a recapitulation of the TELESIM framework and our experimental design implemented in the United Kingdom. Thus, we detail in Section 4.6 a detailed description of our experimental setup, emphasizing the distinctions between our experiments conducted in Japan and the United Kingdom. Finally, in Section 4.7, we provide a comprehensive analysis and comparison of the workload and performance of our teleoperation framework.

4.4 Background

4.4.1 Teleoperation Systems

Direct teleoperation is recognized as an essential precursor to shared autonomy [39]. This recognition arises from the considerable cognitive demands placed on users during direct teleoperation [30] and the challenges users face in executing precise, millimetre-scale adjustments to a robot's end effector. In medical applications, user movements are often scaled down to enhance precision [31, 32]. However, different scaling are not suitable for all manipulation without a way to switch between levels of scaling, as a scaling similar to medical procedure would require the user to extensive arm movement to perform object relocation.

While substantial research has been done in the field, only our previous work [4] has evaluated direct teleoperation across multiple robots without incorporating shared autonomy. Existing literature primarily focuses on shared autonomy to alleviate cognitive strain while maximizing precision. For example, researchers have explored various UIDs methodologies, such as low degree-of-freedom interfaces such as keyboards [41], joysticks [74, 21], touchscreens [42], and gamepads [75], which have shown improved control levels and reduced mental strain [41]. With the advent of virtual reality (VR) and augmented reality (AR) technologies, VR controllers [43, 11] and smartphones [76] have been investigated for their applicability in teleoperation. Some studies have also examined motion mapping of the user's body [72, 77] or gaze control [73], though these methods present challenges such as kinematic discrepancies and increased cognitive load [30]. While extensive research has been conducted, the diversity of findings allows for tailored UID that can be selected based on specific cases. The UID, therefore, remains a choice made on a case-by-case basis.

To address the limitations of direct teleoperation, researchers have emphasized the benefits of shared autonomy in enhancing task success rates. However, the absence of literature focusing solely on direct teleoperation suggests that these limitations may be attributed to the specific implementation of teleoperation in each study. Notably, researchers have employed either MoveIt! [79], which has its limitations, described in Section 4.5, or custom planning interfaces for motion generation, which is either not detailed how teleoperation was implemented [29], or whether teleoperation was developed for the specific shared autonomy framework[40, 38]. In addition, these studies have predominantly focused on single robotic systems and conducted experiments with small user groups (between 8 and 23 participants), emphasizing different autonomy levels rather than UIDs.

4.4.2 Evaluating Workload for Teleoperation Systems

Shared autonomy can potentially enhance direct teleoperation performance, but human factors significantly influence the overall success of tasks. Therefore, an efficient teleoperation system

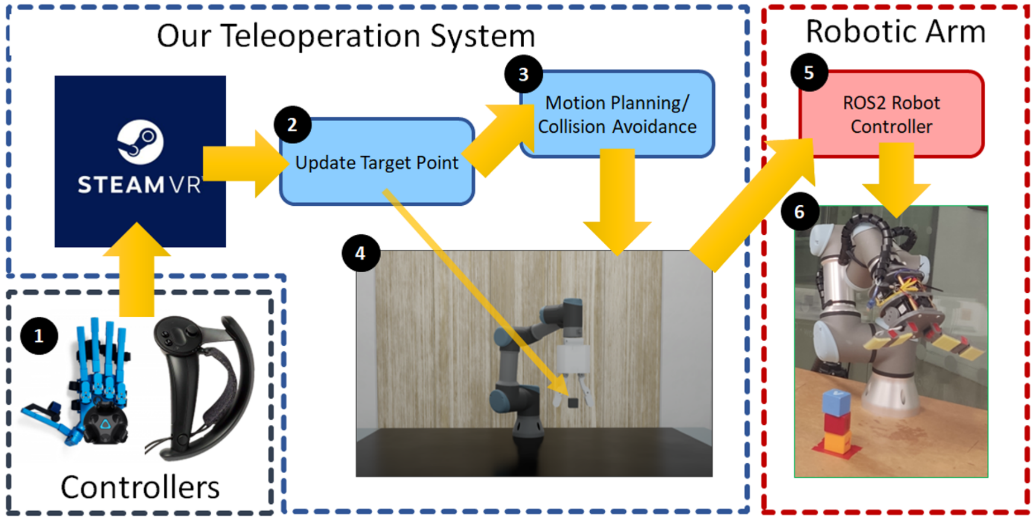


Figure 4.2: Overview of TELESIM. The UIDs (in the black dotted line) (1) can be any system that outputs a 3D pose. TELESIM is depicted in the blue dotted line, which accepts the pose given by (1) to update the 3D pose of a cube in the digital twin. The robot then calculates a path to this cube in real-time while avoiding collision with the world (4). Finally, as shown in the red dotted line, TELESIM can be plugged into any robotic system (6) via a ROS2 robot controller (5).

must minimize both physical and cognitive stress on the operator, and assessing user workload is crucial, with the NASA Task Load Index (NASA-TLX) [90] being the predominant tool utilized in the literature for this evaluation. Sandra Hart developed the NASA-TLX questionnaire at the NASA Ames Research Center in 1988 [91], a widely used subjective assessment tool designed to measure perceived workload across various tasks and environments. It evaluates workload based on six dimensions: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. Two versions of the NASA-TLX are available. The original version includes six 21-point scales for each dimension and a pairwise comparison of each dimension, whereas the "Raw" version omits the pairwise comparison [90]. The "Raw" version is often favoured because it is easier to carry out and analyse [90].

In robotic teleoperation, the NASA-TLX is the preferred instrument for assessing workload [29, 92, 38]. To our knowledge, only a few research works have opted not to use the NASA or "Raw" NASA questionnaire for robotic teleoperation. That is, some researchers instead decided to use the NASA questionnaire to measure the user's workload using extra devices or metrics. For example, Moya *et al.* [93] utilized EEG but corroborated these results with the NASA questionnaire. Additionally, researchers such as Naughton *et al.* [44] were interested solely in the overall NASA score, which they obtained by summing all dimensions. However, it is not the only way to use the NASA questionnaire. Parsa *et al.* [92] conducted an ANOVA analysis on each NASA dimension rather than using the questionnaire score to extract more value from the questionnaire. Others, such as Lin *et al.* [29], focused on specific metrics to emphasize particular results [94]. Finally, Bechtel *et al.* [95] employed a single question for users to self-report their overall workload on a Likert-Scale, similar to the Single Ease Question, which is "Overall, This task was ?"

4.4.3 Trust in Robots

Trust in robots is a critical factor, especially given the recent increase in robots designed to interact with the general population [96]. Consequently, researchers have developed various questionnaires tailored to different use cases, as evaluating trust is complex and often requires multiple scales [97]. Prominent scales include the General Attitudes Towards Robots Scale (GAToRS) [98], the Robotic Social Attributes Scale (RoSAS) [99], and the Negative Attitude Towards Robots Scale (NARS) [100]. These scales evaluate social robots with which users can interact physically or verbally or establish an anthropomorphic connection. NARS [100] remains popular due to its three subscales, which can analyse a broad range of attitudes toward robots: negative attitudes toward interaction situations with robots (S1), negative attitudes toward the social influence of robots (S2), and negative attitudes toward emotions in interaction with robots (S3). Each scale can be employed individually to examine specific cases. NARS is the only trust scale used to evaluate trust in robotic arm manipulation [101].

4.5 Teleoperation Framework

TELESIM [4] (Fig. 4.2) is a framework that enables teleoperation of various robotic arms. The framework comprises three components, with the UID (1) being any UID that outputs a 3D pose (described in Sec. 4.5.1. TELESIM, (2) in Figure 4.2 and described in Sec. 4.5.2, computes a real-time motion plan for the pose provided by the UID (1) using Nvidia Isaac Sim [48], an RTX simulator serving as a digital twin of the environment. The joint states of the robot are then sent through ROS2 to any robotic arm that supports ROS2 control (3) in Fig. 4.2.

4.5.1 User Interface Device

Our choice of UID method was informed by the work of Gottardi et al. [38], who explored integrating multiple UID, including the combination of VR controllers with tracking bands to monitor user movements, and Rakita et al. [32] who compared several UIDs and integrated them into a custom inverse kinematics solver to align the end-effector pose with the user's input. Their findings indicated a preference for VR controllers in completing pick-and-place tasks. Thus, the UIDs for TELESIM, (2) in Fig. 4.2, consisted of a Steam Index VR controller used to manipulate a Rethink Robotics Baxter robot, henceforth referred to as Red Design and the UR5e, henceforth referred to as Yellow Design and a Senseglove to manipulate the UR3, henceforth referred to as Blue Design. We used the Senseglove for mapping individual finger motions, particularly the index and thumb, to the two fingers of the Yale T42 gripper, described in Section 4.5.3. A Vive VR Tracker, mounted on top of the Senseglove, was employed to map user motions into the UR3 robot.

The Senseglove functions analogously to a glove affixed to the palm and fingertips. For our user survey, we opted for the Senseglove development kit (shown in Figure 4.2(1)) due to its ease of application and adjustment. Moreover, we limited the strapping to the index finger and thumb, corresponding to our two-fingered gripper design. Although the Senseglove SDK is compatible with multiple platforms, integration into our framework required us to port it to ROS2. This was accomplished through the creation of a ROS2 control plugin[1]. This adaptation enabled the seamless incorporation of the Senseglove functionality within our TELESIM framework. Using this plugin, we are able to read the state of the user's hand by aggregating the joint angles of each finger and scaling them to a range corresponding to the gripper finger being fully opened and fully closed. This scaling aligns with the motor range, derived from average values recorded in practice for a fully opened hand and a closed hand. This methodology enables a standardized mapping of finger positions to motor inputs, facilitating the accurate translation of user hand movements to gripper actions.

We use the Python interface for SteamVR to develop an application that continuously monitors the UID's position, subsequently transmitting this data to the rest of the framework via multiple ROS2 (Robot Operating System) [80] publishers, one for each UID. An additional publisher conveys information about button states, facilitating custom integration for potential future research by allowing modifications to their interactions with our TELESIM framework. This design also allows us to incorporate new functionalities, as shown in [7].

4.5.2 TELESIM

TELESIM's core functionality is predicated on acquiring a three-dimensional pose from the UID (Fig. 4.2) and subsequently generating a motion plan for any robotic arm. The system's architecture is built upon NVIDIA Isaac Sim and ROS2, as described below.

The robotic systems are dynamically instantiated in Isaac Sim from their Universal Robot Descriptor Files (URDF) to enhance flexibility and adaptability to any robot and make TELESIM a modular and plug-and-play framework. Isaac Sim offers a diverse array of methodologies for motion planning and collision avoidance. For instance, Rapidly Exploring Random Tree (RRT) algorithms [102], specifically Nvidia's Lula RRT implementation, offer global solutions primarily suited for static environments. Conversely, Moveit2 [50], the principal motion planning library for ROS2, utilizes the Open Motion Planning Library (OMPL) [103]. While OMPL yields global solutions, its efficacy in real-time and dynamic applications, executed through *MoveIt Servo*[2], frequently results in singularities, necessitating robot movement away from objectives for recovery. We instead chose Riemannian Motion Policy (RMP) [49], which is distinctively

[1] https://github.com/09ubberboy90/senseglove_ros2_ws

[2] https://moveit.picknik.ai/main/doc/examples/realtime_servo/-realtime_servo_tutorial.html

engineered for dynamic environments and supports real-time operation and collision avoidance.

Although RMP requires extensive fine-tuning, recent Nvidia Isaac Sim updates have significantly streamlined this process by introducing a graphical user interface application. The decision to employ this algorithm was based on its high adaptability for real-time planning in dynamic environments, which is a crucial requirement for an efficient teleoperation system. Furthermore, Nvidia provided example files for UR10 robots, which we successfully adapted for the UR3 and Baxter robotic systems.

4.5.3 Robots and End-effectors

For our experiment we used three different grippers, Baxter robot default linear gripper, depicted in Figure 4.2, used with the Baxter robot, a modified Yale T42 gripper [3], shown in Figure 4.3, used with the UR3 robot and a Robotiq 2F-140 gripper [104], shown in Figure 4.5, used with the UR5e robot. The difference in gripper was informed due to our belief that hardware played a part in teleoperation performance. These three grippers have varying heights and opening lengths. The Baxter gripper was chosen as a pure linear gripper that is fully integrated with the Baxter robot. The Yale T42 gripper is a two-finger gripper designed with individual finger motion, causing a significant difference in height between the opened and closed state. Finally, the Robotiq is a blend of a linear gripper and a finger gripper, giving it a larger opening between the two fingers at the cost of a change in height between its two states.

The Baxter gripper is controlled using the same framework as Baxter, which is a ROS2 control package, communicating with the real robot using a ROS2 to ROS1 bidirectional bridge using the joints states provided by Isaac Sim following the setup described in Section 4.5.2. This package was adapted from the ROS1 package provided by Rethink Robotics to ROS2[3] and modified to allow more control of the gripper. The change of gripper state is triggered by a custom ROS2 node listening to the state of the VR controller, as described in Section 4.5.1, and publishes a trigger to the digital clone in Isaac Sim whenever the trigger button of the VR controller is pressed. Thus, this causes the digital clone joint states to be updated and forwarded to the real robot using the ROS2 control package.

The Yale T42 gripper was also chosen due to its 3D printable components and individually controllable fingers, each utilizing two Dynamixel motors. This configuration aligned with the requirement of mapping individual finger movements to the Senseglove interface. An interface was developed to address the SDK's lack of ROS2 integration. This involved developing a ROS2 package to incorporate the SDK into a ROS2 control interface to enable seamless integration with TELESIM.

Initial testing revealed limitations in the control board's communication capabilities. The single high-speed UART channel resulted in significant latency between issuing a command and its execution. Furthermore, the intended implementation of haptic feedback from motor data to

[3]https://github.com/CentraleNantesRobotics/baxter_common_ros2

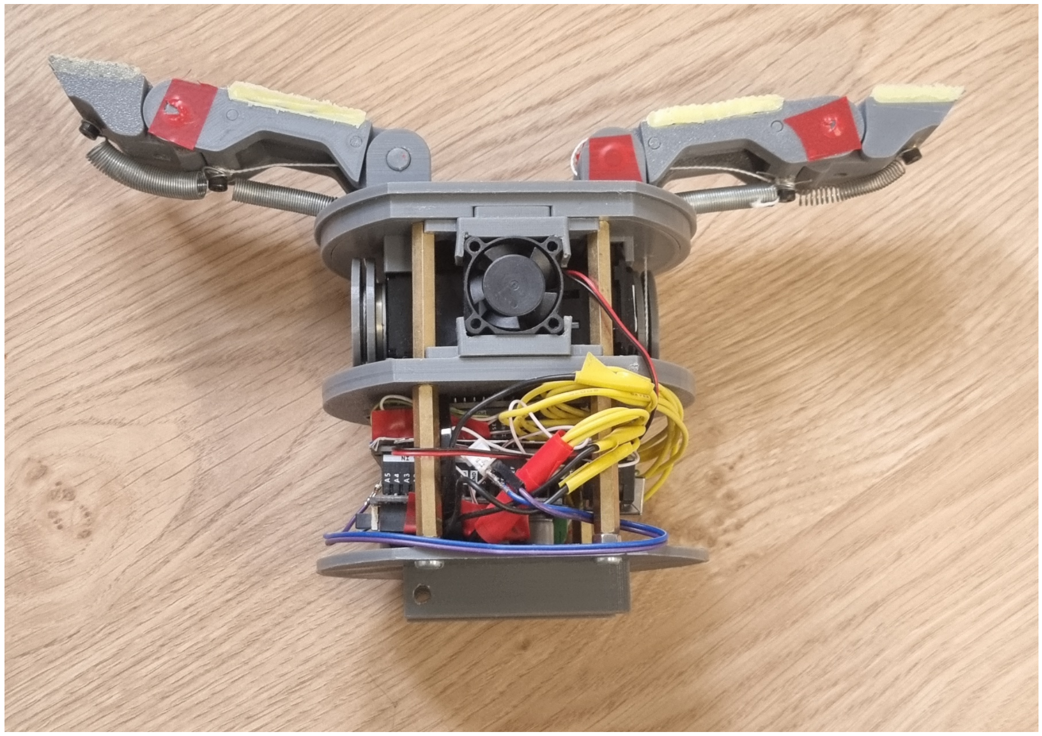


Figure 4.3: Photo of the modified T42 Gripper with the second level added for the control boards

the Senseglove required a higher data transmission rate than the virtual UART communication could support. A dual Arduino board setup employing I2C communication was implemented to address these issues, effectively resolving the bandwidth constraints. The decision was made to forgo haptic feedback implementation for this paper, as it would have been the sole UID featuring this capability, potentially complicating comparisons between UIDs in user surveys and experiments. Although vibration on the Vive VR controller could theoretically serve as a form of haptic feedback, this approach would introduce additional variables requiring evaluation, such as vibration strength and delay, to assess their impact on user performance and perception.

We had to alter the original gripper's mechanical design by incorporating two Arduino boards in the gripper. A secondary level was introduced between the motor and the robot attachment point to accommodate the control boards. This change is shown in Fig. 4.3. This solution was adopted after attempts to mount the boards on the gripper's side, resulting in cable entanglement and damage during robot operation due to spatial constraints. While the additional level resolved the cable management issues, it extended the gripper's length by 5 cm, consequently reducing the UR3 robot's already limited workspace.

The finger control mechanism uses a fishing line under tension driven by the motor. Initial tests revealed that the original fishing line lacked the necessary strength for teleoperation tasks, frequently breaking during use. Although stronger fishing lines were considered, they imposed excessive strain on the motors. To preserve motor integrity, the decision was made to retain the weaker fishing line despite its limitations. The final modification involved the removal of the final degree of freedom on each finger. This alteration was implemented as the original finger configuration hindered the grasping of cubic objects without providing significant advantages for the objectives of this paper. The modifications to the Yale T42 gripper, while addressing critical functional requirements, introduced trade-offs in terms of workspace reduction and grasping force limitations. These adaptations were necessary to meet the specific demands of the teleoperation experiment within the constraints of the available hardware used in our experiments in Sec. 4.6.

The gripper is controlled using a custom ROS2 package that listens to the joint states given by the digital clone in Isaac Sim and sends them through I2C to the first Arduino board. It republishes the states of the motor if needed to provide an optional force feedback mechanism to the Senseglove. This package runs in addition to a ROS2 control plugin provided by the Universal Robot company[4], needed to control both the UR3 and the UR5e.

Finally, the Robotiq 2F-140 gripper [104] represents an intermediate design between Baxter's linear gripper, which maintains a constant height in both open and closed states, and the T42 gripper, which exhibits an 11 cm height differential between these states. This height variation introduces an additional variable, as the user needs to estimate the appropriate hand height for successful object grasping. This estimation is simplified in the absence of height changes, as the user can visually determine whether the target object falls within the gripper's reach. Once the gripper state in the digital clone is updated after a VR controller trigger press, a custom node transmits specific commands to the gripper using serial communication.

4.6 Experimental Material and Methodology

In our experimental setup used in the United Kingdom [4], each robot was positioned in front of a table with cubes arranged in an isosceles triangular pattern, as illustrated in Figure 4.4. The experimental protocol required participants to teleoperate a robot while standing with their backs to the VR headset, which served as both the world's frame and a reference point for the user. The user was tasked with moving and staking the three cubes from their starting position to a central position marked with red tape, as visible in Figure 4.1. The bottom three images show the evolution of the manipulation from one cube in motion to the final completed tower in the bottom right.

Participants were required to complete multiple questionnaires throughout the experiment. Initially, they completed a brief demographics questionnaire, which included age, gender, and experience with virtual reality, robots, and wearable gaming devices (e.g., Wiimote). These questions utilized a five-point Likert scale to assess each participant's perception and understanding. After teleoperating a robot, users answered two additional questionnaires. The first was the Single Ease Question (SEQ) [86], regarded as an effective end-of-task metric. The second was a raw NASA-TLX, described in Section 4.4, employing a seven-point Likert scale

[4]https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver

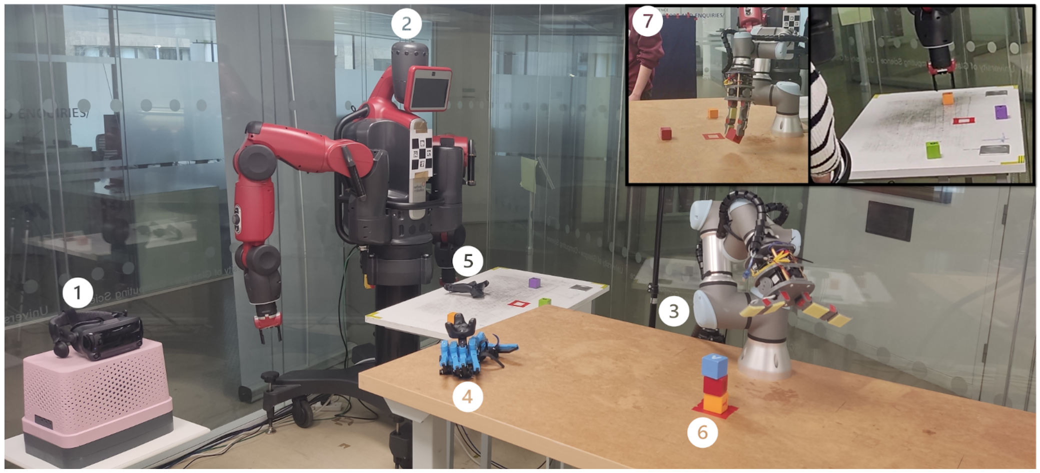


Figure 4.4: Overview of the experimental setup. The Steam Index VR Headset [2] is marked as (1) on the far left, which acts as the world's origin. The Baxter robot on the left (2) is controlled by the Steam Index VR controller (5). In front of it, the UR3 is on the right (3), with the Yale OpenHand T42 gripper [3], controlled by the Senseglove and HTC Vive tracker (4) on the left side of the brown table. Additionally, in the upper right corner (7), a view of the starting setup of the task, which consists of three cubes in a triangle pattern (described in Section 4.6.1), while on the brown table, the cubes are arranged in the goal configuration (6).

instead of the 21-point scale, as research has shown that the 21-point scale does not enhance questionnaire reliability [105, 106]. This research was validated by the University of Glasgow Ethics Committee (Application Number *300220026*).

At the end of the teleoperation experiment, participants completed the Negative Attitude Towards Robots (NARS) questionnaire, described in Section 4.4. This study exclusively employed elements from the S1 subscale, which assesses negative attitudes towards situations involving robot interactions (cf. Section 4.4). The other subscales were deemed irrelevant as they addressed hypothetical future scenarios with robots or general interactions and conversations with robots. It is worth noting that the use of a partial NARS survey is not unprecedented, as Naneva *et al.* [107] reported that the NARS-S2 scale was the most widely used in their systematic review of attitudes. This approach also enables direct comparison with the findings of Bartneck *et al.* [47], who utilized NARS to assess trust across seven different countries and are, as far as we are aware, the only researchers to have compared the Trust towards robots for teleoperation of the robotic arm across countries.

To explore the impact of the capabilities and limitations of the robotic hardware used for teleoperation, as well as the impact of demographics and previous experience, we deployed TELESIM on three different robots: the Rethink Robotics Baxter robot, the UR3 and the UR5e. Baxter and UR3 were used to conduct our experiments at the University of Glasgow in the UK. At the same time, the Universal Robot 5e was at the National Institute of Advanced Industrial Science and Technology (AIST) Waterfront Center in Tokyo, Japan. This section explores the difference in setup and environment from the user survey developed in the UK. Due to time constraints, we were required to recruit our participants through an agency. Thus, we had access to a more significant reach than in the UK, reaching participants who were not involved in academia (i.e. undergraduate and postgraduate students and academics). Additionally, participants were financially compensated for participating in the user survey, unlike in the UK, where they participated for no monetary rewards. In addition, this research was internally approved by the Ergonomics Experiment Committee of the Life Sciences Experiment Management Office in the Environmental Safety Department of AIST with the following application number: *2023-1384*. To keep our analysis consistent, we recruited participants within the same age range as described below:

- Japan–age range of 25.32 ± 6.26; and 28 male and 9 female;
- The United Kingdom–range age of 27.81 ± 7.93 and 29 male and 8 female.

4.6.1 Robotic Setups in the UK and Japan

A comprehensive analysis of their strengths and weaknesses is necessary to help us better understand the three robots employed in this study. Table 4.1 summarizes the robots' capabilities.

Baxter exhibits the longest reach at 1210 mm, followed by the UR5e with 850 mm and the UR3 with 350 mm. However, Baxter's extended reach introduces end effector position imprecision [108], a characteristic not observed in the Universal Robots. This imprecision is denoted as "Not Precise" in Table 4.1.

The primary distinctions between the two Universal robots lie in the UR5e's extended reach, gripper configuration, and UIDs. The UR3 was equipped with a modified Yale T42 gripper [3], measuring 21 cm when fully closed. Due to its size relative to the robot's body, restrictive joint limits were implemented to prevent gripper-body collisions, consequently limiting the robot's operational area. Conversely, the UR5e utilized a Robotiq 2F-140 gripper, measuring 23.3 cm when closed. For consistency, identical joint limits were applied to the UR5e; however, its longer reach mitigated the impact of these restrictions.

A notable difference between the grippers is the height differential between their open and closed states, as mentioned in Section 4.5.3. The Robotiq gripper exhibits a 2.35 cm difference, while the T42 demonstrates an 11 cm difference. This disparity increased users' difficulty in estimating the appropriate gripper height for cube grasping with the UR3, as explained in Section 4.6.1.

The T42 gripper's design introduced additional grasping challenges. As reported in TELESIM [4], the Yale gripper's limited closing force occasionally resulted in cube slippage during transit. This limitation was not observed with the Robotiq gripper. Lastly, the robots' control methods differed. The Baxter robot was operated using a Steam Index VR controller (Red Design), as depicted in Figure 4.4 (5). In contrast, the UR3 robot was controlled via a Senseglove development kit, enabling the mapping of individual finger movements, with an HTC Vive Tracker

Table 4.1: Robots Specifications

Robot Type	UID	Gripper	Precise	Reach (mm)
Baxter	VR Controller	Rethink Gripper	No	1210
UR3	Senseglove	Modified Yale T42 Gripper	Yes	350
UR5e	VR Controller	Robotiq 2F-185	Yes	850

mounted atop the hand (4) (Blue Design). The control of a modified T42 gripper from the Yale OpenHand project [3], attached to the UR3, was limited to the user's thumb and index finger movements (3) in Figure 4.4.

For our Japan experiments, the TELESIM framework underwent minimal modifications, primarily involving the URDF file, which now loads a UR5e robot and replaces the T42 Gripper with a Robotiq 2F-140 gripper [104] (Yellow Design). This change was required given the significant design flaws of the T42 gripper, as discussed in Section 4.5.3. The Digital Twin was updated to reflect the new location and incorporate revised virtual safety settings, constraining the motion planner from approaching certain areas due to safety considerations. Owing to the compatibility between Universal Robots 3 and 5e, we were able to utilize the same ROS2 controller developed during the initial TELESIM implementation.

Consistent with our previous UR3 experiment, the gripper's collision parameters were configured based on its closed state to prevent hardware damage. To maintain analytical consistency, the joint position and speed limits applied to the UR5e were identical to those used in the UR3 experiment despite the UR5e's extended reach. Due to the extended reach, our previous cube placement was no longer suitable, as the robot was able to reach all the locations on our available table. Nevertheless, the experimental design aimed to preserve task difficulty by positioning cubes in a manner that required participants to adjust their body position to reach all cubes. Depending on the target cube, this typically involves a single step in various directions. The cube arrangement and the step required to access the leftmost cube are depicted in Figure 4.5.

4.7 Evaluation

4.7.1 Experimental Results

Figure 4.6 and Figure 4.7 illustrate that all participants using the Yellow Design successfully constructed a minimum of three towers within the 10-minute timeframe, with an median construction time of less than 100 seconds per tower. Moreover, subsequent towers were completed in progressively shorter duration. One participant notably constructed 25 towers, averaging 40

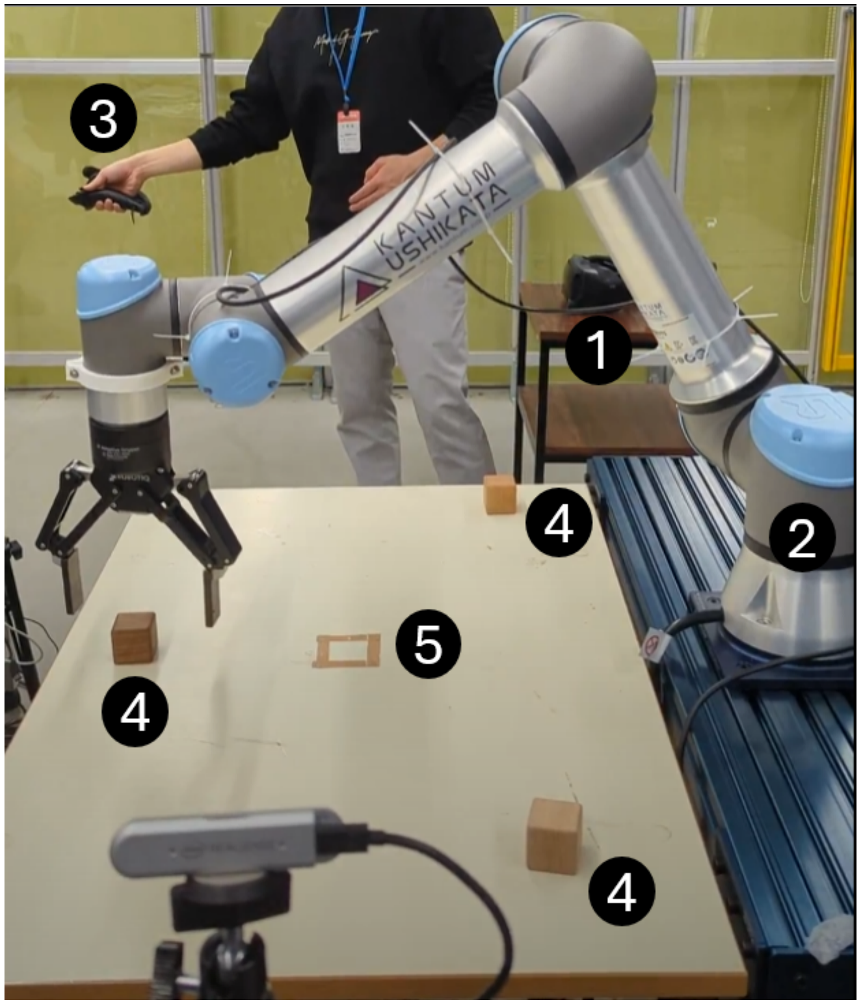


Figure 4.5: Overview of our experimental setup in AIST. The Steam VR headset (1) acts as the world's frame, as seen behind the robot and the user. The UR5e robot (2) in front is controlled by a using Steam Index VR controller (3). There are three cubes set up in their starting position (4) in a triangular pattern similar to [4]. The empty square in the middle of the table (5) represents the location where the user should stack all the cubes, resulting in a tower of three cubes.

seconds per tower. The mean number of towers constructed across all Yellow Design users was 10.50.

In comparison, the average number of towers constructed using the Red Design was 3.25, while the Blue Design yielded an average of 1.03 towers. This performance disparity is shown in Figure 4.6. Furthermore, for both the Red and Blue Design, only participants who constructed four or more towers achieved completion times under 100 seconds, as shown in Figure 4.9.

Figure 4.6 depicts a steep performance improvement for the Yellow Design up to the 18th tower, followed by a plateau for the remaining seven towers. This plateau is attributed to the fact that only two participants managed to build 21 and 25 towers, respectively. The exceptional performance of these participants suggests prior experience with robots or robotic teleoperation. However, the current questionnaire design, being self-reported, may not accurately capture this experience due to potential over- or under-estimation of skills by participants.

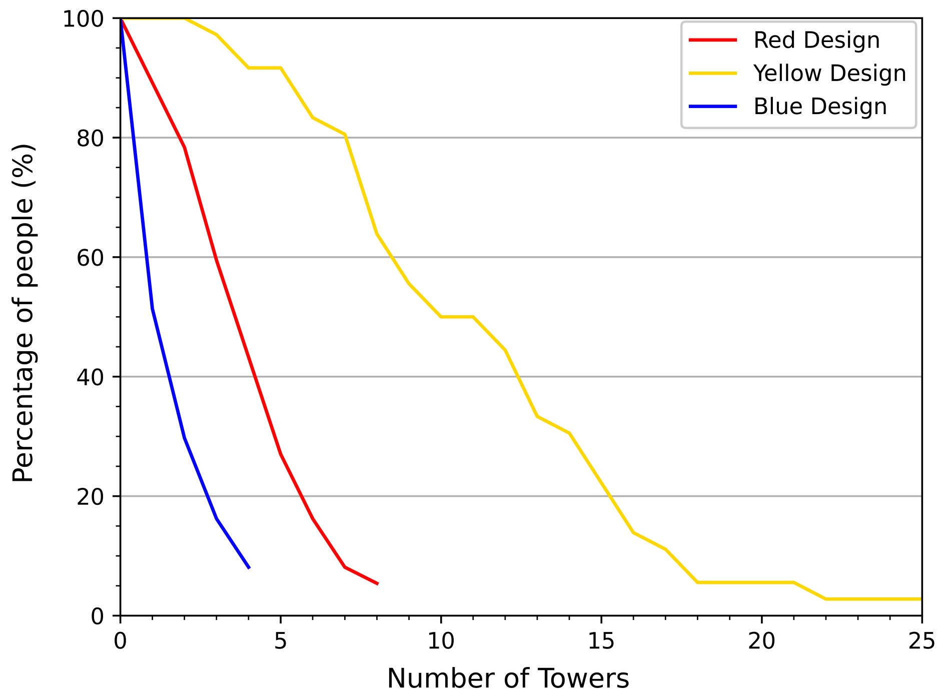


Figure 4.6: Population percentage for each tower for the Red Design in blue, the the Blue Design in orange, and the the Yellow Design in green

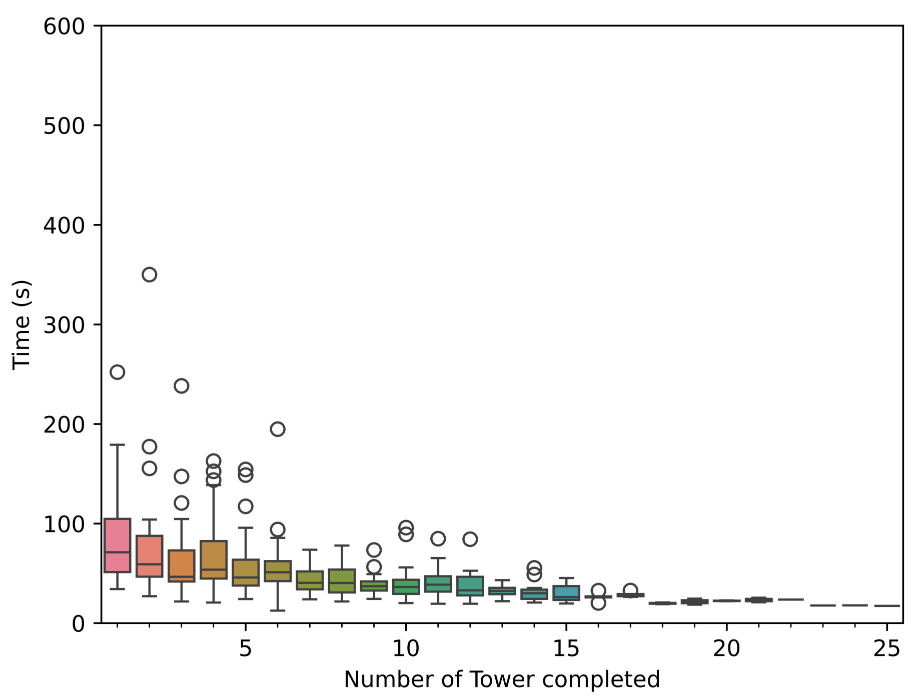


Figure 4.7: Average Time Taken for each tower completed for the the Yellow Design robot.

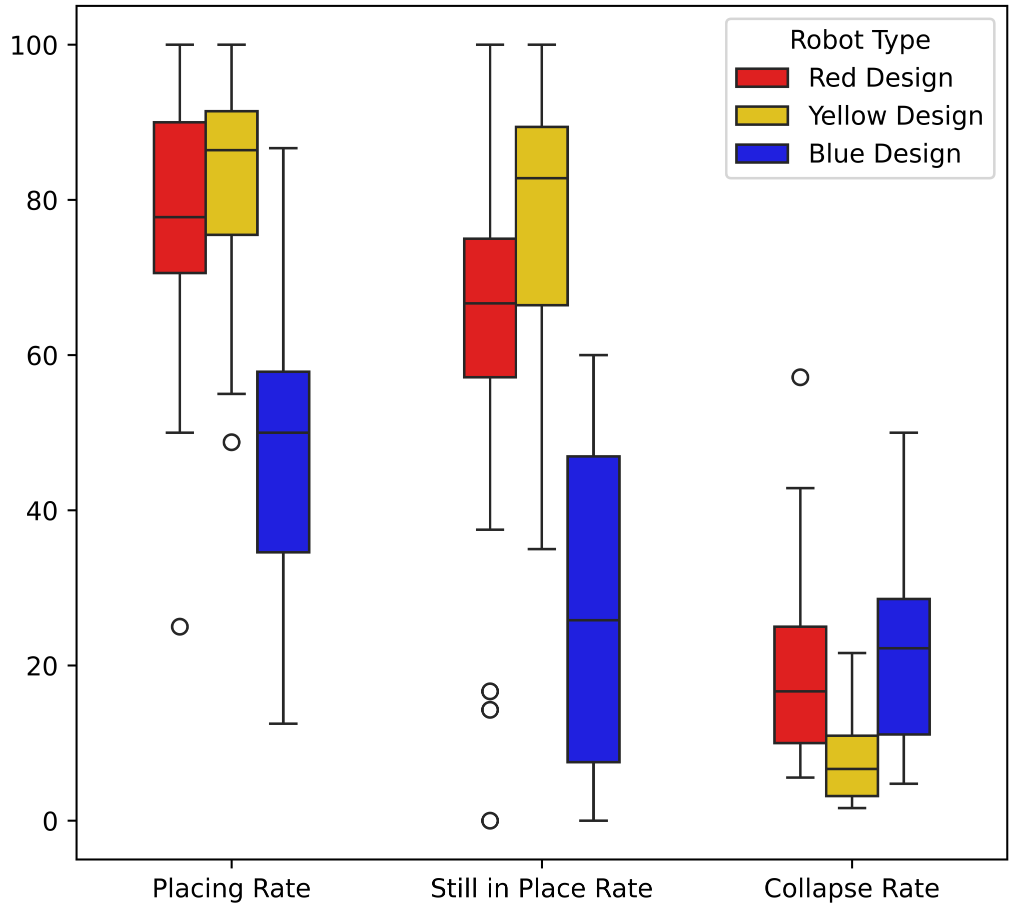
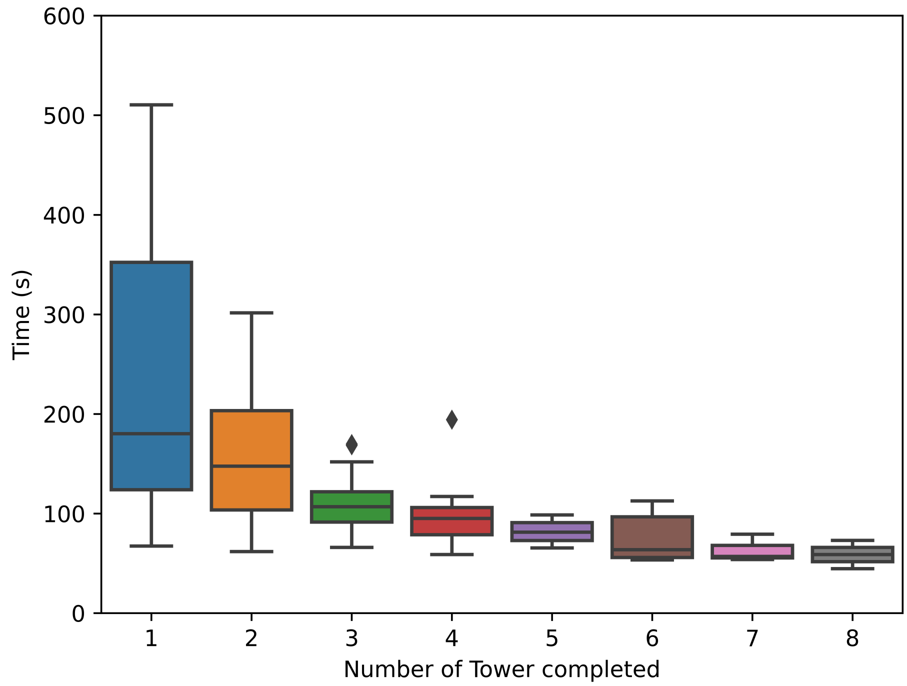


Figure 4.8: Ratio of different statistics collected during the experiment. The Placing Rate is calculated as the number of place actions over the number of picking actions. The Collapse Rate is calculated as the number of Collapse actions over the number of picking actions. The Still in Place Rate is calculated as the number of Place actions minus the number of collapses over the number of picking actions, effectively rating the tower's stability.

An informal discussion with the participant who constructed 25 towers revealed professional experience with robots. This background appears to have provided spatial skills that are directly applicable to robot teleoperation tasks.

Additional statistics were collected to investigate the relationship between teleoperation performance and robot capabilities, as shown in Figure 4.8. It is important to note that the teleoperation user surveys for the Red and Blue Design were conducted in the UK, while those for the Yellow Design were performed in Japan. The differences in experimental setup are detailed in Section 4.6. Furthermore, all of our statistical tests were performed using the Welsh T-Test and using the null hypothesis that the effect being studied does not exist (eg. There are no difference between the placing rate of the Yellow and Red Design).

The *Placing Rate*, defined as the ratio of successfully placed cubes to the total number of cube picks, aligns with the tower construction results. The Yellow Design demonstrated the highest values, followed by the Red and Blue Design. Interestingly, the difference between the



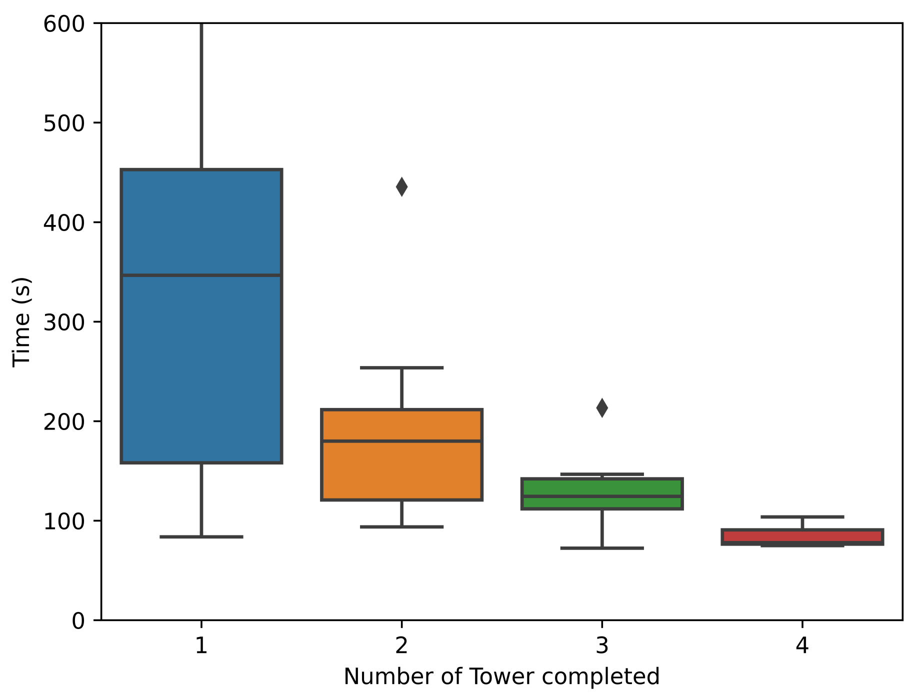


Figure 4.9: Average Time Taken for each tower completed for the the Red Design (top) and the the Blue Design (bottom).

Red and Yellow Design was not statistically significant ($P > 0.1$). However, the Blue Design exhibited a statistically significant difference ($P < 0.001$) compared to both the Red and Yellow Design. This disparity can be attributed to the advantages and disadvantages outlined in Section 4.6 and summarized in Table 4.1, with gripper design and UIDs being the primary differentiating factors.

The *Still in Place* rate, representing the frequency of cubes remaining in their correct position without collapse, revealed a statistically significant advantage for the Yellow Design over the Red Design ($P < 0.01$). This allows rejecting the null hypothesis. This conclusion is further supported by the difference in collapse rates, with the Yellow Design showing a statistically significant difference compared to the Red Design ($P < 0.1$).

The the Blue and Yellow Design exhibited statistically different collapse rates, with the Yellow Design performing better than the Red and Blue Design. While the Yellow Design outperformed both robots, its advantage over the Red Design primarily stemmed from better precision. Conversely, the Blue Design's higher collapse rate can be attributed to other factors, particularly its UID, as noted in Section 4.6.

Interestingly, the results of the Single Ease Question (SEQ) were statistically indistinguishable for the Red and Yellow Design ($P > 0.01$) despite the disparity in tower construction performance. The Red Design achieved a mean SEQ score of 3.32 (SD = 1.27), the Blue Design scored 2.19 (SD = 1.14), and the Yellow Designscored 3.72 (SD = 1.54). Higher scores indicate greater perceived accessibility. We hypothesize that the Yellow Design enhanced precision may have led users to perceive a higher tower construction potential than the Red Design. Notably, participants were not informed of the maximum achievable tower count before completing the questionnaire. It appears that users estimated the theoretical maximum based on their experience, resulting in comparable SEQ scores for both designs despite performance differences. As expected, there is a clear relationship between the number of towers and a higher SEQ score, as can be seen in Figure 4.10

Our analysis primarily focused on the full system capabilities. However, it is important to note that the Yellow Design experiment was conducted in Japan, and we hypothesize that demographic factors may have influenced the results. The decision to conduct this user survey in Japan was based on research by Nam et al. [109], which suggests that user workload varies depending on the level of trust in a system. Bartneck et al. [47] indicate that participants from different countries exhibit varying levels of trust toward robots.

The Japanese participants self-reported higher levels of experience with robots (mean 3.86 ± 1.27) compared to the British participants (mean 2.76 ± 1.24). This difference is statistically significant ($P < 0.001$) and can be attributed to the greater exposure of the Japanese participants to robots in their daily lives. Interestingly, the Japanese participants reported less experience with wearable technology (mean 2.81 ± 1.47) compared to the British participants (3.86 ± 0.87), a difference that is also statistically significant ($P < 0.01$). However, virtual reality (VR) ex-

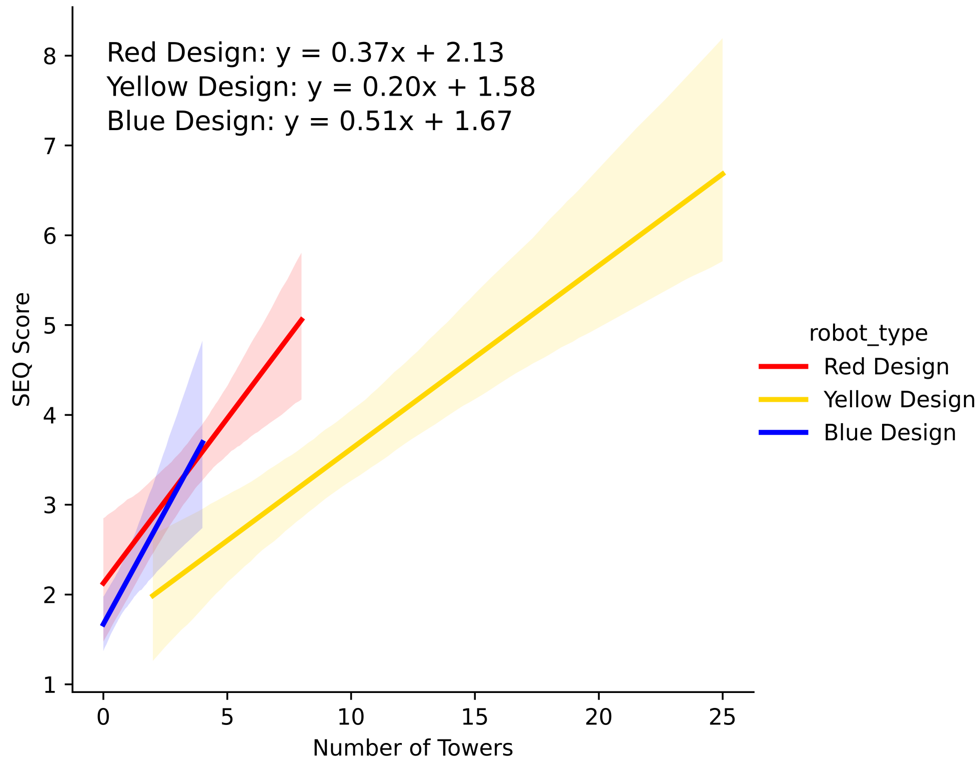


Figure 4.10: Regression plot showing the relationship between SEQ and the number of towers for each individual robot.

perience was statistically similar between the Japanese participants (mean 3.25 ± 1.38) and the British participants (mean 2.97 ± 1.20) participants. These findings suggest that previous experience with robots and wearables varies depending on the user's background. In contrast, VR experience remains consistent across both demographics, possibly due to the recent widespread adoption of VR technology [110].

It is important to note that we are unable to directly correlate previous experience with robots or wearables to teleoperation performance due to the differences in the systems users operate. However, we were able to exclude previous VR experience as a contributing factor to performance differences, due to a lack of statistical significance.

4.7.2 Operator Workload

Figure 4.11 illustrates the individual results for each subscale or dimension of the NASA-TLX survey. The subscales are described as follows:

- Mental: This subscale examines the mental workload imposed on the user during task performance. The Yellow Design demonstrates the lowest mental workload, statistically significant at the 95% level compared to the Red and Blue Design, which show similar results. Interestingly, the UID difference between the Red Design (VR controller) and the Blue Design (Sensglove) does not impact mental workload, nor does the Blue Design's weaker gripper. The Yellow Design's precision and range likely explain this difference, providing users with greater freedom of movement without concern for robotic constraints.

- Physical: This subscale assesses the user's physical workload during task execution. The Blue Design exhibits a statistically significant difference compared to the other two robots, attributable to its distinct UID. The Blue Design, controlled by the Senseglove, requires users to maintain a flat, extended hand position, unlike the VR controller, which allows for a more relaxed hand posture closer to the body.

- Pace: This subscale examines the user's perception of time pressure. The results are statistically consistent across all robots, which is expected since all experiments were conducted under uniform time constraints.

- Performance: This subscale examines the user's perceived task performance. The Blue Design demonstrates the lowest perceived performance, statistically significant at the 95% level compared to the other two robots. This aligns with the Single Ease Question results described in Section 4.7.1, as both metrics evaluate task difficulty.

- Effort: This subscale investigates the perceived effort spent by the user to complete the task. The Blue Design shows the highest effort expenditure, which is statistically significant at 95% compared to the Red and Yellow Design. The consistently high effort scores across all robots corroborate the assertion that robot teleoperation is challenging for non-experts, as noted in previous research [12, 13].

- Frustration: This subscale explores the user's frustration and stress levels during the experiment. The Blue Design leads with a statistically significant difference over the Red and Yellow Design, likely due to the disadvantages outlined in Section 4.6, particularly its limited range and weak gripper. Notably, this is the only subscale where participants utilized the full range of possible answers for all robots. This suggests that frustration tolerance is highly individual and may not be an ideal metric for evaluating teleoperation system performance.

The above NASA-TLX results indicate a relationship between UID type and user impact, particularly in physical and emotional domains, manifesting as frustration and stress. Additionally, robot type influences user mental workload, with the Yellow Design's combination of large reach and high precision resulting in reduced mental strain compared to robots possessing only one of these qualities, such as the Red and Blue Design.

The hypothesis that demographics play a significant role can be discarded, as additional statistics such as *collapse rate* and *still in place rate* are similar for both the Red and Yellow

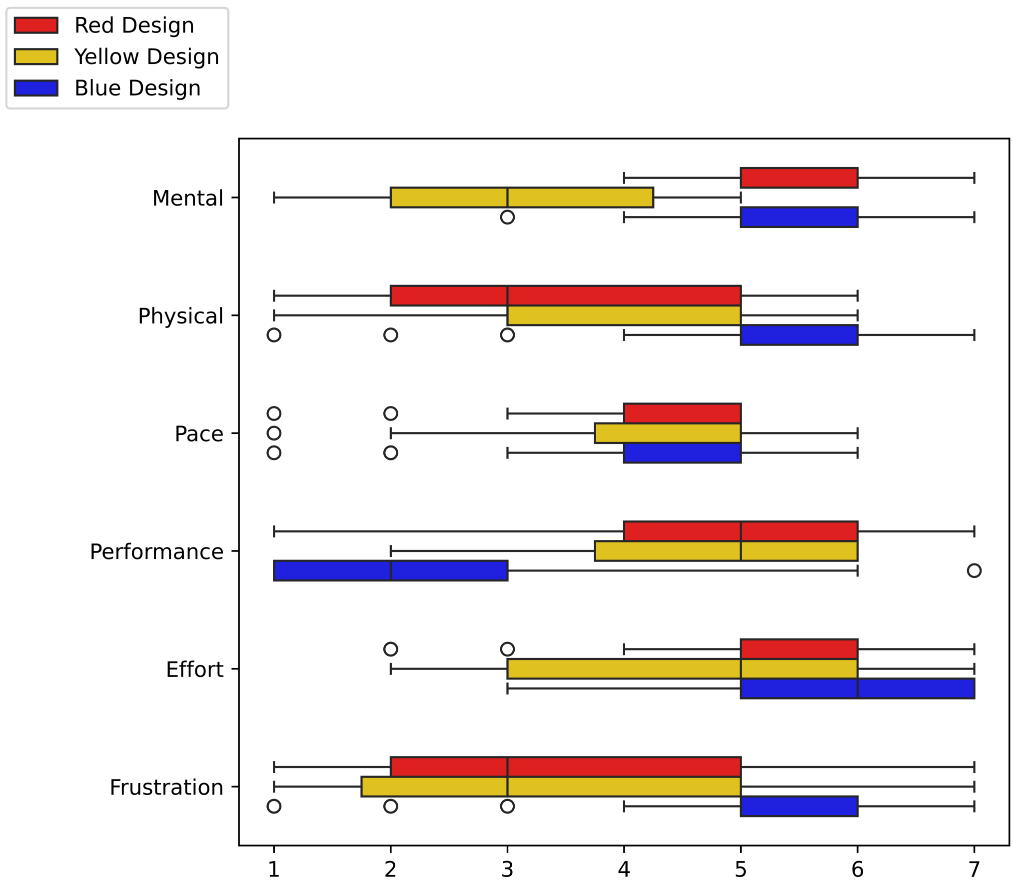


Figure 4.11: Boxplot representing individual results from the NASA task survey, in which a lower score indicates lower difficulty. Blue represents the experiment carried out with the Yellow Design in Japan. Orange represents the experiment with the Blue Design in the UK. Green represents the experiment with the Red Design in the UK.

Design (cf. Figure 4.8). This non-significance is further supported by the similarity across all NASA-TLX subscales, except for mental workload. However, to conclusively prove that demographics does not play any role, the same experiments would need to be carried out in both country (eg. the Yellow desing in the UK). Finally, these results suggest a trend that, regardless of the robot type, users experience frustration and some degree of physical strain. We postulate that this contributes to the difficulty of teleoperation for non-experts, as noted in previous research [12, 13].

4.7.3 Trust Towards Robots

Our NARS questionnaire, which evaluates trust, focused solely on elements from the S1 subscales, as described in Section 4.5. Figure 4.12 illustrates the individual and combined scores for each question of the S1 subscale. The Red and Blue Design results are combined, as the questionnaire was administered only once. In retrospect, administering the questionnaire twice—once before the experiments and again after manipulating the first robot—might have provided insights into the progression and impact on trust for each robot.

For completeness, the questions used in our questionnaire, as found in [100], are:

NARS Q.1 I would feel uneasy if I was given a job where I had to use robots

NARS Q.2 I would feel nervous operating a robot in front of other people

NARS Q.3 I would feel very nervous just standing in front of a robot.

NARS Q.4 The word "robot" means nothing to me.

NARS Q.5 I would hate the idea that robots or artificial intelligence were making judgments about things.

Figure 4.12 reveals that the Japanese users demonstrated higher trust in robots for questions **NARS Q.1**, **NARS Q.2**, and **NARS Q.5**. However, this difference is statistically significant only for **NARS Q.5** ($P<0.05$). This general trend results in a statistically significant difference in total scores between the two demographics ($P<0.05$). This disparity may be attributed to the longer exposure of the Japanese participants to robots in their daily lives, as noted by Bartneck et al. [47]. Additionally, Japanese culture's lesser distinction between natural and artificial entities, influenced by Buddhist beliefs that do not discriminate against spirits in machines (unlike Christianity), may play a role [111, 47]. Media portrayal of robots may also contribute, with Western media often depicting robots as antagonists. At the same time, Japanese manga presents a more nuanced view where robots can assist in combating human-originated evil [47]. The significant difference in **NARS Q.5** particularly supports this notion, as the concept of robots making decisions often leads to their rebellion in Western media [112].

Interestingly, our results appear to contradict the conclusion drawn by Bartneck et al. [47] that the Japanese participants are less trusting than the British participants. We hypothesize that this discrepancy may be due to the advancement and proliferation of robots and AI in daily life since the publication of that research work.

Finally, it is noteworthy that this difference in trust does not appear to affect teleoperation performance, as the Japanese participants achieved the highest number of towers built. This outcome is primarily attributed to hardware differences, given that previous experience with VR-related technology is similar between the Japanese and the British participants, as explained in Section 4.7.1.

4.8 Conclusion and Future Work

This paper presents an extensive exploration of teleoperation's impact on users through three user surveys, encompassing 74 participants across two countries, utilizing three different robots and two UIDs. The study employed NASA-TLX questionnaires to evaluate workload, NARS

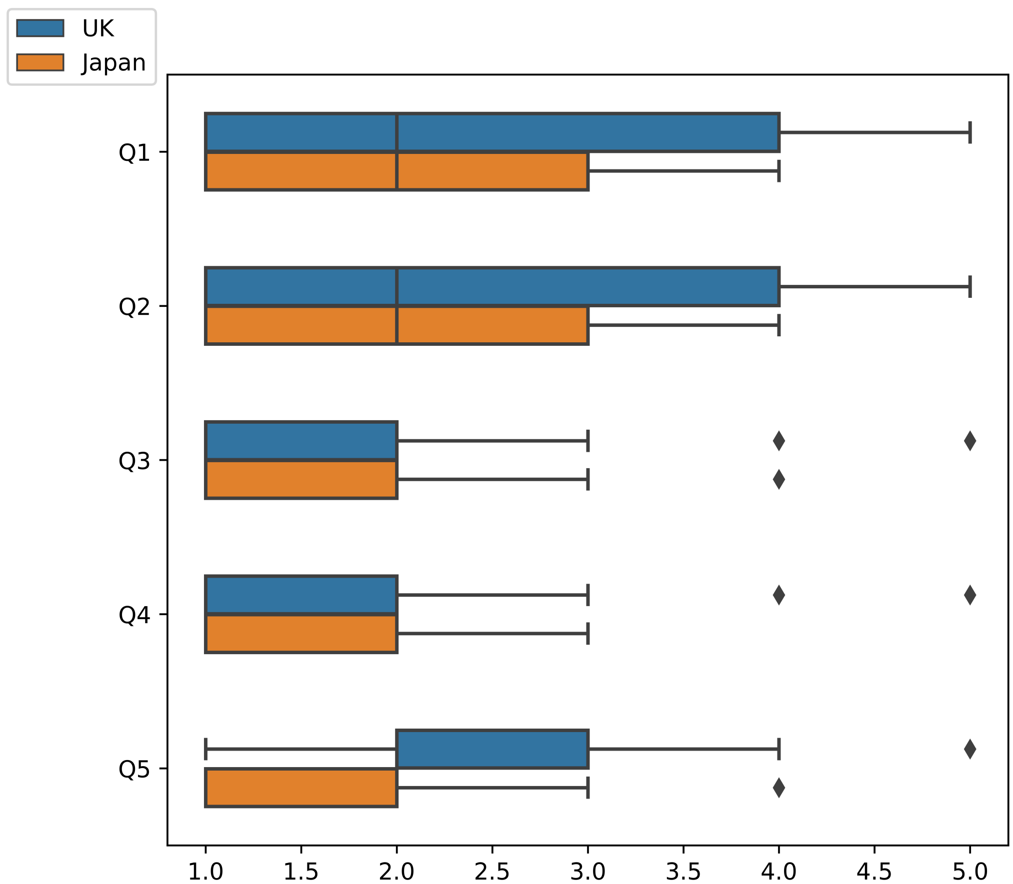


Figure 4.12: Boxplot representing the total score for the NARS (Negative Attitude Towards Robot) where a low score indicates the user is more trusting of the robot. Blue represents the experiment done in the UK. Orange represents the experiment done in Japan.

questionnaires to assess trust toward robots, and SEQ to gauge task difficulty. Additionally, the research investigated trust towards robots across different demographics.

Experimental results demonstrate that participants achieved the highest number of towers built using the Yellow Design. This superior performance is attributed to the UR5e's enhanced precision and reach, allowing for better cube placement and reduced tower collapse during continued teleoperation. The Yellow Design also induced the least cognitive stress, while the Blue Design resulted in the highest physical strain and user frustration (cf. Figure 4.11).

The findings in this paper corroborate previous research [12, 13] indicating that teleoperation is challenging for non-experts, characterized by high frustration and physical stress across all robot types. As anticipated, the Japanese participants reported more VR experience than the British participants. Conversely, the Japanese participants reported less experience with wearable technology. The study demonstrated that prior VR experience does not significantly impact teleoperation performance, regardless of the operator's background.

Interestingly, the Japanese participants exhibited higher trust towards robots compared to the British participants, contradicting the conclusions drawn by Bartneck et al. [47]. These user surveys also validated the TELESIM framework by enabling international-scale teleoperation across multiple robotic platforms. In addition, the consistent results across demographics proved the user-friendliness of the TELESIM framework, along with its plug-and-play capability, as shown by its deployment on multiple robotic interfaces and end effectors.

While this paper provides an in-depth analysis of teleoperation's impact on users, it focuses solely on direct teleoperation with visual feedback. Other teleoperation methods mentioned in the literature, such as VR headset usage or telepresence robot control [11], were not explored. Moreover, additional comparison with the state of the art is difficult due to the disparity in the teleoperation task and UIDs. While the cross-cultural study conducted in Chapter 4 provides valuable insights into teleoperation performance and user experience, several limitations must be acknowledged. A key issue lies in the inherent coupling between the gripper type, robot morphology, and user interface device (UID) configurations. For instance, the UR5e robot paired with the VR controller (Yellow Design) exhibited superior task success rates, but this performance cannot be conclusively attributed to any single factor. The UR5e's extended reach and higher precision may have contributed significantly to the observed results, overshadowing the potential impact of the VR controller itself. Similarly, the UR3 robot paired with the Sense-Glove (Blue Design) demonstrated higher physical strain and frustration among users, which could stem from limitations in the gripper's design or the restricted degrees of freedom in its control.

This interdependence complicates the interpretation of findings, as variations in user performance might reflect hardware capabilities rather than the intrinsic usability of specific UIDs or control methods. To address this limitation, future studies should decouple these variables by systematically testing each gripper type, robot morphology, and UID independently. For example, using a single robotic platform with interchangeable grippers and control devices would enable more controlled comparisons. Additionally, standardized experimental protocols that isolate hardware effects from user interaction patterns could provide more definitive conclusions about the usability and effectiveness of teleoperation systems across diverse setups.

Despite these limitations, the study successfully highlights critical factors influencing teleoperation performance and cultural differences in trust toward robots. However, caution must be exercised when generalizing these findings due to the intertwined nature of hardware configurations and user interface modalities.

In future work, we aim to use these systems and investigate their impact on teleoperation success and user experience, following the methodology presented in this paper. However, our work on direct teleoperation is not completed. While our experiments seem to indicate that demographics do not play a part in the performance of teleoperation systems, we would need a wider breadth of demographics to show this conclusively. Furthermore, the many differences in hardware during our experiments make it difficult to accurately pinpoint the factors that impact the most, either the teleoperation performance or the user's workload. A deeper investigation of the Senseglove and T42 gripper's performance on the UR5 would enable a more direct comparison with the UR5e and further substantiate our hypothesis that hardware correlates with enhanced performance. This analysis would help elucidate whether the UR3's suboptimal performance stems from gripper limitations, UIDs, or its restricted operational range.

Finally, the exact skill set required to perform teleoperation with minimal training beforehand remains an unsolved problem. The study successfully eliminated prior virtual reality experience as a confounding variable. However, a participant from Japan, who possesses expertise in motion retargeting for robotic teleoperation, demonstrated superior performance by completing the highest number of towers. This observation suggests that spatial abilities or other complex skills may influence task performance.

Chapter 5

IMMERTWIN: A Mixed Reality Framework for Enhanced Robotic Arm Teleoperation

This section (Section 5 reproduces the author's version of the peer review manuscript submitted at the 2025 IEEE International Conference on Robotics and Automation (ICRA) as Florent P. Audonnet, Ixchel G. Ramirez-Alpizar, and Gerardo Aragon-Camarasa. IMMERTWIN: A Mixed Reality Framework for Enhanced Robotic Arm Teleoperation, September 2024. URL http://arxiv.org/abs/2409.08964. arXiv:2409.08964 and does not violate the copyright of the publisher

The totality of the work presented below was undertaken by myself, with my supervisor Dr Gerardo Aragon-Camarasa providing edits and corrections. Ixchel G. Ramirez-Alpizar was responsible for organising my exchange to the Advance Institute of Science and Technology in Japan and being my contact while I was working there

The version of this paper presented in this thesis has been altered to reflect the changes suggested by the examining committee

5.1 Chapter Summary

This chapter explores user immersion within a digital twin environment, examining how immersion influences task performance while assessing trust levels and operator workload. Our system replaced direct real-world observation with a virtual robot representation and near real-time workspace visualization, as demonstrated in Figure 5.1. This investigation responded to the increasing adoption of Virtual Environment applications, particularly following the ANA Avatar Xprize Challenge [9] focused on telepresence robotics. Recognizing the absence of established baselines for immersive direct teleoperation, we enhanced our framework with additional functionalities, creating IMMERTWIN. Comparative analysis with TELESIM through identical user surveys revealed comparable task performance metrics while demonstrating significantly reduced mental effort requirements.

5.2 Abstract

We present IMMERTWIN, a mixed reality framework for enhance robotic arm teleoperation using a closed-loop digital twin as a bridge for interaction between the user and the robotic system. We evaluated IMMERTWIN by performing a medium-scale user survey with 26 participants on two robots. Users were asked to teleoperate with both robots inside the virtual environment to pick and place 3 cubes in a tower and to repeat this task as many times as possible in 10 minutes, with only five minutes of training beforehand. Our experimental results show that most users were able to succeed by building at least a tower of 3 cubes regardless of the robot used and a maximum of 10 towers (1 tower per minute). In addition, users preferred to use IMMERTWIN over our previous work, TELESIM, as it caused them less mental workload. The project website and source code can be found at: https://cvas-ug.github.io/immertwin

5.3 Introduction

The ANA Avatar XPRIZE [9] competition has significantly increased interest in telepresence robotics. Telepresence in the competition refers to robots that can be fully controlled by a human operator from a remote location, thereby simulating the operator's presence at the robot's site. The need for a human operator persists due to the limitations of fully autonomous systems, which remain highly constrained and ineffective beyond predefined scenarios [34, 113]. Consequently, researchers have focused on integrating autonomous systems with direct teleoperation to enhance performance and alleviate the cognitive load on users. This integration often involves augmenting visual information or automating specific robot movements [114].

Despite recent advancements, the user's field of view remains restricted to the camera's perspective mounted on the robot in telepresence scenarios or to the user's immediate surroundings in direct teleoperation [34, 4]. This limitation was evident in our previous work, TELESIM [4], where user mobility was restricted due to the robot's movement being directly linked to the user's hand movements in a one-to-one mapping. In this paper, we introduce IMMERTWIN, an immersive and modular plug-and-play framework for robotic arm teleoperation that extends the capabilities of TELESIM. IMMERTWIN situates the user within a Digital Twin environment, allowing for free movement and varied viewpoints while maintaining control over the robot.

We evaluate the performance and user-friendliness of IMMERTWIN through a comprehensive user study involving 26 participants. Each participant was tasked with controlling two robots to repeatedly perform a tower stacking task within ten minutes after a brief five-minute training session. Although none of the participants had prior experience with IMMERTWIN, some were familiar with TELESIM. Our results include cognitive and physical loads and we ex-

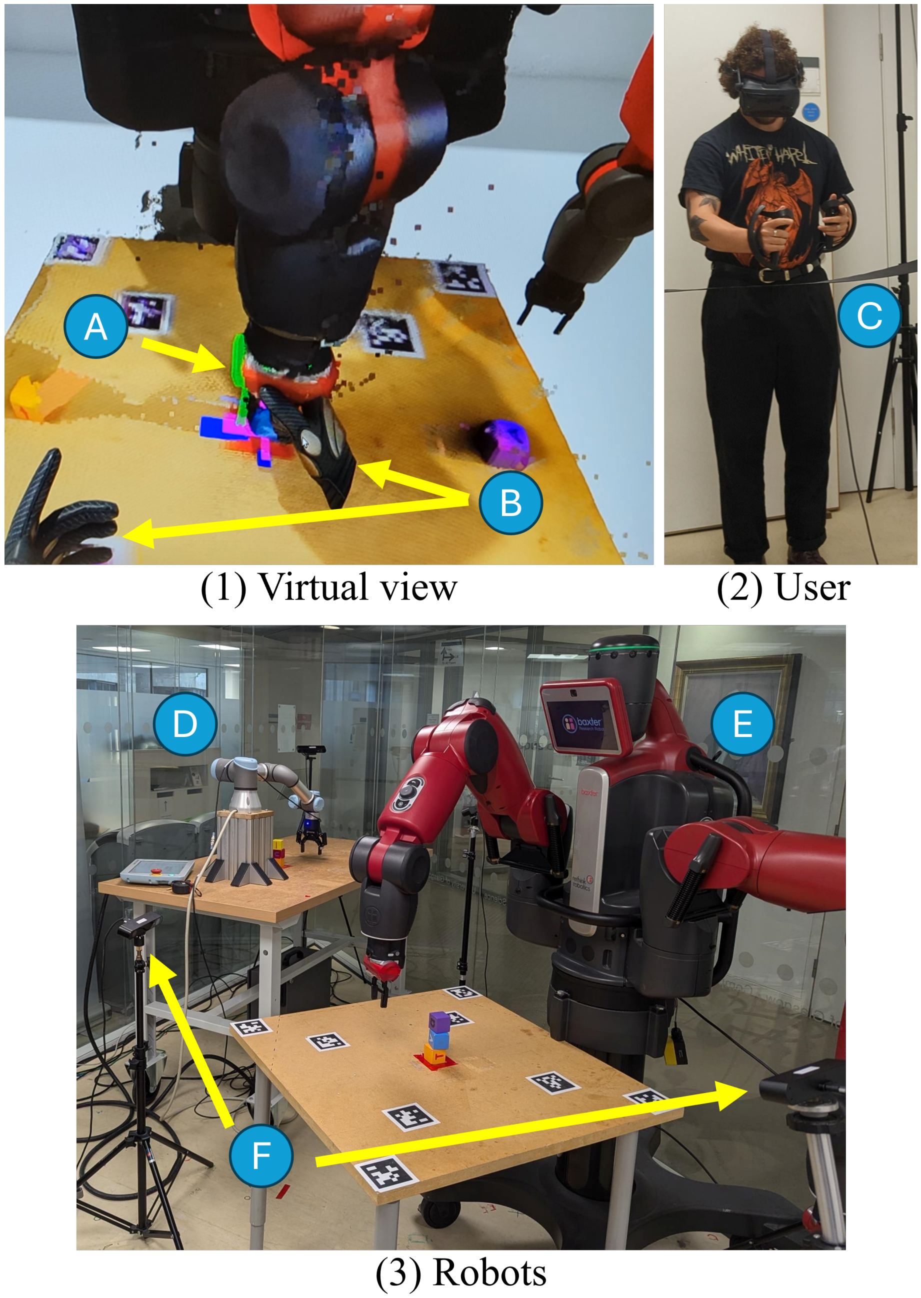


Figure 5.1: Our experimental setup comprises the following components. (1) The view from the user inside Unreal Engine, with their virtual hands (A) and the virtual gripper (B). (2) The user wearing the VR headset, with a black security tape (C) to avoid users walk towards the robots. (3) The room containing both the UR3 robot (D) and the Baxter robot (E), along with 4 ZED 2I cameras (F) to live stream a pointcloud into the virtual environment.

plore the influence of different robot types and prior experience on teleoperation performance. Our findings are then compared with those from TELESIM. Our contributions are:

- Implementing an immersive mixed reality teleoperation system utilising near-real-time point cloud visualisation.
- Experimental validation of the framework's performance using a straightforward task suitable for non-experts.
- A rigorous evaluation involving 26 participants highlighting the user-friendliness of IMMERTWIN.

5.4 Background

Teleoperation allows remote control of robots and has been crucial in enabling Industry 5.0, emphasising human-robot synergy [115]. This technology effectively facilitates the training of robots to execute tasks autonomously, employing methods such as Imitation Learning and Reinforcement Learning [34, 116, 117]. However, teleoperation can exert significant physical and mental stress on users [4]. Moreover, achieving movements similar to those performed by humans during teleoperation remains challenging due to its dependence on factors, including the robot type, User Interface Device (UID) and the operator's prior experience [4].

Researchers have investigated various types of UIDs, including Virtual Reality (VR) [43, 11], joysticks [74], haptic interfaces [118, 21], and vision-based systems [77]. However, direct comparisons of their performance are rare, as studies often focus on enhancing the performance of a single robot or task using assistive methods [114]. These evaluations are typically conducted with small-scale user surveys involving fewer than 20 participants on a single robot [29, 38].

The use of VR headsets to provide additional information through mixed reality has been a significant assistive improvement in teleoperation [44]. VR was initially used as a flight simulator in teleoperation [119] and has since been utilized to control robots. It played a pivotal role in the ANA Avatar XPRIZE challenge, where users teleoperate robots in remote locations, adapting to the robot's capabilities [10, 11]. Typically, users in VR systems view only the camera feed (a 2D video stream) from the robot, which is common in most immersive VR teleoperation systems.

Mixed Reality in robotic teleoperation blurs the line between real and digital environments by overlaying digital information on video streams or integrating real-world data into simulated environments. This technology is regarded as the next step in enhancing VR teleoperation [45]. Most research focuses on the former, such as [44], where surface highlights are added for automatic robot alignment.

Research indicates that visualising a real-time point cloud within a virtual environment yields better results. For example, Su *et al.* [46] conducted a study with 15 participants to evaluate the cognitive and physical load of three systems: one rendering only 2D images, another using a stereoscopic system to supplement the 2D images, and a third rendering a 3D point-cloud. The point-cloud system demonstrated higher performance with reduced cognitive and physical strain on users. However, the study did not report system delays or performance metrics such as average frame rate and camera resolution.

We address the above limitations in this paper by conducting a medium-scale user survey with 26 participants. In our study, participants self-reported minimal experience with robots (Mean 2.8 $\pm$ 1.2 out of 5, with 5 indicating high experience), and our approach exclusively employs a point cloud visualisation, as it has been shown to outperform other methods in creating an immersive environment [46]. However, comparing with the literature, such as [46] is not feasible due to task differences; we report our system's performance, including camera capture rate, resolution, and simulation frames per second. Our study uses Unreal Engine 5.4 and ROS2, making it, to our knowledge, the only teleoperation system to do so, as most studies use Unity. An exception is noted in [120], which used Unreal Engine 4 and ROS1. This difference may be attributed to the higher complexity of Unreal Engine and the relative scarcity of mixed reality tools [121].

5.5 IMMERTWIN Framework

IMMERTWIN is an add-on to TELESIM. TELESIM [4] is a framework designed for the teleoperation of various robotic arms using any control system capable of outputting a 3D pose. For completeness, a brief overview of TELESIM is given below. TELESIM consists of three main components: The first component, *the UID* which, in our previous study [4], consisted of a Steam Index VR controller and a Senseglove, both equipped with a Vive VR Tracker. The second component, *TELESIM* itself, calculates a motion plan in real-time based on the pose provided by the UID. The third component, *the robot interface*, involves transmitting the robot's joint states via ROS2 (Robot Operating System) [80] to any robotic arm compatible with ROS control. TELESIM was deployed into a Rethink Robotics Baxter and a Universal Robots 3 (UR3).

IMMERTWIN functions as an enhancement to TELESIM. It modifies the TELESIM control loop by replacing the UID with a virtual gripper, depicted in bright green in Figure 5.1 as A. This virtual gripper's position is transmitted to TELESIM, enabling Isaac Sim to compute a motion plan for the robot while ensuring collision avoidance. The robot's joint states are then relayed to the robot. In TELESIM, this would conclude its functionality; however, IMMERTWIN extends this by transmitting the real robot's state to Unreal Engine 5.4, as shown in Figure 5.2, which may differ from Isaac Sim's request due to unexpected collisions or hardware malfunctions.

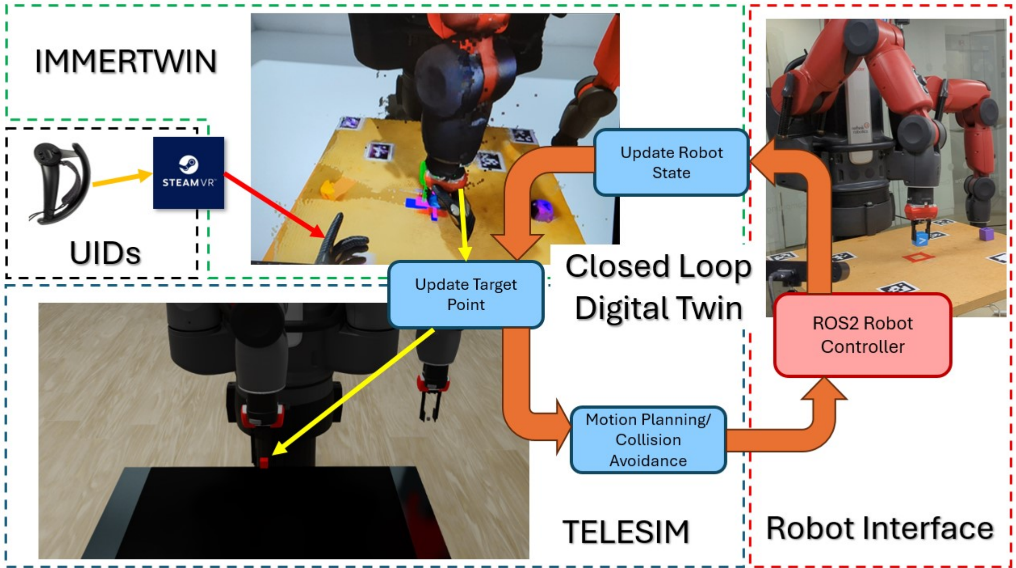


Figure 5.2: Overview of IMMERTWIN. Our framework, IMMERWTIN, shown in the green dotted line, accepts the pose of any 3D VR controller (shown in the black dotted line) to update the position of the user's virtual hand in Unreal Engine. The user can then grab the robotic gripper and move the robot where they want. The new robotic goal is then transmitted to TELESIM (shown in the blue dotted line) to perform motion planning and collision avoidance. The state of the virtual robot is then transmitted to the real robot, shown in the red dotted line, to update its position. Finally, the state of the real robot is transmitted back into Unreal Engine to create a closed-loop digital twin.

These modifications transform TELESIM, which was a digital clone to a closed loop digital twin, thus granting it the name of IMMERTWIN, allowing users to interact with the real robot via ROS2.

In IMMERTWIN's virtual environment, the virtual robot mirrors the real robot. If there are no errors, such as collisions or unreachable positions, the robot gripper aligns with the virtual gripper. Users interacting in VR can manipulate the virtual gripper to control the robot. A point cloud visualisation from two calibrated ZED 2I cameras provides feedback on the world state. This point cloud provides users with near-real-time feedback (100ms) on the robot and its surroundings, enabling them to adjust their actions accordingly. IMMERTWIN's control loop design enables users to release the virtual gripper, maintaining the robot's configuration, thus allowing users to rest, reposition, or gain different perspectives, enhancing manipulation precision.

We opted to use Unreal Engine on Linux as it is the only VR-capable engine on Linux. However, the Unreal Engine cannot build lighting on Linux and we needed to use LUMEN [122], Unreal Engine's resource-intensive real-time lighting system. Linux was required to use ROS2 and interface it with TELESIM. Furthermore, our goal was to run the entire system on a single computer to enhance portability and user-friendliness, reinforcing our plug-and-play approach. Although switching to Unreal Engine for another engine is feasible due to IMMERTWIN's modularity, this task is deferred to future work. Additionally, Unreal Engine was chosen because ROS2 integration had already been developed by Rapyuta Robotics[1]. The integration's sole requirement is a manually operated ROS2 "server" node running FastDDS, functioning similarly to a ROS1 master.

5.6 Experimental Setup

5.6.1 TELESIM

In our previous study [4], we assessed our framework through a user survey involving 37 participants (29 male and 8 female) using the following setup. We employed two robots, Baxter and UR3, each positioned in front of a table with cubes arranged in an isosceles triangular pattern, similar to the setup depicted in Figure 5.1. This experiment was approved by the University of Glasgow Ethics Committee (Application Number 300220026). Participants were tasked with teleoperating both robots while standing with their backs to the VR headset, positioned to the left of each robot at a height of one meter. The headset served as the world's frame and a reference point for the participants. Each robot was operated using a different method: the Baxter robot was operated with a Steam Index VR controller, as illustrated in Figure 5.1, while the UR3 robot was operated using a Senseglove development kit [123]. This kit allowed for the mapping of individual finger movements, with an HTC Vive Tracker mounted on the hand. Only the thumb and index finger movements were used to control a modified T42 gripper from the Yale OpenHand project [3] mounted on the UR3. The combination of Baxter and the VR controller for TELESIM while henceforth be referred to as *Red Design*, while the combination of the UR3, T42 gripper and Senseglove for TELESIM will be referred to as *Blue Design*.

Participants completed a demographics questionnaire using a five-point Likert scale. Following this, they were given five minutes to practice with the Red Design before attempting to stack as many as possible three 40mm cubes in the centre of a table within 10 minutes; cubes were initially arranged in an isosceles triangle. Participants completed the Single Ease Question (SEQ) and the raw NASA-TLX for the Red Design. After that, the same experimental methodology was repeated for the Blue Design. At the end of the experiment, they filled out the Negative Attitude Towards Robots Scale (NARS).

5.6.2 IMMERTWIN

To ensure a fair comparison between IMMERTWIN and TELESIM, we replicated the user survey setup used for TELESIM and the robotic hardware. The combination of Baxter, VR controller and IMMERTWIN will henceforth be referred to as *Orange Design*. However, we implemented several modifications for the Blue Design setup based on feedback received during

[1] https://github.com/rapyuta-robotics/rclUE

the TELESIM user survey. That is, we replaced the custom-built gripper with a Robotiq 2F-85 gripper [104], which has a reduced finger length, addressing issues where the previous gripper impacted robot motion due to safety settings. Additionally, the new gripper resolves issues related to insufficient grasping force, with a maximum force of 235N, preventing cubes from falling during transit. To accommodate the UR3's limited range, we mounted it on a pedestal, enhancing the robot's freedom of movement. During experiments, users reported that the UR3 felt more responsive than the Baxter robot, providing them greater control over its movements. Finally, we transitioned from using a SenseGlove to a Valve Index VR controller for both robots, as the former contributed to the UR3's suboptimal performance in TELESIM. Those changes for the UR3 robot and IMMERTWIN will henceforth be referred to as *Green Design*. Participants no longer need to stand close to the robots; instead, they are positioned two meters away, behind a physical security tape visible on the top-right in Figure 5.1 as C. Additionally, virtual bounding boxes were added in VR to alert users when they are approaching virtual limits set to prevent them from getting too close to the robots. Users were free to move around the room by walking within their available space. For IMMERTWIN, we included the Simulator Sickness Questionnaire [124] to assess potential symptoms such as nausea or headache caused by immersive simulation software. This questionnaire has been validated in mixed reality environments [95].

The cubes were arranged in an isosceles triangle, with each placement presenting varying difficulty levels, similar to [4]. For both the Orange and Green Design, the leftmost cube was the furthest from one of the cameras, resulting in a slightly unstable image in the 3D cloud during the experiments. The other two cubes were placed at the robot's range limit, often requiring users to rotate the gripper. The tower position was at the intersection of all cubes. This position proved challenging for some Orange Design participants, as the robot arm could only sometimes rise sufficiently to stack the cubes, depending on the gripper's angle. Users encountering this difficulty were advised to adjust their hand rotation. This issue was not present with the Green Design.

We recruited 26 participants (21 male, 5 female), with an average age of 27.8 years and a standard deviation of 6.8, from the University of Glasgow. Initially, participants completed a brief questionnaire about their background, following the same experimental methodology as TELESIM. The experiment's conditions and objectives were then explained: participants had to manipulate a virtual gripper in a virtual world to grasp and stack three cubes arranged in an isosceles triangle in front of the robot to the centre of the table. They were given five minutes of practice and 10 minutes to build as many towers as possible. Users were only informed of the remaining time if they inquired. After using the Orange Design, participants completed three questionnaires: the raw NASA-TLX questionnaire, the SEQ, and the Simulator Sickness Questionnaire. They then repeated the procedure for the Green Design. We decided against randomising the order, first to keep results consistent and comparable with TELESIM[4] and second, as randomizing the order would require more participants in order to achieve statistical significance.

5.6.3 Hardware

As detailed in Section 5.5, the point cloud data for each robot is sourced from two Zed2I cameras[2], positioned 120 degrees apart and oriented towards the working area. This setup encompasses the robot, the table, and the three cubes intended for user manipulation, as described in Section 5.6.1. The point clouds are updated every 100 milliseconds, as higher refresh rates adversely affect the frame rate. The point cloud is generated from RGB and depth images transmitted to Unreal Engine using ROS2, processed via a compute shader on the GPU to enhance performance. We developed this compute shader[3] to reconstruct the 3D point cloud from RGB and Depth images transmitted by the two ZED2I cameras, as ROS2 point cloud messages are more expensive to parse and need to be processed on the CPU. Since compute shaders executes every frame, we limit the point cloud's refresh rate by sending images at 10Hz. The images are captured at 720p resolution since higher resolutions result in performance degradation.

To maximise performance, given VR's suboptimal optimisation on Linux, image capture and transmission were performed on an Nvidia RTX 2080TI, while the Unreal Engine ran on an Nvidia RTX 4090. Both GPUs were housed in the same computer to minimise latency that could arise from networked machines, especially when transmitting large volumes of images. Additionally, the TELESIM component of our framework was executed on a separate computer equipped with an Nvidia RTX 3060. This setup ran Isaac Sim alongside ROS2 control for robot control. The only data exchanged between the two computers via ROS2 over an Ethernet connection were the 3D pose of the virtual gripper and the real robot joint states, resulting in negligible network delay. However, the approximate 100ms delay inherent to TELESIM persisted in IMMERTWIN, primarily due to path planning and the robot's slow movement for safety reasons. This configuration allowed us to achieve an average of 40 fps on the VR headset, with minor tearing occurring if the user moved their head rapidly. Nevertheless, as described in Section 5.7, few users reported experiencing nausea or other adverse effects from using VR.

5.7 Evaluation

Figure 5.3 shows that participants of IMMERTWIN successfully constructed a maximum of 10 towers with both robots. Notably, users operating the UR3 with IMMERTWIN (Green Design) consistently completed more towers than participants using any other robot across both TELESIM and IMMERTWIN. Interestingly, for the Baxter robot, participants who built fewer than five towers did so more frequently in TELESIM than in IMMERTWIN. This trend can

[2]https://www.stereolabs.com/en-fr/store/products/zed-2i
[3]https://github.com/cvas-ug/immertwin

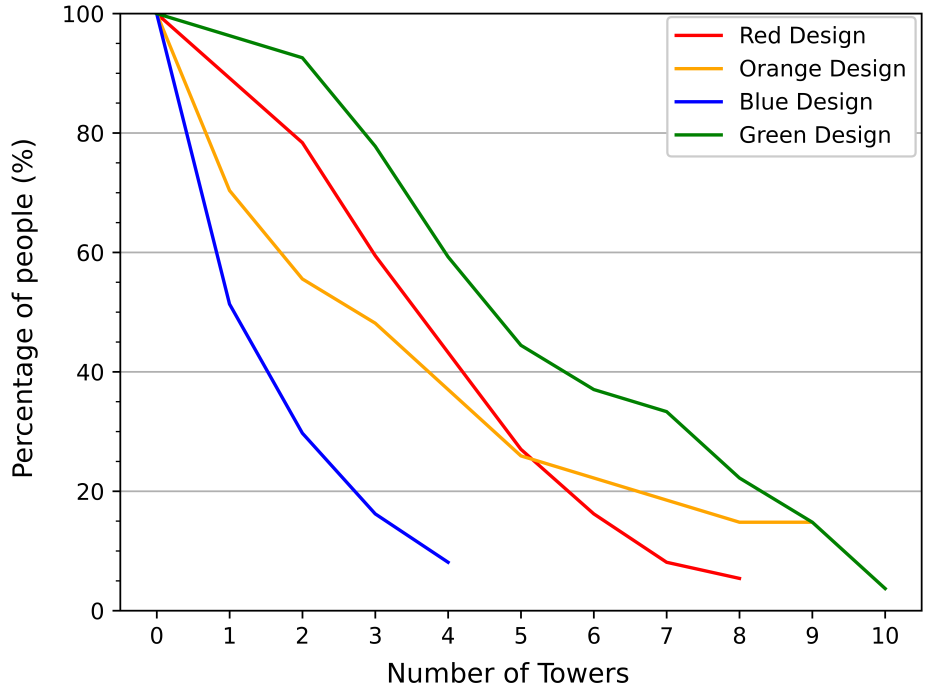


Figure 5.3: Population percentage for each tower completed for both robots for TELESIM and IMMERTWIN, respectively.

be attributed to user feedback during and after the experiments, where participants noted that Baxter was less precise, particularly participants who encountered the stacking issue discussed in Section 5.6.2, and preferred the Green Design. One reason cited was that the UR3's gripper is larger when open than Baxter's, allowing for more margin of error when grasping objects. However, participants who achieved more than five towers favoured the Orange Design over the Green Design, although they did not provide specific reasons other than the UR3 being slower than Baxter. The UR3 movement speed was intentionally set low to prevent damage to the robot. This preference is also reflected in the results of the Single Ease Question, with a mean score of 3.81 ± 1.52 for the Green Design and 3.5 ± 1.50 for the Orange Design where a higher score indicates an easier task difficulty. Although there is no significant statistical difference, users tended to prefer the Green Design. All statistical tests were conducted using the Mann-Whitney U Test. Given our relatively small sample size, a non-parametric test was deemed appropriate, as recommended by Rochon *et al.* [125]. Furthermore, all of our statisticals test use as a null hypothesis that the effect being studied does not exist (eg. There are no difference between the placing rate of the Orange and Green Design)

The perceived superiority of the Green Design is further corroborated by Figure 5.4, which present the ratios of Placing Rate, Collapse Rate, and Still in Place Rate. The Blue Design exhibits a statistically significant difference (99.9%) in Placing Rate, Collapse Rate, and Still in Place Rate. However, no statistical difference is observed for the Red Design [4] or either

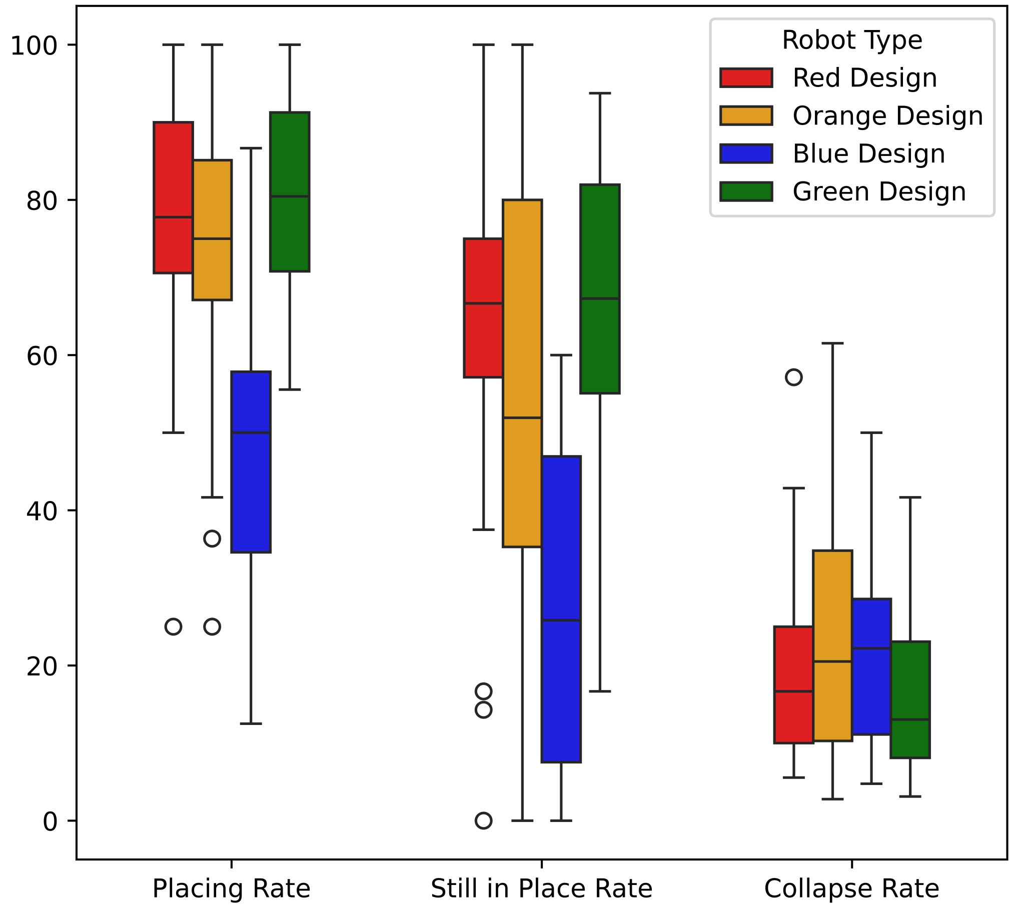


Figure 5.4: Ratio of different statistics collected during the experiment. The Placing Rate is calculated as the number of place actions over the number of picking actions. The Collapse Rate is calculated as the number of Collapse actions over the number of picking actions. The Still in Place Rate is calculated as the number of Place actions minus the number of collapses over the number of picking actions, effectively rating the tower's stability.

design using IMMERTWIN. This indicates that IMMERTWIN does not significantly improve these statistics. Interestingly, it also suggests that the additional freedom of motion afforded by immersion in a Digital Twin does not significantly enhance teleoperation performance for certain robot types. The poor performance of the Blue Design may be attributed to an inadequate UID or the low quality of the gripper used, as mentioned in Section 5.6.2.

The observation that IMMERTWIN does not yield performance gains in teleoperation does not negate the utility of VR. Among the 26 participants, 12 had previously participated in TELESIM and were asked to express their preference between the two systems. Three-quarters of these participants favoured IMMERTWIN, with only one participant preferring TELESIM. Furthermore, Figure 5.5, which illustrates the mental effort required during the task, indicates that IMMERTWIN demands significantly less effort than TELESIM, with at least a 99% significance level. However, this does not hold for the physical aspect, as shown in Figure 5.6, where the Red design exhibits at least a 95% significantly lower physical effort than all other experi-

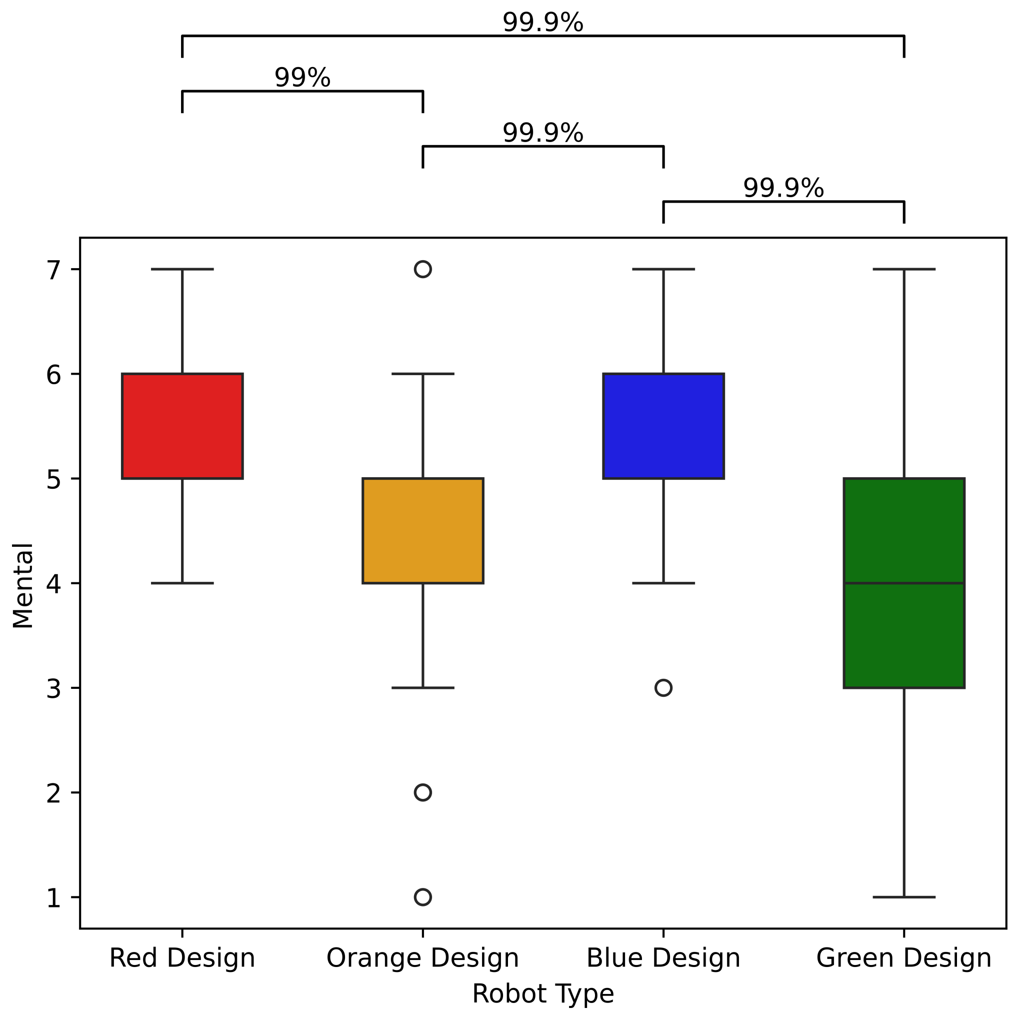


Figure 5.5: Result of the raw NASA-TLX mental aspect, which evaluates how mentally demanding the task was. A low score indicates a lower effort. The horizontal bar at the top indicates a significance value between the two items indicated by the ticks at both ends of the bar.

ments. Participants using IMMERTWIN with both robots observed a similar level of exhaustion, while the Blue Design was statistically more exhausting than the other robots. Based on user feedback, we hypothesise that IMMERTWIN causes an average level of physical exhaustion due to users needing to bend for a closer view.

An interesting difference noted in the NASA questionnaire is the pacing. Although the experimental setup was identical for both robots, IMMERTWIN users perceived the pace as being 95% significantly slower, with an average pace of 3.8 for IMMERTWIN compared to 4.1 for TELESIM, in which a higher number means that they felt more rushed. We attribute this difference to users losing their sense of time, as intense mental activity has been shown to cause time distortion in VR environments [126].

Finally, the Simulator Sickness Questionnaire results indicate that the most common symptoms, in order of prevalence, are fatigue, general discomfort, eyestrain, and headache. However, for most users, these symptoms are mild, with fatigue having a mean score of 1.73 ± 0.92, where

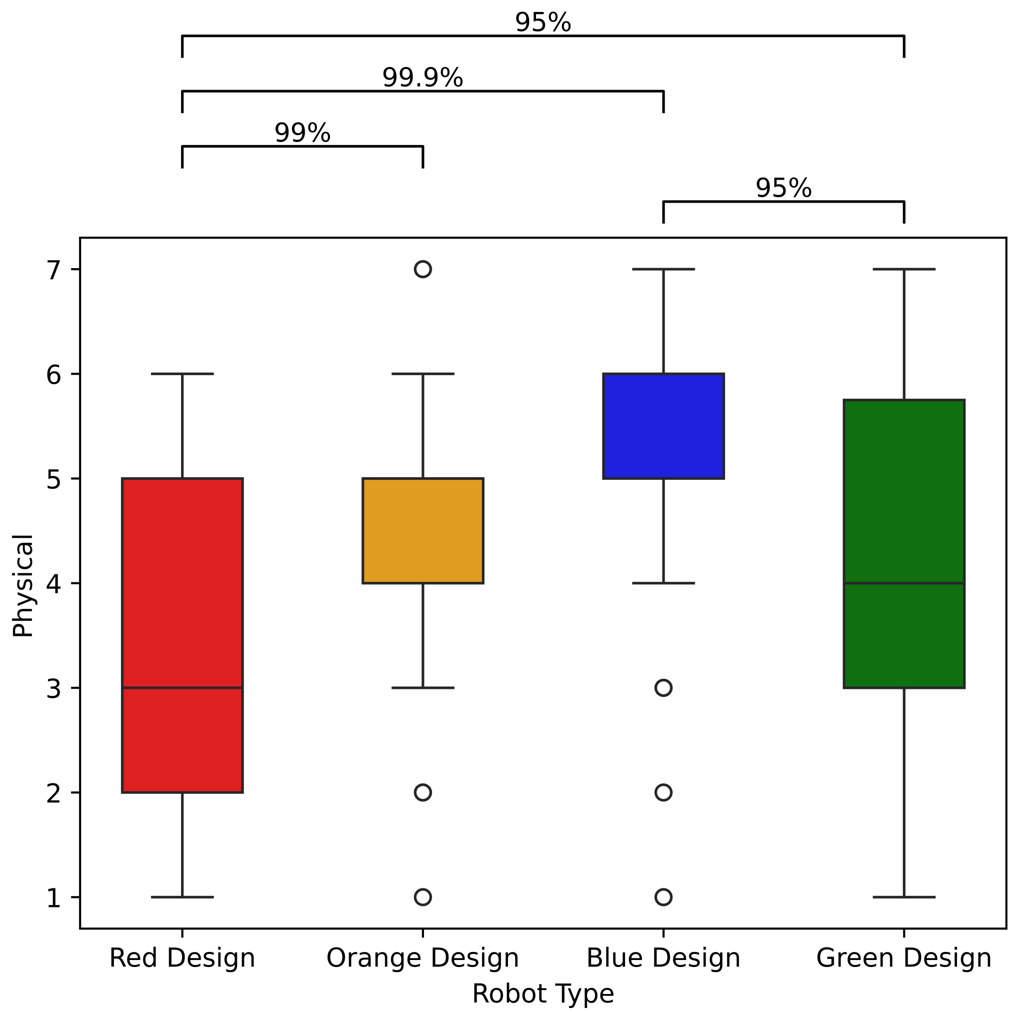


Figure 5.6: Result of the NASA-TLX physical aspect, which evaluates how physically demanding the task was. A low score indicates a lower effort. The horizontal bar at the top indicates a significance value between the two items indicated by the ticks at both ends of the bar.

1 indicates no symptoms, and 4 indicates severe symptoms.

5.8 Conclusion and Future Work

This paper explored the capabilities and user experiences of IMMERTWIN compared to its predecessor TELESIM [4]. That is, IMMERTWIN allows for plug-and-play teleoperation in an immersive digital twin using a VR headset, while TELESIM only allows teleoperation using direct visual feedback. Overall, IMMERTWIN demonstrates the potential of integrating VR into teleoperation frameworks, offering a more engaging and less mentally taxing user experience. IMMERTWIN is available on GitHub[4], allowing developers to perform teleoperation on their robots with minimal setup time. Despite IMMERTWIN not demonstrating significant

[4]https://cvas-ug.github.io/immertwin

performance improvements in teleoperation metrics, it was favoured by most participants who had previously used TELESIM. This preference underscores the potential benefits of immersive VR environments in enhancing user experience, even if quantifiable performance gains are not immediately evident. The mental effort required by IMMERTWIN was significantly lower than that of TELESIM, suggesting that the immersive nature of the VR environment may alleviate cognitive load. However, physical effort remained at the same level, particularly with the UR3 in TELESIM, which was noted to be more exhausting than other setups. This highlights the need for further refinement in balancing cognitive and physical demands in VR-based teleoperation systems.

The study in this paper also revealed interesting insights into user perceptions of time and pacing within VR environments, with IMMERTWIN users experiencing a slower perceived pace. This aligns with existing research indicating that intense mental activity in VR can alter time perception, a factor that should be considered in future VR application designs [126]. Moreover, while mild symptoms of simulator sickness were reported, they were generally manageable, indicating that IMMERTWIN's VR implementation is well within acceptable limits for user comfort. However, the long lasting effect of virtual reality teleoperation are left as future work. The feedback gathered provides valuable guidance for future iterations of IMMERTWIN, focusing on enhancing physical ergonomics and further reducing cognitive load.

While IMMERTWIN demonstrates the potential of mixed reality to enhance teleoperation by reducing cognitive load and improving user interaction, it is important to acknowledge a key limitation: the interdependence between robot type, gripper design, and levels of immersion. Specifically, the experiments conducted in this chapter used fixed combinations of these elements, such as pairing the UR3 robot with a specific gripper and immersion level. This coupling makes it difficult to disentangle the individual contributions of each factor to the observed performance improvements. For example, while immersive virtual reality might reduce cognitive load, the performance gains could also be influenced by the robot's design or the gripper's limitations.

This limitation restricts the generalizability of the findings, as it is unclear whether similar results would be achieved with different robots or grippers under varying levels of immersion. To address this in future work, a more modular experimental design is needed. This could involve systematically varying each factor independently, such as using interchangeable grippers on a single robot platform or testing multiple levels of immersion across different robotic configurations. Additionally, employing standardized performance metrics across all setups would enable more robust comparisons and clearer insights into how each component impacts teleoperation performance.

Despite this limitation, IMMERTWIN provides valuable insights into how immersive technologies can enhance teleoperation systems. However, caution must be exercised when interpreting these results due to the intertwined nature of hardware and immersion variables. Future research consists of optimising VR systems further, exploring ways to mitigate physical strain and enhancing the intuitive control of robotic systems in immersive environments. This approach aims to leverage human intuition and a high-level task overview while allowing robots the autonomy to execute precise actions, such as accurate object picking and placing. Previous research in this domain has demonstrated that shared autonomy system could help reduce the cognitive strain of the user[30]. We intend to examine whether the level of environmental realism affects teleoperation performance. Currently, the teleoperation environment is a featureless, white room with only the robot and workspace rendered. We plan to enhance this by immersing users in 3D reconstructed representation of the real environment. We hypothesize that while this change may not directly impact performance metrics, it could enhance the overall user experience. Additionally, we aim to investigate whether the increased physical exhaustion reported with IMMERTWIN compared to TELESIM is attributable to the vertical height of the virtual environment, as taller users needed to bend in order to see the environment, as the virtual height of the workplace matched that of the real world. To address this, we will provide users with the ability to adjust the environment's height to their preference, potentially reducing physical strain. Finally, we plan to replace the cube stacking task used as a benchmark for a peg-in-the-hole task which is a current manufacturing task, requiring high level of details and precision.

Chapter 6

Conclusion

6.1 Contributions

This thesis advances current literature in terms of the following key major contributions as mentioned in Section 1.6:

- Development of a framework for systematic evaluation and comparative analysis of robotic simulation platforms, with specific implementation in ROS2-based environments (Chapter 2).
- Development of baseline performance metrics for direct teleoperation through systematic analysis of user interaction across multiple robotic platforms and control interfaces (Chapter 3).
- Quantitative investigation of teleoperation performance factors through a large-scale international study ($n = 73$), examining the influence of robot types, end-effector design, and control methodologies on operator performance (Chapter 4).
- Analysis of correlations between operator experience, robot trust metrics, and teleoperation performance outcomes (Chapter 4).
- Integration of immersive control capabilities within the modular framework, enabling enhanced operator interaction within the digital twin environment (Chapter 5).
- Experimental evaluation of immersion levels and their impact on teleoperation performance metrics and operator's workload (Chapter 5).

And the following minor contributions:

- Development of a modular, plug-and-play teleoperation framework using digital twin technology for robotic manipulator control, emphasizing system adaptability (Chapter 3).

The research objectives outlined in Section 1.4 have been thoroughly investigated and validated through our experimental work. The subsequent sections provide a detailed exposition of our key findings.

6.1.1 Benchmarking Simulation Software

In Chapter 2, we investigated the following question:

> **RQ_1** Are current simulation software for robotics suitable for acting as a digital twin or digital clone?

For this, we developed a rigorous methodology to systematically benchmark robot simulations under consistent parameters, tasks, and scenarios. Our approach involved replicating identical environments across different simulation platforms.

The first task employed a complex manipulation scenario with 35 cubes, where the robot randomly constructed three towers of five cubes each. We selected this task because robotic manipulation fundamentally involves object pick-and-place operations, and it effectively evaluates the simulation's capability to execute prolonged tasks while assessing the physics engine's stability. The second task featured a six-cube pyramid, with the robot strategically throwing a cube to collapse the structure. This scenario was designed to demonstrate the physics engine's accuracy and repeatability. For both tasks, we meticulously recorded task success metrics—specifically, the number of cubes remaining in towers one minute post-simulation and cubes displaced from the pyramid—alongside comprehensive CPU and memory usage data.

Initially, we examined eight robotics-compatible simulation platforms. However, we eliminated Unreal Engine and Unity due to incomplete ROS2 plugin support and Mujoco because its non-modifiable environment contradicted digital twin requirements. Our investigation focused on five simulation software packages: Ignition, Webots, Isaac Sim, PyBullet, and CoppeliaSim. We discovered that using default parameters, at least three simulations, Ignition, Isaac Sim, and Webots, could complete robotic manipulation tasks reliably. These platforms demonstrated digital twin capabilities by rendering real-world data and controlling robots in near-real-time, aligning with the the AMRC (Advanced Manufacturing Research Centre) definition of digital twin as *a live digital coupling of the state of a physical asset or process to a virtual representation with a functional output.* [127] Isaac Sim particularly stood out, with practical validation from its implementation by Amazon Robotics [128]. Considering its photorealistic rendering and integrated real-time path-planning capabilities, we selected this platform for our subsequent research. CoppeliaSim and PyBullet, while resource-efficient, did not meet our specific research requirements without significant parameter optimization.

Our research addressed a significant literature gap by providing the first comprehensive evaluation of simulation software compatibility with ROS2. However, we acknowledge the rapidly evolving nature of simulation technologies. The landscape has changed since our initial investigation, with platforms like Isaac Sim reducing resource consumption and Unreal Engine expanding ROS2 support as demonstrated in Chapter 5.

6.1.2 Performance of Teleoperation

Drawing from our findings in Chapter 2, we used Isaac Sim as a foundational platform to develop a modular framework for robotic direct teleoperation. This framework was subsequently employed in Chapters 3 and 4 to investigate the following research question:

> **RQ_2** How do variations in User Interface Device (UID), robotic arm configurations, and gripper designs influence user performance and cognitive workload during teleoperation tasks?

Our research encompassed comprehensive user surveys across multiple countries, recruiting 73 participants. We systematically evaluated three distinct robots, each equipped with three different gripper types, controlled through two separate methods utilizing direct visual feedback.

Each participant was placed beside the robot at a safe distance to prevent injuries. Users were first given 5 minutes of practice time to learn how to teleoperate the robot, then 10 minutes where they were asked to move 3 cubes from their position setup in an isosceles triangle to a stacked tower in the centre of the tower as many times as they could. They could manipulate the robot using a VR controller or a VR-tracked glove. Whenever the user moved their hand, the target object's position was updated inside of Isaac Sim. This caused the simulation software to start planning a path to the target object while performing collision avoidance. Finally, the state of the virtual robot is transmitted to the real robot, creating a digital clone.

Our user surveys revealed significant insights into teleoperation performance. The Yellow Design (UR5e, VR controller and Robotiq Gripper) robot demonstrated superior task completion rates compared to other platforms. Specifically, the hardware plays an important role in teleoperation performance, with the Yellow Design achieving a higher average number of towers completed. Interestingly, the statistics, such as placing and collapse rates between the Yellow and Red Design (Baxter, VR controller) shown in Figure 4.8, indicate that user abilities are similar, but the hardware allows for more towers to be built. This result is corroborated by the Single Ease Question (SEQ), which measures the user's perceived task difficulty. Despite the difference in the number of towers built, the Red and Yellow Design have statistically similar results. It appears that users estimated the theoretical maximum based on their experience, resulting in comparable SEQ scores for both robots despite performance differences. The Blue Design (UR3, Senseglove and T42 gripper) performed worse than the Yellow Design and Red due to a combination of a different controller type, a custom-built gripper, and a short range of motion. The controller type was a Senseglove, which allows for mapping the user's finger directly to the individual fingers of a custom T42 gripper with a weak grip. These differences made it harder for all users to control the Blue Design while increasing the mental and physical exhaustion of the users. We suppose the additional range of motion of the UR5, will help reduce the mental and physical exhaustion but that it will still be higher than the Yellow Design due to the controller and design of the gripper.

To comprehensively explore the relationship between hardware and user workload, we employed the NASA TLX questionnaire. Our results confirm that teleoperation is challenging for users, as they have to spend a higher-than-average amount of effort to perform the task. However, the Yellow Design allowed participants to experience less mental workload than other robots. Additionally, the Blue Design forced users to experience a higher physical workload than other robots. These results answer $\mathbf{RQ_2}$, showing that a combination of robot stability, operating range, gripper design, and controller type enhances teleoperation performance. We can confirm that a setup similar to the Yellow Design configuration, a precise robot with a range similar to a human arm and a VR controller such as the Vive Index, is optimal, as we suppose the additional degrees of freedom for the controller, such as individual finger control do not bring any performance improvement.

To answer the following question:

> $\mathbf{RQ_6}$ Does Virtual Reality Immersion help reduce the mental and cognitive load of the user during teleoperation ?

We enhanced our teleoperation framework by introducing an additional control loop step. Instead of directly observing the real robot, users interacted with a digital twin representation, featuring a near real-time world view generated through merged point cloud rendering from two depth cameras, as detailed in Chapter 5. This modification transformed our system from a digital clone to a closed-loop digital twin. We utilized Unreal Engine as the digital twin, as it was the sole software compatible with Virtual Reality on Linux working with ROS2. Users could teleoperate the robot by manipulating a virtual gripper within the digital twin using a VR controller. The virtual gripper's position was transmitted to our framework, which executed motion planning and collision avoidance. The real robot's state was subsequently updated based on the motion planner's results, updating the digital robot's state within our Digital Twin.

Using our established methodology, we evaluated the framework by conducting a user survey with 26 participants as described in Chapter 5. Unlike previous experiments, users were positioned far from the robot, with tape demarcating their operational area as the virtual environment eliminated the need for physical proximity. Our findings revealed nuanced insights. Participants did not demonstrate statistically significant performance improvements in the immersive environment. However, users reported significantly lower mental workloads compared to the direct visual teleoperation system. Notably, participants who experienced both methods expressed a clear preference for the immersive version. This preference highlights the potential of immersive VR environments to enhance user experience, even when quantifiable performance gains are not immediately evident. Our research also uncovered intriguing observations regarding time perception. Participants using the immersive framework perceived time as progressing more slowly, corroborating existing research findings [126].

Our results underscored the need for comprehensive long-term evaluation. While users reported minimal simulator sickness, extended operating sessions may yield different physiological and psychological responses. Future research should investigate the prolonged effects of immersive teleoperation on user workload, time perception, and overall health.

This thesis contributes a versatile direct teleoperation framework. It establishes a comprehensive baseline for understanding how hardware configurations and immersion levels influence user physical and mental workload and task performance in robotic teleoperation scenarios. Our approach provides valuable insights into the complex interactions between human operators and robotic systems, opening new avenues for research in immersive teleoperation technologies.

6.1.3 Importance of External Factors

To establish an accurate baseline for direct teleoperation, we needed to examine the human factors influencing performance and workload, focusing on previous experience in related fields. We investigated the following research questions:

> **RQ$_3$** To what extent does prior user experience with robotics and virtual reality affect task performance and adaptability in teleoperation scenarios?
>
> **RQ$_4$** How do cultural differences between operators from different regions (e.g., UK and Japan) impacts workload and performance in robotic teleoperation?

Our investigation was conducted alongside user surveys described in Chapters 3, 4, and 5. We examined previous experience with Virtual Reality, robots, and wearables using a 5-point Likert scale, asking users to self-report their experience in specific fields.

While self-reporting introduces limitations in consistency and objectivity, our results provided valuable insights. We found that prior Virtual Reality experience did not significantly impact teleoperation performance, regardless of the user's cultural background. However, experience with robots and wearables demonstrated cultural variations. Japanese users reported, on average, more experience with robots and less with wearables than British users. These findings did not establish a definitive relationship between previous robotic or wearable experience and teleoperation performance. We acknowledge that the broad question about robot experience may be problematic, as the Japanese population is more frequently exposed to robots, particularly in service contexts such as food delivery [129]. Further research is needed to determine whether such exposure translates to teleoperation capabilities.

We observed that spatial skills potentially play a crucial role in teleoperation performance. This hypothesis emerged from the notably higher performance of two Japanese users with experience in motion retargeting systems. However, comprehensive spatial skill assessments would have excessively prolonged our user surveys. While this thesis initiates research on the impact of previous experience in robotic teleoperation, more extensive investigation is required to draw definitive conclusions about the necessary skills.

While our investigation towards **RQ$_4$** led to the discovery of the difference in experience with robots and wearables between British users and Japanese users, we were also able to investigate the following question and the difference in trust between both cultural backgrounds:

> **RQ$_5$** What is the relationship between operator trust in robots and their performance during teleoperation tasks, particularly across different cultural contexts?

Our results did not conclusively demonstrate a correlation between robot trust levels and teleoperation performance. However, we discovered a significant cultural difference: Japanese users exhibited higher trust in robots, particularly concerning robotic decision-making, compared to British participants. This finding contradicts previous research [47], underscoring the need for further investigation in this domain.

Our research provides a foundational understanding of the complex interactions between cultural background, technological experience, and robotic teleoperation performance. It simultaneously highlights the complexity of these relationships and the necessity for continued scholarly exploration.

6.2 Future Work

This section provides possible research directions and future extensions of the research presented in this thesis.

6.2.1 Validation of Experimental Findings

The experiments conducted in the UK and Japan provided valuable insights into the impact of controller types, gripper designs, and robot stability on teleoperation performance. However, to conclusively validate these findings, future work should focus on repeating these experiments under more controlled and standardized conditions. This would involve ensuring consistency in hardware configurations, task setups, and participant demographics across both countries.

Firstly, the experiments should be conducted with identical robotic systems and grippers in both locations to isolate the effects of hardware variations. For example, using the same robot models (e.g., UR3, UR5e) and gripper types (e.g., linear, fingered) would eliminate potential biases introduced by differences in hardware capabilities. Additionally, randomizing task orders and control methods for participants can help mitigate learning effects observed during sequential trials. Secondly, the sample size should be expanded to include a more diverse range of participants from more countries. This would allow for a more comprehensive analysis of cultural differences in teleoperation performance and trust toward robots. Finally, expanding upon the methods of analysis to use more complex statistical approaches such as Bayesian methods could improve insight generation.

By addressing these areas in future research, it will be possible to draw more definitive conclusions about the interplay between hardware configurations and user performance in teleoperation systems across diverse cultural contexts.

6.2.2 Influence of Previous Expertise

As discussed in Section 6.1.3, our investigation into the relationship between prior expertise and its effects on teleoperation performance and operator workload remains preliminary. Future research necessitates a more comprehensive experimental framework evaluating fundamental skills: enhanced motor capabilities, spatial reasoning abilities, and domain-specific expertise. While additional skills might influence teleoperation performance, these capabilities likely exert the most significant impact.

Such investigations would employ a methodology similar to our current user surveys, with specific refinements. The experimental protocol would utilize a single robot type with consistent control methods, as investigating multiple hardware configurations would render the research impractical due to the extensive time requirements for skill evaluation. This focused approach could validate our initial findings confirming that cultural influence does not have an influence on teleoperation performance.

Furthermore, systematically analysing required skills would reveal essential operator expertise characteristics. We could develop a standardized operator assessment framework by systematically identifying and quantifying critical skills. This approach aligns with Industry 5.0 standards outlined in Section 1.3, potentially establishing operator certification criteria. Such certification would confirm the operator's possession of fundamental and advanced skills necessary for complex modern industrial tasks. Identifying specific skills would facilitate targeted training programs, establishing efficient operator development pathways.

6.2.3 Virtual Reality Applications and Limitations

Our research in Chapter 5 implemented a digital environment utilizing 3D point cloud visualization for world state representation. As discussed in Section 5.4, alternative methods exist for virtual environment rendering. The ANA Avatar Challenge employed stereoscopic cameras [11], while Su et al. [46] explored multiple 2D screen embeddings within digital environments, representing various camera perspectives. Their findings suggested point cloud rendering yielded superior performance, though additional research is required to provide further evidence of these results across different operational contexts and user populations.

Our current implementation presents users with a minimalist white environment containing only the robot and workspace elements. We hypothesize that implementing fully reconstructed 3D environments through Neural Radiance Fields (NERF) [130] or Gaussian Splatting [131] techniques could enhance user immersion. These advanced rendering technologies could create photorealistic representations of the operational environment, potentially offering a more natural and intuitive interaction space. While such enhancements might not directly impact teleoperation performance metrics, they could significantly affect cognitive load and temporal perception during extended operational sessions.

Users already report temporal distortion within virtual environments. Enhanced environmental realism could produce two contrasting effects: either anchoring users more firmly in reality by providing familiar visual cues, potentially normalizing temporal perception and challenging existing research linking VR to time distortion [126], or intensifying immersion through increased environmental fidelity, further altering temporal perception. This investigation would contribute valuable insights into VR's influence on human temporal cognition and provide practical guidance for future teleoperation interface design. The results could significantly impact the development of more effective and user-friendly teleoperation systems while advancing our understanding of human perception in virtual environments.

Furthermore, participants reported minimal VR sickness symptoms during our experiments. However, exposure remained limited to 10-minute sessions without breaks. We identified the need to increase the Frames Per Second (FPS) in simulated environments, as frame rate significantly influences VR sickness occurrence [132]. Research indicates that frame rates below 120 FPS can induce VR sickness symptoms [132]. Our simulation operated at 40 FPS, primarily constrained by computational demands of near real-time point cloud updates. Additional performance limitations stemmed from Linux-specific Unreal Engine constraints, requiring resource-intensive real-time lighting implementations. Future research addressing these technical challenges could substantially reduce VR sickness, particularly during extended operational sessions. Performance optimization could potentially decrease user cognitive load by eliminating the need for neural interpolation between frames in the simulated environment.

Moreover, investigating methods to optimize rendering pipelines and implement efficient point cloud processing algorithms could significantly enhance the user experience. These improvements would enable longer operational sessions while maintaining user comfort and performance levels. Such advancements would prove particularly valuable for industrial applications requiring sustained teleoperation periods, aligning with the broader goals of establishing a framework that could be used for Industry 5.0.

6.2.4 Impact of Trust in Human-Robot Interactions

In our experimental design, trust assessment was limited to post-experiment questionnaires. We subsequently recognized the value of implementing pre and post-experiment trust evaluations to measure teleoperation's impact on user trust dynamics. While this thesis established a baseline comparing robot trust levels between British and Japanese populations, it did not capture potential trust evolution through teleoperation exposure. Dual questionnaire implementation would have enabled analysis of these temporal changes. We hypothesize that higher system trust levels correlate with improved performance, as users confident in the system's precise movement replication may operate more efficiently. This aligns with Dhuliawala et al. [133]'s findings regarding human-AI interaction, where system errors progressively erode user trust and diminish performance. Additional research specifically focusing on robotic teleoperation could validate these trust-performance relationships.

Further investigation could identify specific actions or interface elements that enhance robot trust and improve performance. One innovative approach might involve replacing virtual robot representations with human avatars, creating an abstraction layer that frames the interaction in more familiar human terms. The potential impacts of such anthropomorphic representations on trust development and operational performance remain unexplored. Moreover, this research could investigate how different avatar designs, interaction modalities, and feedback mechanisms influence trust formation and maintenance. Understanding these dynamics could inform the development of more effective teleoperation interfaces that naturally build and sustain user trust while maintaining operational efficiency. Such findings would contribute valuable insights into human-robot interaction theory and practical interface design.

6.2.5 Shared Autonomy Frameworks

Our research has established a robust foundation for direct teleoperation, addressing a critical gap in the existing literature, which predominantly focused on shared autonomy systems. The development of this baseline opens up new avenues for empirical investigation. Specifically, it enables the reevaluation of shared autonomy systems within the context of the cube-stacking task presented in this thesis. Such an approach would allow for a quantitative assessment of the autonomous components' contributions to teleoperation performance. Based on current literature, we hypothesize that shared autonomy systems will reduce cognitive load and enhance overall task performance depending on the specific task[114, 30]. However, more research would be needed to explore the impact of the robotic task on the user mental and physical workload.

Furthermore, this experimental framework facilitates a comparative analysis of diverse shared autonomy techniques. This could potentially elucidate which specific methods have the most significant impact on mitigating cognitive burden during teleoperation tasks. Currently, insufficient empirical evidence exists to determine the relative efficacy of various shared autonomy approaches. For instance, it remains unclear whether the integration of automatic grasping mechanisms [72] offers superior benefits compared to strategies that align the robot with surface normals [44] or any other shared autonomy methodologies that have been developed. This research trajectory promises to yield valuable insights into optimizing human-robot collaboration in teleoperation scenarios, potentially informing the design of more efficient and user-friendly robotic systems.

Bibliography

[1] Michael A. Goodrich and Alan C. Schultz. Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, 1(3):203–275, 2007. ISSN 1551-3955, 1551-3963. doi: 10.1561/1100000005. URL http://www.nowpublishers.com/article/Details/HCI-005.

[2] Valve Index, 2022. URL https://store.steampowered.com/valveindex.

[3] Yale OpenHand Project - Model T42, 2023. URL https://www.eng.yale.edu/grablab/openhand/model_t42.html.

[4] Florent P Audonnet, Jonathan Grizou, Andrew Hamilton, and Gerardo Aragon-Camarasa. TELESIM: A Modular and Plug-and-Play Framework for Robotic Arm Teleoperation using a Digital Twin. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17770–17777, Yokohama, Japan, May 2024. IEEE. ISBN 9798350384574. doi: 10.1109/ICRA57147.2024.10610935. URL https://ieeexplore.ieee.org/document/10610935/.

[5] Florent P. Audonnet, Andrew Hamilton, and Gerardo Aragon-Camarasa. A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pages 755–762, November 2022. doi: 10.23919/ICCAS55662.2022.10003832. ISSN: 2642-3901.

[6] Florent P Audonnet, Andrew Hamilton, Yukiyasu Domae, Ixchel G Ramirez-Alpizar, and Gerardo Aragon-Camarasa. Breaking Down the Barriers: Investigating Non-Expert User Experiences in Robotic Teleoperation in UK and Japan. *arXiv preprint arXiv:2410.18727*, page 13, 2024. doi: https://doi.org/10.48550/arXiv.2410.18727. URL https://arxiv.org/abs/2410.18727.

[7] Florent P. Audonnet, Ixchel G. Ramirez-Alpizar, and Gerardo Aragon-Camarasa. IMMERTWIN: A Mixed Reality Framework for Enhanced Robotic Arm Teleoperation, September 2024. URL http://arxiv.org/abs/2409.08964. arXiv:2409.08964.

[8] Lipeng Zhuang, Shiyu Fan, Yingdong Ru, Florent Audonnet, Paul Henderson, and Gerardo Aragon-Camarasa. Flat'n'Fold: A Diverse Multi-Modal Dataset for Garment

Perception and Manipulation, September 2024. URL http://arxiv.org/abs/2409.18297. arXiv:2409.18297.

[9] Ana avatar xpriza challenge, 2023. URL https://www.xprize.org/prizes/avatar.

[10] Max Schwarz, Christian Lenz, Raphael Memmesheimer, Bastian Pätzold, Andre Rochow, Michael Schreiber, and Sven Behnke. Robust Immersive Telepresence and Mobile Telemanipulation: NimbRo wins ANA Avatar XPRIZE Finals. Technical Report arXiv:2303.03297, arXiv, March 2023. URL http://arxiv.org/abs/2303.03297. arXiv:2303.03297 [cs] type: article.

[11] Stefano Dafarra, Kourosh Darvish, Riccardo Grieco, Gianluca Milani, Ugo Pattacini, Lorenzo Rapetti, Giulio Romualdi, Mattia Salvi, Alessandro Scalzo, Ines Sorrentino, Davide Tomè, Silvio Traversaro, Enrico Valli, Paolo Maria Viceconte, Giorgio Metta, Marco Maggiali, and Daniele Pucci. iCub3 Avatar System. Technical Report arXiv:2203.06972, arXiv, March 2022. URL http://arxiv.org/abs/2203.06972. arXiv:2203.06972 [cs] type: article.

[12] Daniel J. Rea and Stela H. Seo. Still Not Solved: A Call for Renewed Focus on User-Centered Teleoperation Interfaces. *Frontiers in Robotics and AI*, 9, 2022. ISSN 2296-9144. URL https://www.frontiersin.org/articles/10.3389/frobt.2022.704225.

[13] Takafumi Muto and Joo-Ho Lee. A touch based Multi-Modal interface for teleoperation to improve usability for the non-expert. In *2012 IEEE/SICE International Symposium on System Integration (SII)*, pages 589–594, December 2012. doi: 10.1109/SII.2012.6427343.

[14] Praveen Kumar Reddy Maddikunta. Industry 5.0: A survey on enabling technologies and potential applications. *Journal of Industrial Information Integration*, 2022.

[15] Saeid Nahavandi. Industry 5.0—A Human-Centric Solution. *Sustainability*, 11(16): 4371, January 2019. ISSN 2071-1050. doi: 10.3390/su11164371. URL https://www.mdpi.com/2071-1050/11/16/4371. Number: 16 Publisher: Multidisciplinary Digital Publishing Institute.

[16] Sandra Grabowska, Sebastian Saniuk, and Bożena Gajdzik. Industry 5.0: improving humanization and sustainability of Industry 4.0. *Scientometrics*, April 2022. ISSN 1588-2861. doi: 10.1007/s11192-022-04370-1. URL https://doi.org/10.1007/s11192-022-04370-1.

[17] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of Manufacturing Systems*, 61:

530–535, October 2021. ISSN 02786125. doi: 10.1016/j.jmsy.2021.10.006. URL https://linkinghub.elsevier.com/retrieve/pii/S0278612521002119.

[18] Amr Adel. Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas. *Journal of Cloud Computing*, 11(1):40, 2022.

[19] Michael Grieves. Digital Twin: Manufacturing Excellence through Virtual Factory Replication. Technical report, March 2015.

[20] Tom P Huck, Martin Kaiser, Constantin Cronrath, Bengt Lennartson, Torsten Kroger, and Tamim Asfour. Reinforcement Learning for Safety Testing: Lessons from A Mobile Robot Case Study. *arXiv preprint arXiv:2311.02907*, 2023.

[21] Stefan Scherzinger, Arne Roennau, and Rüdiger Dillmann. Learning Human-Inspired Force Strategies for Robotic Assembly. Technical Report arXiv:2303.12440, arXiv, March 2023. URL http://arxiv.org/abs/2303.12440. arXiv:2303.12440 [cs] type: article.

[22] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-Loop Imitation Learning using Remote Teleoperation. *arXiv:2012.06733 [cs]*, December 2020. URL http://arxiv.org/abs/2012.06733. arXiv: 2012.06733.

[23] Anders Orebäck and Henrik I. Christensen. Evaluation of Architectures for Mobile Robotics. *Autonomous Robots*, 14(1):33–49, January 2003. ISSN 1573-7527. doi: 10.1023/A:1020975419546. URL https://doi.org/10.1023/A:1020975419546.

[24] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*, 3:6, 2009.

[25] Aaron Staranowicz and Gian Luca Mariottini. A survey and comparison of commercial and open-source robotic simulator software. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '11, pages 1–8, New York, NY, USA, May 2011. Association for Computing Machinery. ISBN 978-1-4503-0772-7. doi: 10.1145/2141622.2141689. URL https://doi.org/10.1145/2141622.2141689.

[26] Lenka Pitonakova, Manuel Giuliani, Anthony Pipe, and Alan Winfield. Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators. In Manuel Giuliani, Tareq Assaf, and Maria Elena Giannaccini, editors, *Towards Autonomous Robotic Systems*, pages 357–368, Cham, 2018. Springer International Publishing. ISBN 978-3-319-96728-8.

[27] Angel Ayala, Francisco Cruz, Diego Campos, Rodrigo Rubio, Bruno Fernandes, and Richard Dazeley. A Comparison of Humanoid Robot Simulators: A Quantitative Approach. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 1–6, October 2020. doi: 10.1109/ICDL-EpiRob48136.2020.9278116. URL https://ieeexplore.ieee.org/document/9278116. ISSN: 2161-9484.

[28] Marian Körber, Johann Lange, Stephan Rediske, Simon Steinmann, and Roland Glück. Comparing Popular Simulation Environments in the Scope of Robotics and Reinforcement Learning. *arXiv:2103.04616 [cs]*, March 2021. URL http://arxiv.org/abs/2103.04616. arXiv: 2103.04616.

[29] Tsung-Chi Lin, Achyuthan Unni Krishnan, and Zhi Li. Shared Autonomous Interface for Reducing Physical Effort in Robot Teleoperation via Human Motion Mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9157–9163, May 2020. doi: 10.1109/ICRA40945.2020.9197220. ISSN: 2577-087X.

[30] Adam Pettinger, Cassidy Elliott, Zhiyuan Fan, and Mitchell Pryor. Reducing the Teleoperator's Cognitive Burden for Complex Contact Tasks Using Affordance Primitives. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 6, 2020.

[31] Anthony R. Lanfranco, Andres E. Castellanos, Jaydev P. Desai, and William C. Meyers. Robotic Surgery: A Current Perspective. *Annals of Surgery*, 239(1):14–21, January 2004. ISSN 0003-4932. doi: 10.1097/01.sla.0000103020.19595.7d. URL https://journals.lww.com/00000658-200401000-00003.

[32] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. A Motion Retargeting Method for Effective Mimicry-based Teleoperation of Robot Arms. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 361–370, Vienna Austria, March 2017. ACM. ISBN 978-1-4503-4336-7. doi: 10.1145/2909824.3020254. URL https://dl.acm.org/doi/10.1145/2909824.3020254.

[33] Andrew Lobbezoo, Yanjun Qian, and Hyock-Ju Kwon. Reinforcement Learning for Pick and Place Operations in Robotics: A Survey. *Robotics*, 10(3):105, September 2021. ISSN 2218-6581. doi: 10.3390/robotics10030105. URL https://www.mdpi.com/2218-6581/10/3/105. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[34] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation, January 2024. URL http://arxiv.org/abs/2401.02117. arXiv:2401.02117 [cs].

[35] Frank Joublin, Antonello Ceravola, Pavel Smirnov, Felix Ocker, Joerg Deigmoeller, Anna Belardinelli, Chao Wang, Stephan Hasler, Daniel Tanneberg, and Michael Gienger. CoPAL: Corrective Planning of Robot Actions with Large Language Models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8664–8670, May 2024. doi: 10.1109/ICRA57147.2024.10610434. URL https://ieeexplore.ieee.org/document/10610434.

[36] Hassan Ali, Philipp Allgeuer, and Stefan Wermter. Comparing Apples to Oranges: LLM-powered Multimodal Intention Prediction in an Object Categorization Task, April 2024. URL https://ui.adsabs.harvard.edu/abs/2024arXiv240408424A. Publication Title: arXiv e-prints ADS Bibcode: 2024arXiv240408424A.

[37] Siddarth Jain and Brenna Argall. Grasp Detection for Assistive Robotic Manipulation. *IEEE International Conference on Robotics and Automation : ICRA : [proceedings]. IEEE International Conference on Robotics and Automation*, 2016:2015, May 2016. doi: 10.1109/ICRA.2016.7487348. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC6082626/.

[38] Alberto Gottardi, Stefano Tortora, Elisa Tosello, and Emanuele Menegatti. Shared Control in Robot Teleoperation With Improved Potential Fields. *IEEE Transactions on Human-Machine Systems*, pages 1–13, 2022. ISSN 2168-2305. doi: 10.1109/THMS.2022.3155716. Conference Name: IEEE Transactions on Human-Machine Systems.

[39] Dawei Zhang, Roberto Tron, and Rebecca P. Khurshid. Haptic Feedback Improves Human-Robot Agreement and User Satisfaction in Shared-Autonomy Teleoperation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3306–3312, May 2021. doi: 10.1109/ICRA48506.2021.9560991. ISSN: 2577-087X.

[40] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research*, 37(7):717–742, June 2018. ISSN 0278-3649. doi: 10.1177/0278364918776060. URL https://doi.org/10.1177/0278364918776060. Publisher: SAGE Publications Ltd STM.

[41] Kapil D. Katyal, Christopher Y. Brown, Steven A. Hechtman, Matthew P. Para, Timothy G. McGee, Kevin C. Wolfe, Ryan J. Murphy, Michael D.M. Kutzer, Edward W. Tunstel, Michael P. McLoughlin, and Matthew S. Johannes. Approaches to robotic teleoperation in a disaster scenario: From supervised autonomy to direct control. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1874–1881, Chicago, IL, USA, September 2014. IEEE. ISBN 978-1-4799-6934-0 978-1-4799-6931-9. doi: 10.1109/IROS.2014.6942809. URL http://ieeexplore.ieee.org/document/6942809/.

[42] Yue Peng Toh, Shan Huang, Joy Lin, Maria Bajzek, Garth Zeglin, and Nancy S. Pollard. Dexterous telemanipulation with a multi-touch interface. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 270–277, Osaka, Japan, November 2012. IEEE. ISBN 978-1-4673-1369-8. doi: 10.1109/HUMANOIDS.2012.6651531. URL http://ieeexplore.ieee.org/document/6651531/.

[43] Liangliang Wang, Qiang Li, James Lam, Zheng Wang, and Zhengyou Zhang. Intent inference in shared-control teleoperation system in consideration of user behavior. *Complex & Intelligent Systems*, October 2021. ISSN 2198-6053. doi: 10.1007/s40747-021-00533-4. URL https://doi.org/10.1007/s40747-021-00533-4.

[44] Patrick Naughton, James Seungbum Nam, Andrew Stratton, and Kris Hauser. Integrating Open-World Shared Control in Immersive Avatars. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17807–17813, Yokohama, Japan, May 2024. IEEE. ISBN 9798350384574. doi: 10.1109/ICRA57147.2024.10611618. URL https://ieeexplore.ieee.org/document/10611618/.

[45] Yun-Peng Su, Xiao-Qi Chen, Cong Zhou, Lui Holder Pearson, Christopher G. Pretty, and J. Geoffrey Chase. 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*, 13(22):12129, November 2023. ISSN 2076-3417. doi: 10.3390/app132212129. URL https://www.mdpi.com/2076-3417/13/22/12129.

[46] Yunpeng Su, Xiaoqi Chen, Tony Zhou, Christopher Pretty, and Geoffrey Chase. Mixed reality-integrated 3D/2D vision mapping for intuitive teleoperation of mobile manipulator. *Robotics and Computer-Integrated Manufacturing*, 77:102332, October 2022. ISSN 07365845. doi: 10.1016/j.rcim.2022.102332. URL https://linkinghub.elsevier.com/retrieve/pii/S0736584522000217.

[47] Christoph Bartneck, Tomohiro Suzuki, Takayuki Kanda, and Tatsuya Nomura. The influence of people's culture and prior experiences with Aibo on their attitude towards robots. *AI & SOCIETY*, 21(1):217–230, January 2007. ISSN 1435-5655. doi: 10.1007/s00146-006-0052-7. URL https://doi.org/10.1007/s00146-006-0052-7.

[48] Isaac Sim, December 2019. URL https:developer.nvidia.comisaac-sim.

[49] Ching-An Cheng, Mustafa Mukadam, Jan Issac, Stan Birchfield, Dieter Fox, Byron Boots, and Nathan Ratliff. RMPflow: A Computational Graph for Automatic Motion Policy Generation. In Marco Morales, Lydia Tapia, Gildardo Sánchez-Ante, and Seth Hutchinson, editors, *Algorithmic Foundations of Robotics XIII*, Springer Proceedings

in Advanced Robotics, pages 441–457, Cham, 2020. Springer International Publishing. ISBN 978-3-030-44051-0. doi: 10.1007/978-3-030-44051-0_26.

[50] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. *arXiv:1404.3785 [cs]*, April 2014. URL http://arxiv.org/abs/1404.3785. arXiv: 1404.3785.

[51] Open Robotics. https://ignitionrobotics.org/home, 2022. URL https://ignitionrobotics.org/home.

[52] Webots. http://www.cyberbotics.com, 2022. URL http://www.cyberbotics.com. Open-source Mobile Robot Simulation Software.

[53] NVIDIA Corporation. Nvidia isaac sim, 2022. URL https://developer.nvidia.com/isaac-sim.

[54] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[55] Eric Rohmer, Surya P. N. Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326, 2013.

[56] Jerry Banks. 1999: INTRODUCTION TO SIMULATION. *Winter Simulation Conference*, page 7, 1999.

[57] Yuqian Lu, Chao Liu, Kevin I-Kai Wang, Huiyue Huang, and Xun Xu. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61:101837, February 2020. ISSN 0736-5845. doi: 10.1016/j.rcim.2019.101837. URL https://www.sciencedirect.com/science/article/pii/S0736584519302480.

[58] Pedro Tavares, Joao Silva, Pedro Costa, Germano Veiga, and A. Moreira. Flexible Work Cell Simulator Using Digital Twin Methodology for Highly Complex Systems in Industry 4.0. pages 541–552, November 2018. ISBN 978-3-319-70832-4. doi: 10.1007/978-3-319-70833-1_44.

[59] Stanford Artificial Intelligence Laboratory et al. Robotic operating system, 2018. URL https://www.ros.org.

[60] James Kramer and Matthias Scheutz. Development environments for autonomous mobile robots: A survey. *Autonomous Robots*, 22(2):101–132, January 2007. ISSN 0929-5593, 1573-7527. doi: 10.1007/s10514-006-9013-8. URL http://link.springer.com/10.1007/s10514-006-9013-8.

[61] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, September 2004. doi: 10.1109/IROS.2004.1389727.

[62] Lucas Nogueira. Comparative Analysis Between Gazebo and V-REP Robotic Simulators. December 2014. doi: 10.13140/RG.2.2.18282.36808.

[63] M. Santos Pessoa de Melo, J. Gomes da Silva Neto, P. Jorge Lima da Silva, J. M. X. Natario Teixeira, and V. Teichrieb. Analysis and Comparison of Robotics 3D Simulators. In *2019 21st Symposium on Virtual and Augmented Reality (SVR)*, pages 242–251, October 2019. doi: 10.1109/SVR.2019.00049.

[64] Serena Ivaldi, Jan Peters, Vincent Padois, and Francesco Nori. Tools for simulating humanoid robot dynamics: A survey based on user feedback. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 842–849, November 2014. doi: 10.1109/HUMANOIDS.2014.7041462. URL https://ieeexplore.ieee.org/document/7041462. ISSN: 2164-0580.

[65] Sachin Chitta, Eitan Marder-Eppstein, Wim Meeussen, Vijay Pradeep, Adolfo Rodríguez Tsouroukdissian, Jonathan Bohren, David Coleman, Bence Magyar, Gennaro Raiola, Mathias Lüdtke, and Enrique Fernandez Perdomo. ros_control: A generic and simple control framework for ROS. *Journal of Open Source Software*, 2(20): 456, December 2017. ISSN 2475-9066. doi: 10.21105/joss.00456. URL https://joss.theoj.org/papers/10.21105/joss.00456.

[66] T. Coleman David. Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. 2014. doi: 10.6092/JOSER_2014_05_01_P3. URL https://aisberg.unibg.it//handle/10446/87657. Publisher: Università degli studi di Bergamo.

[67] Peng Chang and Taşkin Padif. Sim2Real2Sim: Bridging the Gap Between Simulation and Real-World in Flexible Object Manipulation. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 56–62, November 2020. doi: 10.1109/IRC.2020.00015.

[68] Nikos Pitsillos, Ameya Pore, Bjorn Sand Jensen, and Gerardo Aragon-Camarasa. Intrinsic Robotic Introspection: Learning Internal States From Neuron Activations. In *2021 IEEE International Conference on Development and Learning (ICDL)*, pages 1–7, Beijing, China, August 2021. IEEE. ISBN 978-1-72816-242-3. doi: 10.1109/ICDL49984.2021.9515672. URL https://ieeexplore.ieee.org/document/9515672/.

[69] Amir Aminzadeh Ghavifekr, Seyedshahab Chehraghi, and Giacomo De Rossi. An Analysis of Sampling Effect on the Absolute Stability of Discrete-time Bilateral Teleoperation Systems. In *2019 IEEE East-West Design & Test Symposium (EWDTS)*, pages 1–6, September 2019. doi: 10.1109/EWDTS.2019.8884389. URL https://ieeexplore.ieee.org/document/8884389. ISSN: 2472-761X.

[70] Nathan D. Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian Motion Policies. Technical Report arXiv:1801.02854, arXiv, July 2018. URL http://arxiv.org/abs/1801.02854. arXiv:1801.02854 [cs] type: article.

[71] Shervin Javdani, Siddhartha Srinivasa, and Andrew Bagnell. Shared Autonomy via Hindsight Optimization. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, July 2015. ISBN 978-0-9923747-1-6. doi: 10.15607/RSS.2015.XI.032. URL http://www.roboticsproceedings.org/rss11/p32.pdf.

[72] Nutan Chen, Keng Peng Tee, Chee-Meng Chew, and Rui Yan. Intuitive interaction for robotic grasping. In *Proceedings of the Workshop at SIGGRAPH Asia*, WASA '12, pages 105–111, New York, NY, USA, November 2012. Association for Computing Machinery. ISBN 978-1-4503-1835-8. doi: 10.1145/2425296.2425315. URL https://doi.org/10.1145/2425296.2425315.

[73] Henny Admon and Siddhartha Srinivasa. Predicting User Intent Through Eye Gaze for Shared Autonom. *2016 AAAI Fall Symposium Series: Shared Autonomy in Research and Practice*, 2016.

[74] Reuben M. Aronson, Thiago Santini, Thomas C. Kübler, Enkelejda Kasneci, Siddhartha Srinivasa, and Henny Admoni. Eye-Hand Behavior in Human-Robot Shared Manipulation. In *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 4–13, March 2018. ISSN: 2167-2148.

[75] Mark Micire, Munjal Desai, Jill L. Drury, Eric McCann, Adam Norton, Katherine M. Tsui, and Holly A. Yanco. Design and validation of two-handed multi-touch tabletop controllers for robot teleoperation. In *Proceedings of the 16th international conference on Intelligent user interfaces*, pages 145–154, Palo Alto CA USA, February 2011. ACM. ISBN 978-1-4503-0419-1. doi: 10.1145/1943403.1943427. URL https://dl.acm.org/doi/10.1145/1943403.1943427.

[76] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. ROBOTURK: A Crowdsourcing Platform for Robotic Skill Learning through Imitation. In *Proceedings of The 2nd Conference on Robot Learning*, pages 879–893. PMLR, October 2018. URL https://proceedings.mlr.press/v87/mandlekar18a.html. ISSN: 2640-3498.

[77] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays. *The International Journal of Robotics Research*, 38(12-13):1513–1526, October 2019. ISSN 0278-3649. doi: 10.1177/0278364919842925. URL https://doi.org/10.1177/0278364919842925. Publisher: SAGE Publications Ltd STM.

[78] Florian Kennel-Maushart, Roi Poranne, and Stelian Coros. Manipulability optimization for multi-arm teleoperation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3956–3962, May 2021. doi: 10.1109/ICRA48506.2021.9561105. ISSN: 2577-087X.

[79] Thorsten Schüppstuhl, Kirsten Tracht, and Annika Raatz, editors. *Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2021*. Springer International Publishing, Cham, 2022. ISBN 978-3-030-74031-3 978-3-030-74032-0. doi: 10.1007/978-3-030-74032-0. URL https://link.springer.com/10.1007/978-3-030-74032-0.

[80] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, May 2022. doi: 10.1126/scirobotics.abm6074. URL https://www.science.org/doi/abs/10.1126/scirobotics.abm6074. Publisher: American Association for the Advancement of Science.

[81] SteamVR on Steam, 2022. URL https://store.steampowered.com/app/250820/SteamVR/.

[82] ROBOTIS e-Manual, 2023. URL https://emanual.robotis.com/docs/en/dxl/mx/mx-28/.

[83] David Whitney, Eric Rosen, Daniel Ullman, Elizabeth Phillips, and Stefanie Tellex. 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)*, pages 1–9, October 2018. doi: 10.1109/IROS.2018.8593513. ISSN: 2153-0866.

[84] Lula Robot Description Editor — Omniverse IsaacSim latest documentation, 2023. URL https://docs.omniverse.nvidia.com/isaacsim/latest/advanced_tutorials/tutorial_motion_generation_robot_description_editor.html.

[85] Max Schwarz and Sven Behnke. Low-Latency Immersive 6D Televisualization with Spherical Rendering. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 320–325, July 2021. doi: 10.1109/HUMANOIDS47582.2021.9555797. ISSN: 2164-0580.

[86] Andrew Hodrien and Terrence Fernando. A review of post-study and post-task subjective questionnaires to guide assessment of system usability. *Journal of Usability Studies*, 16 (3):203–232, May 2021.

[87] John Brooke. SUS – a quick and dirty usability scale. In *Usability Evaluation in Industry*, pages 189–194. January 1996.

[88] Zhan Shi, Yongping Xie, Wei Xue, Yong Chen, Liuliu Fu, and Xiaobo Xu. Smart factory in industry 4.0. *Systems Research and Behavioral Science*, 37 (4):607–617, 2020. doi: https://doi.org/10.1002/sres.2704. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sres.2704.

[89] Benjamin Meindl, Néstor Fabián Ayala, Joana Mendonça, and Alejandro G. Frank. The four smarts of industry 4.0: Evolution of ten years of research and future perspectives. *Technological Forecasting and Social Change*, 168:120784, 2021. ISSN 0040-1625. doi: https://doi.org/10.1016/j.techfore.2021.120784. URL https://www.sciencedirect.com/science/article/pii/S004016252100216X.

[90] Sandra G Hart. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *th ANNUAL MEETING*, 2006.

[91] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, volume 52, pages 139–183. Elsevier, 1988. ISBN 978-0-444-70388-0. doi: 10.1016/S0166-4115(08)62386-9. URL https://linkinghub.elsevier.com/retrieve/pii/S0166411508623869.

[92] Soran Parsa, Horia A. Maior, Alex Reeve Elliott Thumwood, Max L Wilson, Marc Hanheide, and Amir Ghalamzan Esfahani. The Impact of Motion Scaling and Haptic Guidance on Operators' Workload and Performance in Teleoperation. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–7, New Orleans LA USA, April 2022. ACM. ISBN 978-1-4503-9156-6. doi: 10.1145/3491101.3519814. URL https://dl.acm.org/doi/10.1145/3491101.3519814.

[93] Viviana Moya, Emanuel Slawiñski, Vicente Mut, and Eduardo H.M. Couto. Workload detection based on EEG device for teleoperation of a mobile robot. In *2017 XVII Workshop on Information Processing and Control (RPIC)*, pages 1–6, September 2017. doi: 10.23919/RPIC.2017.8211636. URL https://ieeexplore.ieee.org/document/8211636.

[94] Davide Torielli, Leonardo Franco, Maria Pozzi, Luca Muratore, Monica Malvezzi, Nikos Tsagarakis, and Domenico Prattichizzo. Wearable Haptics for a Marionette-inspired Teleoperation of Highly Redundant Robotic Systems. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[95] Nicolai Bechtel, Bernhard Weber, Pascal Severin, Jaime Sancho Aragon, Laurie Van Bogaert, and Michael Panzirsch. 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*, 31(12):663–674, 2023. ISSN 1938-3657. doi: 10.1002/jsid.1262. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jsid.1262. _-eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jsid.1262.

[96] Robert Bogue. The first half century of industrial robot: 50 years of robotic developments. *Industrial Robot: the international journal of robotics research and application*, 50(1):1–10, January 2022. ISSN 0143-991X. doi: 10.1108/IR-10-2022-0251. URL https://doi.org/10.1108/IR-10-2022-0251. Publisher: Emerald Publishing Limited.

[97] Lennart Wachowiak, Oya Celiktutan, Andrew Coles, and Gerard Canal. A Survey of Evaluation Methods and Metrics for Explanations in Human–Robot Interaction (HRI). In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[98] Mika Koverola. General Attitudes Towards Robots Scale (GAToRS): A New Instrument for Social Surveys. *International Journal of Social Robotics*, 2022.

[99] Colleen M. Carpinella, Alisa B. Wyman, Michael A. Perez, and Steven J. Stroessner. The Robotic Social Attributes Scale (RoSAS): Development and Validation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 254–262, Vienna Austria, March 2017. ACM. ISBN 978-1-4503-4336-7. doi: 10.1145/2909824.3020208. URL https://dl.acm.org/doi/10.1145/2909824.3020208.

[100] Dag Sverre Syrdal, Kerstin Dautenhahn, Kheng Lee Koay, and Michael L Walters. The Negative Attitudes towards Robots Scale and Reactions to Robot Behaviour in a Live Human-Robot Interaction Study. page 7, 2009.

[101] Katherine M. Tsui, Munjal Desai, Holly A. Yanco, Henriette Cramer, and Nicander Kemper. Using the "negative attitude toward robots scale" with telepresence robots. In *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop on - PerMIS '10*, page 243, Baltimore, Maryland, 2010. ACM Press. ISBN 978-1-4503-0290-6. doi: 10.1145/2377576.2377621. URL http://dl.acm.org/citation.cfm?doid=2377576.2377621.

[102] S. LaValle. Rapidly-exploring random trees : a new tool for path planning. *The annual research report*, 1998.

[103] Ioan A. Sucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. ISSN 1558-223X. doi: 10.1109/MRA.2012.2205651. URL https://ieeexplore.ieee.org/document/6377468/?arnumber=6377468. Conference Name: IEEE Robotics & Automation Magazine.

[104] robotiq, 2024. URL https://robotiq.com/products/2f85-140-adaptive-robot-gripper.

[105] Sherry Ruan, Jacob O Wobbrock, Kenny Liou, Andrew Ng, and James A Landay. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. 1(4), 2017.

[106] James R. Lewis and Oğuzhan Erdinç. User experience rating scales with 7, 11, or 101 points: does it matter? *J. Usability Studies*, 12(2):73–91, February 2017.

[107] Stanislava Naneva. A Systematic Review of Attitudes, Anxiety, Acceptance, and Trust Towards Social Robots. *International Journal of Social Robotics*, 2020.

[108] Sven Cremer, Lawrence Mastromoro, and Dan O Popa. On the performance of the Baxter research robot. 2016.

[109] Changjoo Nam, Huao Li, Shen Li, Michael Lewis, and Katia Sycara. Trust of Humans in Supervisory Control of Swarm Robots with Varied Levels of Autonomy. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 825–830, October 2018. doi: 10.1109/SMC.2018.00148. URL https://ieeexplore.ieee.org/document/8616144/?arnumber=8616144. ISSN: 2577-1655.

[110] Ayah Hamad and Bochen Jia. How Virtual Reality Technology Has Changed Our Lives: An Overview of the Current and Potential Applications and Limitations. *International Journal of Environmental Research and Public Health*, 19(18):11278, September 2022. ISSN 1661-7827. doi: 10.3390/ijerph191811278. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9517547/.

[111] FRÉDÉRIC KAPLAN. WHO IS AFRAID OF THE HUMANOID? INVESTIGATING CULTURAL DIFFERENCES IN THE ACCEPTANCE OF ROBOTS. *International Journal of Humanoid Robotics*, January 2012. doi: 10.1142/S0219843604000289. URL https://www.worldscientific.com/worldscinet/ijhr. Publisher: World Scientific Publishing Company.

[112] Hirotaka Osawa, Dohjin Miyamoto, Satoshi Hase, Reina Saijo, Kentaro Fukuchi, and Yoichiro Miyake. Visions of Artificial Intelligence and Robots in Science Fiction: a computational analysis. *International Journal of Social Robotics*, 14(10):2123–2133, December 2022. ISSN 1875-4791, 1875-4805. doi: 10.1007/s12369-022-00876-z. URL https://link.springer.com/10.1007/s12369-022-00876-z.

[113] Charles Sun, Abhishek Gupta, Glen Berseth, and Sergey Levine. Fully Autonomous Real-World Reinforcement Learning with Applications to Mobile Manipulation. 2022.

[114] Md Moniruzzaman, Alexander Rassau, Douglas Chai, and Syed Mohammed Shamsul Islam. Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey. *Robotics and Autonomous Systems*, 150:103973, April 2022. ISSN 09218890. doi: 10.1016/j.robot.2021.103973. URL https://linkinghub.elsevier.com/retrieve/pii/S0921889021002414.

[115] Qinglei Zhang, Qinghao Liu, Jianguo Duan, and Jiyun Qin. Research on Teleoperated Virtual Reality Human–Robot Five-Dimensional Collaboration System. *Biomimetics*, 8 (8):605, December 2023. ISSN 2313-7673. doi: 10.3390/biomimetics8080605. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10741399/.

[116] Weiyong Si, Ning Wang, and Chenguang Yang. A review on manipulation skill acquisition through teleoperation-based learning from demonstration. *Cognitive Computation and Systems*, 3(1):1–16, March 2021. ISSN 2517-7567, 2517-7567. doi: 10.1049/ccs2.12005. URL https://onlinelibrary.wiley.com/doi/10.1049/ccs2.12005.

[117] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning Human-to-Humanoid Real-Time Whole-Body Teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8944–8951, October 2024. doi: 10.1109/IROS58592.2024.10801984. URL https://ieeexplore.ieee.org/document/10801984. ISSN: 2153-0866.

[118] Shaodong Li, Peiyuan Gao, and Yongzheng Chen. A Bilateral Teleoperation Strategy Augmented by EMGP-VH for Live-Line Maintenance Robot. *IEEE Transactions on Human-Machine Systems*, 54(4):362–374, August 2024. ISSN 2168-2291, 2168-2305. doi: 10.1109/THMS.2024.3412910. URL https://ieeexplore.ieee.org/document/10572265/.

[119] Aryabrata Basu. A brief chronology of Virtual Reality, November 2019. URL http://arxiv.org/abs/1911.09605. arXiv:1911.09605 [cs].

[120] Abdeldjallil Naceri, Dario Mazzanti, Joao Bimbo, Yonas T. Tefera, Domenico Prattichizzo, Darwin G. Caldwell, Leonardo S. Mattos, and Nikhil Deshpande. The Vicarios Virtual Reality Interface for Remote Robotic Teleoperation: Teleporting for Intuitive Tele-manipulation. *Journal of Intelligent & Robotic Systems*, 101(4):80, April 2021. ISSN 0921-0296, 1573-0409. doi: 10.1007/s10846-021-01311-7. URL https://link.springer.com/10.1007/s10846-021-01311-7.

[121] Enrique Coronado, Shunki Itadera, and Ixchel G. Ramirez-Alpizar. Integrating Virtual, Mixed, and Augmented Reality to Human–Robot Interaction Applications Using Game

Engines: A Brief Review of Accessible Software Tools and Frameworks. *Applied Sciences*, 13(3):1292, January 2023. ISSN 2076-3417. doi: 10.3390/app13031292. URL https://www.mdpi.com/2076-3417/13/3/1292.

[122] Lumen Global Illumination and Reflections in Unreal Engine | Unreal Engine 5.4 Documentation | Epic Developer Community, 2024. URL https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-global-illumination-and-reflections-in-unreal-engine.

[123] VR & Haptic Technology Gloves | SenseGlove | Feel the virtual like it's real, 2023. URL https:www.senseglove.com.

[124] Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, July 1993. ISSN 1050-8414. doi: 10.1207/s15327108ijap0303_3. URL https://doi.org/10.1207/s15327108ijap0303_3.

[125] Justine Rochon, Matthias Gondan, and Meinhard Kieser. To test or not to test: Preliminary assessment of normality when comparing two independent samples. *BMC Medical Research Methodology*, 12:81, 2012. doi: 10.1186/1471-2288-12-81. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3444333/. Publisher: BMC.

[126] Marc-Antoine Moinnereau, Alcyr A. Oliveira, and Tiago H. Falk. Quantifying time perception during virtual reality gameplay using a multimodal biosensor-instrumented headset: a feasibility study. *Frontiers in Neuroergonomics*, 4:1189179, July 2023. ISSN 2673-6195. doi: 10.3389/fnrgo.2023.1189179. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10790866/.

[127] AMRC Factory 2050 The University of Sheffield. AMRC_digital_twin. Technical report, Advanced Manufacturing Research Centre, 2024. URL https://www.amrc.co.uk/files/document/406/1605271035_1604658922_AMRC_Digital_Twin_AW.pdf.

[128] Amazon Robotics Builds Digital Twins of Warehouses with NVIDIA Omniverse and Isaac Sim, 2022. URL https://resources.nvidia.com/en-us-omniverse-enterprise/amazon-robotics.

[129] Tatsuya Nomura. Experiences, knowledge of functions, and social acceptance of robots: an exploratory case study focusing on Japan. *AI & Soc*, 37:367–374, 2022. doi: https://doi.org/10.1007/s00146-021-01196-y.

[130] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, December 2021. ISSN 0001-0782. doi: 10.1145/3503250. URL https://dl.acm.org/doi/10.1145/3503250.

[131] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *arXiv preprint arXiv:2308.04079*, 42(4), 2023.

[132] Jialin Wang, Rongkai Shi, Wenxuan Zheng, Weijie Xie, Dominic Kao, and Hai-Ning Liang. Effect of Frame Rate on User Experience, Performance, and Simulator Sickness in Virtual Reality. *IEEE transactions on visualization and computer graphics*, PP, February 2023. ISSN 1941-0506. doi: 10.1109/TVCG.2023.3247057.

[133] Shehzaad Dhuliawala, Vilém Zouhar, Mennatallah El-Assady, and Mrinmaya Sachan. A Diachronic Perspective on User Trust in AI under Uncertainty. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5567–5580, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.339. URL https://aclanthology.org/2023.emnlp-main.339/.