

École polytechnique de Louvain

Control Interfaces for Human-Robot Collaboration in Extended Reality

Author: **Alicia TORCK**

Supervisors: **Renaud RONSSE, Benoît MACQ, Lotfi EL HAFI, Gustavo Alfonso GARCIA RICARDEZ**

Reader: **Eric PIETTE**

Academic year 2024–2025

Master [120] in Electro-mechanical Engineering

Acknowledgments

I would like to express my endless gratitude to my four supervisors, Prof. Lotfi El Hafi, Prof. Gustavo Alfonso Garcia Ricardez, Prof. Renaud Ronsse and Prof. Benoit Macq, for accepting my application for this master's thesis in the first place, and allowing me to complete my studies by taking part in such an inspiring and incredible research project. I will be forever grateful for the personal growth and professional development I gained from this experience.

I am deeply thankful to Prof. Lotfi El Hafi and Prof. Gustavo Alfonso Garcia Ricardez for their invaluable mentorship and academic guidance, and for giving me helpful advice on self-improvement. I am also profoundly grateful to them for welcoming me so warmly in Japan and for making this three-month internship possible.

I would like to extend my sincere thanks to Carl Törnberg for his generous help throughout the thesis, particularly with Unity and the experiments, and for his unconditional support.

I would also like to thank Anahide Silahli for the unforgettable memories we shared during our time in Japan.

Finally, I want to express my heartfelt gratitude to my family, my loved ones, and my friends, who have consistently encouraged me and helped me believe in myself throughout my entire academic journey.

– Alicia Torck

Abstract

Nowadays, robots and automated machines are deployed in many environments, such as industries and domestic households. Current developments are aligned with initiatives such as Society 5.0 and the Moonshot R&D Program. A complete fusion of the physical and cybernetic spaces, while focusing on human needs and overcoming the physical, spatial and temporal limitations of our bodies, is promoted by these projects to solve our society’s ethical and economic issues. Despite notable progress in service robotics, public acceptance of autonomous robots remains a strong barrier for development, due to their cost, the lack of transparency in their decisions, as well as their unreliability in unpredictable environments.

Human-Robot Collaboration (HRC) through semi-autonomous control provides a middle ground, offering human guidance in tasks of high computational complexity. However, conventional control methods are still often inaccessible to novice users. **Extended Reality (XR)** presents promising tools to augment robots and their surroundings, allowing users to visualise overlaid robots’ information in real-world environments and interact with augmented elements, using an XR headset as one of the possible augmenting devices. Although this technology induced the development of XR-based control methods that have shown promising results in enhancing efficiency in HRC tasks, these new control methods remain unintuitive for non-expert users.

Among the existing XR control devices associated with such interfaces, XR motion controllers and hand-tracked inputs are two common techniques. The XR pen, a pen-like XR controller, is another type of device that has shown great potential in improving the intuitiveness of XR-based control interfaces, however, it has been rarely included in recent comparative studies for robot control in XR.

This thesis presents a novel interface using the XR pen device to control a semi-autonomous mobile robot in **Augmented Reality (AR)**, by selecting navigation goals. When using the interface, the operator points at a position in the room, selects it and drags an augmented arrow (that appeared at the selection position) to the orientation he/she wants the robot to be when it has reached that destination. Additionally, two other interfaces have been developed using the same framework for two other XR control devices: the XR motion controllers and the operator’s hands being tracked as XR controllers.

To assess the performance and users’ perceptions of these XR control interfaces, and of the XR pen in particular, this thesis presents a comparative study in which these three XR-based control interfaces are evaluated against a conventional computer-based control interface, used as a baseline in this study. The results show that the XR pen significantly outperforms the other XR-based and computer-based control devices in task performance, but does not improve the user experience compared to the other devices.

Use of AI Tools in the Redaction of this Thesis

GPT-4 and Grammarly were used during the redaction of this thesis to improve the grammar, the spelling and the academic tone of the text.

Acronyms

AI Artificial Intelligence. 9

AR Augmented Reality. 2, 12, 14

DOF Degrees of Freedom. 31

HRC Human-Robot Collaboration. 2, 12, 70, 73

MRS Multi-Robot System. 73

POV Point of View. 43

ROS Robot Operating System. 12, 37

XR Extended Reality. 2, 10, 37, 70

Contents

1	Introduction	9
2	Related Work	14
2.1	Visual Effects	14
2.1.1	Survey and Taxonomy	14
2.1.2	Communicating Robot’s Motion Intent	15
2.1.3	AR Programming Industrial	16
2.2	Control Methods	18
2.2.1	Target Pointing for Manipulator Arm Robot	18
2.2.2	Selection with XR: Eye Gaze	19
2.2.3	Control Interface for MRS	20
2.3	Introduction of a Pen-Like Control Device	21
2.3.1	Object Selection and Manipulation in VR Headsets	21
2.4	Remaining Challenges	23
3	Proposed Approach	24
3.1	Proposed Approach	24
3.1.1	Control Method	24
3.1.2	System Overview	25
3.2	Comparative Study	26
3.2.1	Evaluation Techniques for User Experience	26
3.2.2	Quantitative Comparative Metrics	28
4	Implementation	31
4.1	Hardware Description	31
4.1.1	Fetch Robot	31
4.1.2	Meta Quest 3 Headset	31
4.1.3	MX-Ink Pen from Logitech	32
4.1.4	Unity	32
4.2	Implementation Details	32
4.2.1	Implementation Overview	32

4.2.2	XR-Based Control Interfaces	34
5	Experiments	37
5.1	Context of the Experiment	37
5.2	Experiment Setup	38
5.2.1	Experiment Procedure	39
5.2.2	Trial Procedure	40
5.2.3	Trial Rules	41
5.3	Measurements	41
5.3.1	Qualitative Measurements	41
5.3.2	Quantitative Measurements	43
5.3.3	Recorded Data	44
6	Results	45
6.1	Participants	45
6.2	Qualitative Results	46
6.2.1	NASA-TLX	46
6.2.2	SUS	48
6.2.3	Global Questions	50
6.3	Quantitative Results	51
6.3.1	Task Completion Time	51
6.3.2	Task Selection Time	53
6.3.3	Task Waiting Time	54
6.3.4	Number of Navigation Requests	55
6.3.5	Mean of the Time Differences Between Navigation Requests	56
7	Discussion	60
7.1	Observations during Experiments	60
7.1.1	Behaviour Towards the Robot	60
7.1.2	Task Efficiency	61
7.1.3	The Control Devices	62
7.1.4	Advice From the Participants	64
7.2	Issues Encountered During the Experiments	64
7.2.1	Hand Gesture: False Positive	64
7.2.2	XR Pen: Disconnection	65
7.2.3	Robot's Path Planning Parameters	66
7.3	Discussion of the Results	67
7.3.1	The Hypotheses	67
7.3.2	User Experience	68
7.3.3	Objective Task Efficiency	69

8 Conclusion	70
A Results	74
A.1 Questionnaires	74
A.1.1 NASA-TLX	74
A.1.2 SUS	76
A.1.3 Global Questionnaire	80
A.2 Raw Quantitative Data	81
A.2.1 Time Duration	81

Chapter 1

Introduction

As our society arises in its cultural and demographic development, technology is one of humanity's ways of solving societal problems.

Since the dawn of civilisation, human societies have evolved from small hunting groups to agricultural communities, and later into industrialised nations driven by mechanised production. The past decades were marked by the transition to an information society, with the emergence of communication technologies and data science, accelerating industrial automation. Experts lately presented a new stage our civilisation is currently moving towards, Society 5.0 ¹. As new societal and economic challenges arise from our current evolution, a complete fusion of our physical space with the cybernetic technologies we created during the previous decades, with the help of advanced technologies such as **Artificial Intelligence (AI)**, would help us face these obstacles for the next 30 years.

Among these societal challenges that still exist, one that was particularly emphasised during the COVID pandemic is our dependence on the physical capabilities of our body to move forward in our lives. Elders, physically disabled people are often not able to function fully on their own, even in their own homes. Aside from that, people work more and longer, reducing the time they can spend at home. As they work more, their rest time at home is particularly important to keep being efficient at work: household chores are a direct obstacle to that. Thus, the general public is always looking for solutions avoiding them to waste time on these daily life annoying tasks. The Moonshot R&D Program ², initiated by the Japanese government, gathers experts to address these types of cultural challenges. Their research is structured around several defined goals, with the first goal focusing on developing an avatar-symbiotic society where humans could live without any physical, spatial and temporal constraints through cybernetic avatars [1].

¹Official website of Society 5.0

²Official website of the Moonshot R&D Program

In the context placed by these two projects, robots appeared as efficient tools in industrial production as well as in service and medical environments [2]. However, their integration in these different sectors varies. In industries, articulated robots, consisting of a manipulator arm with several degrees of freedom, replace human workers on repetitive or physically difficult tasks for reasons of manufacturing efficiency as well as for the worker’s safety [3]. In the medical field, current automated integrations mainly consist of rehabilitation robots, surgical robots as well as telepresence robots. In service environments, nowadays the robots deployed in houses consist of autonomous mobile machines, often vacuum cleaners or lawn-mowers [4].

Roboticists, motivated by the goals presented by Society 5.0 and Moonshot, seek a stronger integration of robots in our homes to free people from their dependence to their body, place and time. However, the deployment of autonomous robots in service environments is highly slowed down by the price of these devices, and by users’ lack of confidence and trust around such mobile devices. Non-expert users often struggle to predict such robots’ intentions and do not feel very comfortable with them around [5]. Moreover, fully autonomous robots often struggle in environments that are too unpredictable for them [6]. In this context, developments are made to ease the integration of service robots.

One solution is to use semi-autonomous robots, partially controlled by an operator, to combine robot and human reasoning to enhance task efficiency and to allow the user to keep control of the robot and gain confidence. This solution is currently deployed in many factories, especially in the automotive industry, as seen in Figure 1.1. Such robots are equipped with safety mechanisms based on force sensors to detect a collision with the human operator. However, these robots are still controlled or programmed using methods requiring some training, not suitable for novice users.

In service robotics, mobile robots such as the Kachaka robot ³ can also be controlled with a tablet, on which you can select targets by clicking on a map of the room (that was previously scanned by the robot). This control type is easily accepted by the general public as touch screens are well integrated in our daily life: most users are well used to them. However, researchers develop new control methods to avoid forcing the user to reduce their field of view by looking at a touch screen, which would be impractical once we are integrating more mobile robots, as in the near future [6].

Another solution permitting the user to not give up his focus on the scene is to directly communicate the robot’s intentions to the user, using XR to overlay in real time the robot’s information onto the real world environment. XR has been proven to enhance the performance and fluidity of human-robot collaborative tasks [7], by

³Official webpage for the Kachaka robot from the company Preferred Robotics.



Figure 1.1: A UR5 from Universal Robotics used as a collaborative robot on production lines in a car factory (reference: [Picture link](#)).

allowing the user to visualise future navigation paths, waypoints and other robot information [8].

As XR headsets are compatible with different types of XR control devices, XR also provides new control methods and control devices. One example of an XR control device is the XR motion controllers, offering various inputs such as buttons and joysticks.

The hands of the user can be tracked by the XR headset, and in this case, the control inputs are the detected hand gestures that can be mapped to trigger an action. For example, in [9], the user performs a "pinch gesture" with his hands to manipulate augmented objects. Such control devices like the XR motion controllers allow the user to manipulate augmented elements and a robot's part [10] and to control the speed of a mobile robot or the joints of an arm manipulator using buttons and joysticks [11]. XR can provide selection-based control methods as well, enabling the user to point at and select target objects to be picked by a manipulator robot [12], or directly select mobile robots to control their motion [6].

However, despite the progress of these studies, several challenges remain concerning the accessibility of such methods for novice users. In the studies that evaluated a hand-ray pointing interface, it was mentioned that the specific hand gesture used to trigger a selection had sometimes been reported as "not very intuitive"[6] and several users mentioned difficulties while performing this gesture [12].

Researchers advise to pursue further investigations concerning control methods and user experience using larger robots, other types of control devices such as 6 DOF controllers, and investigate different types of hand gestures for hand-based control interfaces [6]. Future studies are advised to evaluate their interfaces for

environments and tasks of various difficulties, such as including occluded [6] and distant targets[13].

There is a clear need in the field for comparative studies of XR-based selection devices evaluating users' perceived workload and fatigue, as well as their perceived usability of the developed systems [13]. Several studies have also developed pen-like control devices to enhance selection efficiency [14], and advise to pursue further investigations with such devices [15].

Following these development suggestions, this master's thesis presents a comparison of different XR-based control interfaces to enhance HRC in service environments. Using a computer-based control interface, the 3D visualiser RVIZ for Robot Operating System (ROS) [16], as a baseline, three interfaces were developed for different control devices using the same framework in AR. The three control devices are the XR motion controllers, the operator's hands tracked as XR controllers, and a pen-like XR controller.

The goal of this thesis is to evaluate the three XR-based control interfaces with the baseline interface, and observe if one or many outstand themselves from current developments for their enhanced efficiency and intuitiveness. Following previous studies of the field, such as [15], the pen-like XR controller is expected to yield a better pointing accuracy than the XR motion controllers, improving task efficiency and the user's mental and physical demand.

This master's thesis presents 2 research questions concerning the XR pen objective task performance and user experience:

- **Q1:** "Will the XR pen device be rated as an intuitive, easy to use and learn control method?"
- **Q2:** "Will the XR pen controller improve task efficiency by yielding shorter task completion time, selection time and waiting time?"

The proposed approach of this thesis aims to verify the following hypotheses that were formulated from prior findings in the literature.

Concerning the user's perception of the control interfaces:

- **H1:** "The XR-based control interfaces improve the user experience and decrease the user's workload, compared to the computer-based control interface."
- **H2:** "The control interface based on the XR pen device enhances the user experience and decreases the user's workload, compared to the other XR-based control devices."

Concerning the objective task efficiency of the developed system:

- **H3:** “The XR-based control interfaces enhance the navigation task efficiency compared to the computer-based control interface, by reducing the task selection times, but not for difficult and occluded target locations.”
- **H4:** “The control interface based on the XR pen device improves the navigation task efficiency by yielding shorter task selection times, compared to the other XR-based control interfaces.”

To evaluate the validity of these hypotheses, this study presents a comparative study of the three XR-based interfaces with the baseline computer-based interface. Experiments were conducted with 10 participants who were asked to control a semi-autonomous mobile robot and perform a simple navigation task. Objective task efficiency was measured with conventional metrics used in comparative studies for XR [12], such as the task completion time, from which the task selection time and the task waiting time were isolated. The number of selection requests and the mean of time difference between successive requests were recorded as well. To evaluate the user experience, the NASA-TLX and SUS questionnaires were used [17] [18].

The remainder of the manuscript starts with an overview of the related work in the field of XR interface for robotics. After describing more deeply the remaining challenges identified by researchers, the proposed approach of this thesis is presented, followed by a proposed implementation of this approach. The conducted experiments and their setup details are described in the subsequent chapter, followed by a presentation of the results obtained for the set of participants. To conclude the manuscript, a discussion of the results will then be presented as well as suggestions for further developments for this field of research.

Chapter 2

Related Work

This chapter presents an overview of the recent research made in the field of Augmented Reality for Human-Robot Interaction. The analysed studies are organised in several sections in function of their type of contribution to the research field (visual effects, control methods, or new devices). Afterwards, a summary of the remaining challenges is presented.

2.1 Visual Effects

On the reality-virtuality continuum [19], AR describes “any systems that augment physical objects or surroundings environments in the real world (...)”. A first contribution of AR is a **visual** augmentation of reality: rendering virtual objects on top of reality. Figure 2.1 shows several examples of visual effects made in AR, in the context of robotics. Using specific hardware, AR can be used to render small objects such as targets or path lines, as well as larger effects, like a partial or complete digital twin of a robot, or a virtual section of a room.

2.1.1 Survey and Taxonomy

This first paper [5] is a complete taxonomy of AR technologies and tools used to enhance Human-Robot Interaction.

As shown in Figure 2.2, the different visual effects previously mentioned depend on the hardware and its placement, used to display the AR effects. The most popular AR hardware is a head-mounted display. It consists of a headset put on by the user, which renders AR effects on top of reality using lenses in which the user is directly looking through. This method is classified as an on-body placement, which includes also head-worn projectors and hand-held devices, such as mobile smartphones and tablets, or handheld projectors. The visual effects can also be

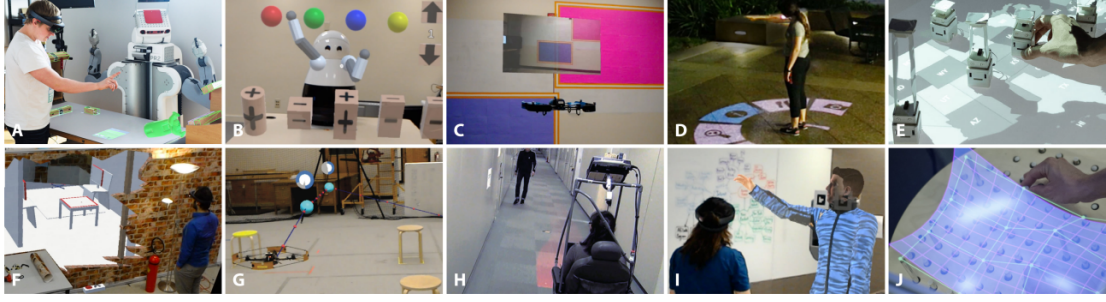


Figure 2.1: Several examples of visual effects in Augmented Reality for Robotics, from [5].

directly displayed in the environment, using see-through windows or projectors strategically placed in a room. These methods are referenced in the on-environment category. The last category of augmented effects is called the On-robot placement, as we can also directly place AR hardware, such as projectors and see-through displays, on the robot itself.

More than just visual effects, AR also allows the user to interact with a robot through augmented virtual elements, with the necessary hardware. As presented in Figure 2.3, user inputs such as the user’s eye-gaze and hand gestures can be tracked and mapped as control requests to interact with a robot. Moreover, most AR headsets are equipped with motion controllers.

AR for HRI is either established or still emerging in several fields such as entertainment, domestic maintenance, industry, and health. In service environments, on-body equipment, such as headsets and mobile devices, are widely adopted, as they are easy to use, require no difficult installation from the user and are compatible with any environment.

2.1.2 Communicating Robot’s Motion Intent

In [8], researchers evaluate how different visual cues added via Augmented Reality improve human-robot interaction. In the context of robot assistance for logistics in warehouses, they recreated a warehouse-like scene where 6 workstations were placed, as seen in Figure 2.5. For their experiment, they asked participants to collect and assemble beads that were placed on each workstation, while an aerial robot is flying in the same scene and occasionally visiting the workstations. If the robot flew to a workstation where the participant was currently working, they had to step out and wait for the robot to perform its task and leave. With this setup, they compared 5 cases: one where the participant was wearing a headset but without any augmented effect, and the other 4 were 4 different types of visually augmented information

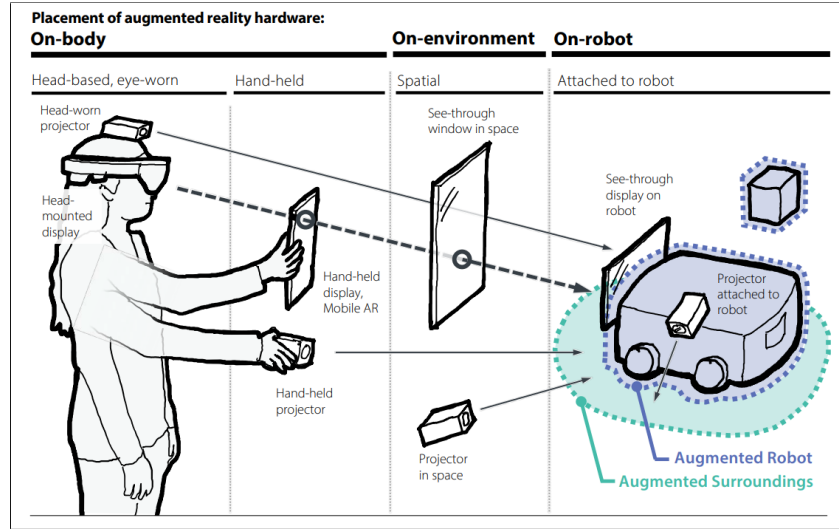


Figure 2.2: Different hardware placement for Augmented Reality in Robotics, from [5].

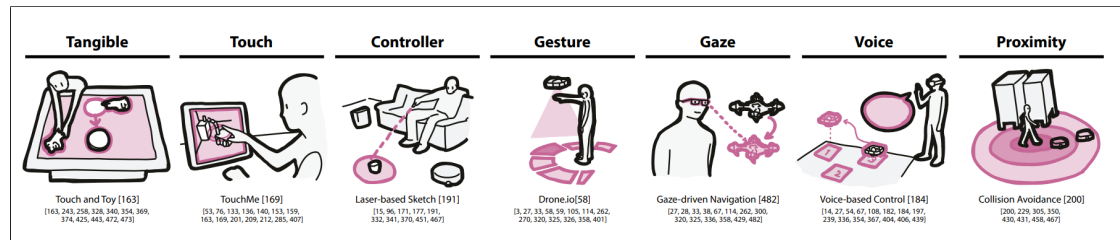


Figure 2.3: A presentation of several types of interaction techniques of Augmented Reality in robotics, from [5].

about the robot navigation, as you can see in Figure 2.4 above. By measuring the participants' waiting time and using 7-point Likert-style questionnaires, experiment conductors demonstrated that showing visual cues of robot motion intent, such as an arrow representing the future path or future waypoints, significantly improves task efficiency and how the users perceived the robot as a work companion.

2.1.3 AR Programming Industrial

This research [14] addresses the lack of intuitiveness and the need for low-level programming skills in conventional techniques of industrial robot programming, such as teaching pendants. Other programming techniques more friendly to novice users do exist, such as the lead-through technique, where the operator controls the manipulator robot's path by holding it with a handle and guiding it in the



Figure 2.4: 4 different AR-based visual effects to improve human-robot collaboration in the case of an aerial robot, proposed by the paper [8].

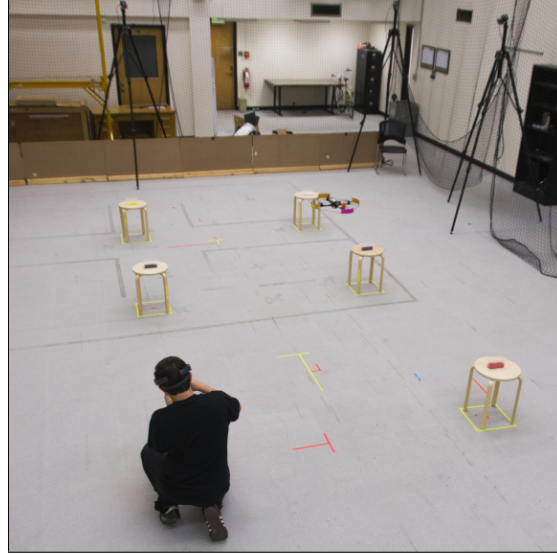


Figure 2.5: Experimental setup proposed by the paper [8].

space. However, this technique is not feasible with large robots frequently employed in factories. These problems refrain small and medium enterprises (SME's) from integrating robots in their manufacturing process, as the efficiency that robots promise is not enough compared to the high cost of employing robots, including material price, a specialized technician's salary and training period, and the time consumed to re-program the industrial robots between tasks. Their proposed approach is to overcome the different enounced limitations by using AR to provide guidance during the programming session. For the path programming method, the paper proposes three methods where the user has to use a wireless handheld pointer to define positions and paths in the manipulator robot's workspace. The three path-programming tools, as shown in Figure 2.6, include one consisting of directly drawing the Cartesian path, one by defining the path using point-to-point intermediate targets, and a last one by selecting the workpiece edges. In the meantime, the operator is wearing an AR headset to visualise the different drawn paths, points, or workpiece edges in the workspace directly.



Figure 2.6: The three interfaces proposed by this paper [14] to program an industrial manipulator robot.

To evaluate the performance of their control interface, they defined two use-case applications to be performed by participants during a user study: a pick-and-place task and a welding task. Each task requires the operator to perform the different control techniques shown in Figure 2.6. During the experiments in which 20 novice participants took part, the user set was separated in two subsets of 10 participants to assign one subset per task. Once a task was performed by a user, a simulation of the robot performing the programmed task was shown to the user for visualisation, before being executed on the real robot. To assess the effect of their implementation on task performance, experiment conductors measured the task completion time of each participant, only including the programming session and motion simulation. They evaluated user experience as well using a custom questionnaire using a 5-point Likert scale. They observed in the results that participants thought the AR-based interface was user-friendly and intuitive, in addition to the robot’s motion simulation improving their confidence.

2.2 Control Methods

2.2.1 Target Pointing for Manipulator Arm Robot

In [12], they also compare the task performance of a hand-tracking-based control method, with this time an eye-tracking-based control. Their context is an arm manipulator cobot, whose purpose is to assist a human worker, equipped with an AR headset, for a simple assembly task. They present 2 control interfaces: a eye-gaze pointing based as seen in Figure 2.7, in which the user stares at the object and uses a pinch hand gesture to select it; and a hand-ray pointing based as shown in Figure 2.8, for which the user also selects with a pinch gesture but points using rays leaving from his hands.

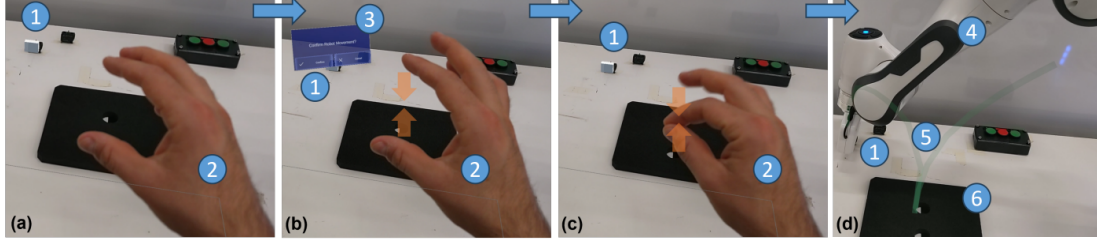


Figure 2.7: XR-based control interface using eye-gaze, proposed by [12].

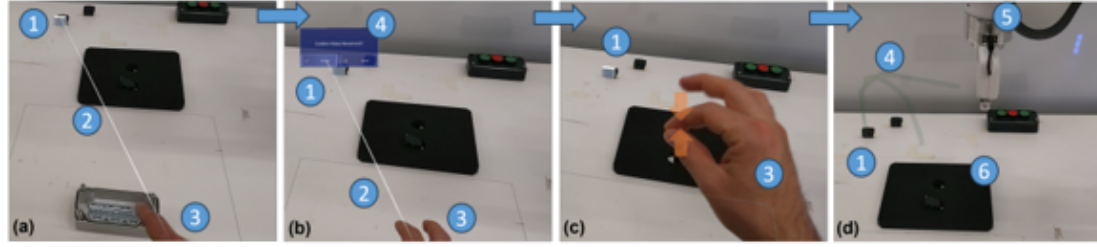


Figure 2.8: XR-based control interface using hand-ray pointing, proposed by [12].

For the experiment, 16 participants had to help an arm manipulator perform a pick and place operation by pointing at the object to be picked by the robot, using one of the two presented interfaces. When picked, the robot would automatically place the object on an assembly space, where the participant would assemble 5 objects together. Recorded task completion and selection times are relatively shorter for the eye-gaze selection interface. Questionnaires showed that the user experience was more consistent and smoother with the eye-gaze pointing interface compared to the hand-ray pointing interface. A SSQ questionnaire was used to detect any motion of sickness issue, and the measured value still stated below the critical threshold. In general, the eye-gaze interface was favoured by the participants and led to better task performance results. However, unlike the previous research, only one participant on this study struggled with the pinch hand gesture.

2.2.2 Selection with XR: Eye Gaze

Similarly, in this paper-[20], eye-gaze tracking is also used for robot control, moreover, as a baseline method. Their study works on suggesting a control interface using a Brain-Computer-Interface (BCI) to select one among different formation types in AR, to control a fleet of mobile robots. The BCI's principle consists of asking a user to stare at one formation icon, placed among others. Like this, the experiment conductors can first record the EEG signals associated to the

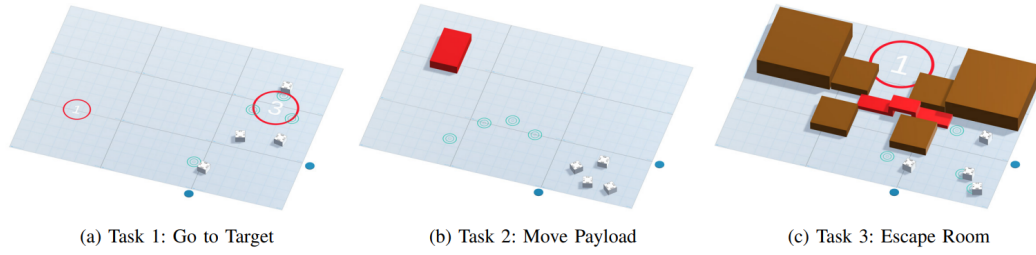


Figure 2.9: The three different tasks used by [6] to evaluate their interface.

eye-tracking signals, later on synchronise both sources and classify the EEG signals with reference to their associated eye-tracking signal. Then they redo another experiment in which they can directly deduce the EEG signals from the eye-tracking information, and execute the selected formation based only on the EEG signals.

The concept of this study essentially emphasises how current robot control methods are not considered efficient enough to one day be transposed to Multi-Robot Systems, and that new control interfaces such as the AR-based BCI are feasible and promising.

2.2.3 Control Interface for MRS

In [6], researchers compared 2 input devices to control a fleet of little mobile robots, hand-tracking with an MR headset or touch gestures on a tablet. Their goal is to evaluate which interface is the most efficient to control a multi-robot system (MRS), using three navigation and collaborative tasks as seen in Figure 2.9: guiding robots to a target, making them move a payload, or making them escape a room by pushing obstacles. They present two different interaction methods: a discrete control for which you trigger actions using a pointing and selection action, and a continuous control that continuously tracks the fingertips position of the user, or a multi-touch action on the tablet, and use them to trigger the closest robots. With the hand-tracking interface, the select input was a pinch hand gesture for the headset and a “tap” on the screen for the tablet. During experiments on a set of 14 participants, task completion times were recorded, but did not highlight any significant improvement with the headset control method. NASA-TLX surveys were used to measure the user’s workload with the different control methods, as well as a post-survey study where users were asked about their source of frustration and interface preferences. Both quantitative and survey results indicated that users clearly prefer the tablet and touch-based actions for simpler tasks, but would perform better and prefer the hand-tracking method for more complex tasks. However, the pinch gesture is frequently mentioned as a high

source of frustration and as not intuitive to trigger, in addition to hand-tracking failures. Researchers recommend deeper studies about convenient and intuitive hand gestures to trigger actions for novice users.

2.3 Introduction of a Pen-Like Control Device

Searching for more intuitive control methods, a pen-like XR controller appeared in many studies as promising in terms of pointing performance and user comfort.

2.3.1 Object Selection and Manipulation in VR Headsets

This paper [13] describes how object selection is a well-established interaction method in Virtual Reality (VR) and presents a complete literature review of the developments made in the context of object selection and manipulation for VR headsets, highlighting trends and research challenges. The latter are organised in 8 main challenges, in which 3 subjects are particularly related to the research context of this master's thesis:

- **Complexity in 3D interaction scenarios:** Need for optimised selection and manipulation techniques for complex VR scenarios (targets far of sight, occluded, ...)
- **Unknown Comparative Usability:** Need for conducting usability studies by comparing different choices of devices
- **Ergonomic Issues:** Limit users' workload and fatigue after periods of extended VR headset use.

For the first research goal, the study describes how the virtual hands and ray casting selection techniques, such as those shown in the papers [12] and [6], appeared at first as simple and intuitive 3D interaction techniques. However, usability studies proved such techniques like hand-ray casting as imprecise and inefficient. This affirmation show how much evaluating interfaces in scenarios of different complexity is very important, as such selection devices might work for simple task but be very inefficient in difficult situations.

For the second challenge concerning ergonomics, this study mentions a paper [15] that conducted a comparative study of the performance of three pointing devices using a headset in VR and AR: VR motion controllers, a 2D mouse, and a pen-like device. This study evaluates whether a custom pen-like pointing device can outperform VR motion controllers, a well-integrated control device in VR applications, and/or a 2D computer mouse, considered as a baseline device with the most promising pointing accuracy. They performed a user study on 12 participants,



Figure 2.10: Experimental setup to evaluate pointing performances of several devices, from [15].

during which participants were asked to select successively 11 targets disposed in a circle on a vertical plane in 3D, in front of them, as shown in Figure 2.10.

To select them, participants had to move a green spherical cursor, fixed on the plane, to one of the targets that was highlighted. Once a target was successfully selected, the trial continued till each of the 11 targets had been selected once. Participants performed this trial 6 times: twice for each pointing device, successively in VR and AR. They measured several objective variables, which were analysed with an ANOVA statistical analysis: the movement time, the error rate, the throughput and the cursor speed. For the subjective measures, they used custom post-study questionnaires based on a 0-100 Likert scale. The study evaluated metrics such as comfort, perceived movement, ease of interaction, and preference. They observed that the pen-like device was comparable to the mouse in terms of performance and was better than the VR controllers. Participants also preferred the pen-like device compared to the VR controllers in terms of comfort and in general. Moreover, trials performed in VR resulted in a better pointing performance than the ones in AR, for which experiment conductors blame the difference in display latencies.

They conclude their paper by supporting the integration of pen-like pointing devices in VR applications, for their accurate distant selection and intuitiveness. However, they explain that their study assessed that they evaluated their interfaces on a small number of participants, and that the display latencies were a strong limitation.

2.4 Remaining Challenges

All these papers clearly highlight how AR enhances user’s understanding of a robot’s intentions and improves global task performance. Several studies have evaluated new AR-based control methods for mobile and manipulator robots, but none of these papers have really highlighted a global solution. 3 AR selection-based control methods were frequently studied: hand-ray casting, eye-gaze tracking, and different custom designs of a pen-like AR device. The use of hand-ray casting, despite being described as a promising hand-free design in this paper [6], comparing it with a touch-screen based control, is highly discouraged compared to eye-tracking in this research [12], in particular for the effect of the device on users’ frustration levels. On the other hand, eye-tracking is frequently not available in recently released devices such as the Meta Quest 3.

Pen-like devices have shown promising results compared to conventional pointing devices, however, the results of the different papers are difficult to compare objectively, as these studies all had their own custom-made prototype device.

The paper [13] mentioned the need of objectively evaluate a comparison of different 3D pointing devices in scenarios of different difficulty, using well-established usability studies.

In addition to that, the study [6] recommends scenarios for AR-based control where targets are occluded behind walls, to force the user to move around and really take advantage of the 3D immersion the headset gives.

In the papers that worked with mobile robots, in many cases, the employed robots are very small devices, with which it’s difficult to have a representative idea of how strong the user’s confidence and trust in robots is. For this exact reason, the paper [6] also suggests to perform user studies with bigger robots.

As a conclusion for this section, since the publication dates of the majority of these papers, new AR headsets have been released. As some of them have mentioned issues related to display latencies and narrow field of view of certain devices, such as the HoloLens, more recent devices with higher graphical capabilities will improve the user experience.

Chapter 3

Proposed Approach

Following the technical limitations of the current developments in the field, the use of a pen-like controller in AR for selection tasks is highly motivated, but hindered by the lack of an official control device being released.

One of the motivations of this thesis is that this limitation could finally be addressed as a commercial XR pen controller had been released a few months before the start of my three-month internship in the context of my thesis. Using a commercial device would make it possible to assess repeatability and allow a fair comparison between the results of several studies evaluating this device.

This master's thesis proposes a new control method based on navigation target selection for semi-autonomous mobile robots in AR, using an XR pen controller. The motivation for this interface is to enhance Human-Robot Interaction with a more intuitive control method and improve navigation efficiency with a more performant target selection method.

The goal of this thesis is to assess the performance and user experience of the XR pen controller by conducting a comparative study with three other control devices, using the same control method for all four devices. The other three devices include a well-established control software in mobile robotics, the VR motion controllers, and the operator's hands.

3.1 Proposed Approach

This section describes the global architecture of the proposed control system.

3.1.1 Control Method

The semi-autonomous mobile robot can only be controlled in navigation and orientation. The XR-based control methods reproduce a similar framework as the

one used in the 3D visualiser software, as it has been developed for intuitiveness and control accuracy.

To navigate the robot with the AR devices, the operator points at targets in the room he is currently using rays coming from the AR device he is using, and selects them by making a selection action. If he holds the select action, the operator can control the robot's orientation by dragging an arrow to make it point in the direction he wants the robot to look. If he releases the select action, the robot present in the room will move to that selected position and turn to the requested orientation. It is the same technique for all AR devices; the only input that varies is the select action used.

The 3D visualiser software works the same way, but instead of pointing at targets in the room, the operator selects a position on a map of the room on a computer, using a mouse input to select the targets and drag the orientation arrow to a certain direction. Similarly, the robot will move to the selected pose once the mouse input is released.

3.1.2 System Overview

The system is constructed as shown in Figure 3.1, and as described in the following paragraph.

For the three XR-based devices, an operator wears an AR headset while being in a room with a mobile robot. Each of the three control devices is tracked by the headset in real time. Once a target in the room is pointed at and the selection action has been activated, the action is detected by the AR application in the headset. The selected position is saved, and the virtual elements associated with the target selection, a virtual target and the orientation arrow, will appear in the operator's point of view at the selected position. Once he releases the selection action, the last orientation of the arrow is saved, and a message containing the complete pose, position and orientation is sent to the network. The network is composed of a laptop that keeps the robot in a controlled state and establishes a connection between the AR application running on the headset and the robot using Wifi. The pose request message sent on the network reaches the robot, which will interpret this message and activate its motors to autonomously navigate, using its various sensors to orient itself to this pose in the room. To use the 3D visualiser software, the operator does not need to wear the headset and must take it off. The software is directly running on the laptop, ensuring the network connection, thus the pose request is just sent from the computer directly to the robot, without any intervention of the headset.

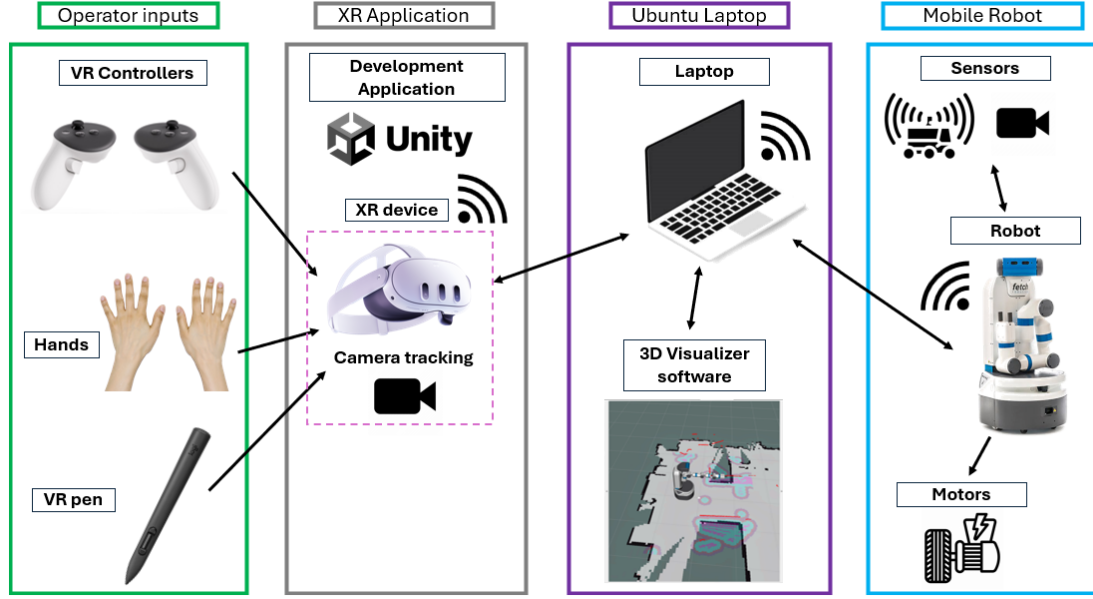


Figure 3.1: System overview of the proposed approach.

3.2 Comparative Study

This section presents the tools selected to compare the four different control devices.

As the goal of this thesis is to compare the four XR-based control interfaces in navigation efficiency and user experience, this master's thesis consequently proposes a user study conducted on a set of participants who were asked to use the different control devices to perform a simple navigation task with a semi-autonomous mobile robot.

3.2.1 Evaluation Techniques for User Experience

An important input of this thesis is to assess if the proposed control interface with the XR pen device will yield a better user experience, especially for intuitiveness, comfort and frustration, among other qualitative metrics.

Similar research studies usually evaluate user experience with conventional questionnaires such as the NASA Task Cognition Load (NASA-TLX) [17], the System Usability Scale (SUS) [18]. It is also frequent to come across papers using custom questionnaires using conventional rating scales like the 5-point Likert-scale [21].

In this study, a NASA-TLX and a SUS questionnaire were used to assess the user experience with each control device, as it was a conventional decision in

comparative studies for AR control methods [12], [6].

The SUS questionnaire was developed as an efficient tool to evaluate both the learnability and usability of a system using only 10 questions, which are translated into a usability score, easy to compute. Moreover, the SUS questionnaire has been proven to give a good assessment of perceived usability even for small samples of participants (around 10 people) as explained in [18], and it is directly our case.

Such questionnaires are also described as a trend for performance evaluation for AR in robotics [5], which will establish the verifiability of our results in the field.

Research Question

Concerning User Experience, the raised question is: **Q1:** “Will the XR pen device be rated as an intuitive, easy to use and learn control method?”

Hypothesis

The control device type is expected to have a significant effect on the user’s workload, especially for the frustration, mental demand and physical demand ratings.

Previous studies have shown that the hands will likely yield high frustration ratings for the NASA-TLX questionnaire, since we are using the same select action as the papers that analysed this control method [12] [6].

The XR pen device has been designed to be lighter than the standard VR motion controllers, thus, it is expected to have a more positive rating for physical demand. However, the XR pen device is expected not to be better rated than the hands, since by definition, they require no handheld device, thus no hand constraint.

The 3D visualiser software is assumed to yield worse mental demand ratings than the XR-based interfaces, since the operator is forced to switch his/her focus between the scene and the computer when he/she wants to take a more difficult navigation decision.

Concerning the SUS questionnaire, the 3D visualiser software is anticipated to yield better ratings than the three other XR-based interfaces for the question “I found the various functions in this system were well integrated.”, since it is a long-established and robust software, commonly used in the robotic community.

The motivation of this thesis is to observe positive ratings from the participants for the XR pen control device, for the following questions:

- “I would imagine that most people would learn to use this system very quickly.”
- “I thought the system was easy to use.”
- “I think that I would like to use this system frequently.”

The last question is especially important for this thesis, as it directly measures the acceptance of the technology by the public.

As a conclusion, the two hypotheses related to the user experience are:

- **H1:** “The XR-based control interfaces improve the user experience and decrease the user’s workload, compared to the computer-based control interface.”
- **H2:** “The control interface based on the XR pen device enhances the user experience and decreases the user’s workload, compared to the other XR-based control devices.”

3.2.2 Quantitative Comparative Metrics

To objectively compare the navigation efficiency between each control device, task completion time, or the time duration of the chosen navigation task is measured during the experiment. This is an objective metric frequently mentioned in the literature, such as in the papers [12] and [6]. Studies comparing selection control methods also go further sometimes by isolating the time duration of pure Human-Robot Interaction from the global task time, such as in this research [12]. Another study evaluating HRI rather measures the time during which the operator was inactive to assess task efficiency, as in this study [8].

In this research, 3 metrics based on task duration were chosen for task efficiency evaluation:

- The task completion time: to use a universal metric for objective comparisons with other studies in the field.
- The task selection time: the time between the start of the task and the moment the first navigation target is selected. This metric’s purpose is to isolate a pure interaction time, during which the autonomous navigation system of the robot has no impact yet.
- The task waiting time: the time necessary for the robot to reach its destination after the last target selection was sent. This metric is used to assess the time spent by the operator waiting for the autonomous system to complete the requested task.

In addition to these metrics commonly used in the literature, two other metrics were recorded:

- The number of navigation requests sent during each trial.
- The mean of the time differences between each navigation request sent.

The purpose is to detect the different navigation strategies used among the participants to complete the task. This way, experimenters can observe if a participant used only one navigation request to take advantage of the autonomous path planning of the robot, or instead tried to stay in control of the robot as much as possible by selecting many intermediate targets to force the robot to go down a path the user had in mind. With the mean of time differences, researchers can observe if participants wasted a lot of time waiting for the robot to move between navigation requests, or rather spam-clicked the control device input, as this would give a numerical explanation of the participants’ perceived frustration and trust in the robot’s actions.

Research Question

Concerning quantitative metrics, the emerging question this thesis states is: **Q2:** “Will the XR pen controller improve task efficiency by yielding shorter task completion time, selection time and waiting time?”

Hypothesis

As the task completion time is directly dependent on the robot’s autonomous navigation efficiency, which is relatively unpredictable and difficult to control for a novice operator, we expect the different control devices to have an effect on this measure, however, it will be difficult to observe due to the strong dependence to autonomous navigation. Similarly, for the task waiting time, we expect both of these metrics to be rather affected by the proximity of obstacles near the target location, and the operator’s understanding of the robot’s safety radius for obstacles.

However, the device type will probably have a strong effect on the task selection time, as this variable is completely isolated from the robot’s autonomous navigation delays.

The XR pen device is expected to yield a shorter selection time compared to other XR-based controllers, based on the results of previous studies on the subject as described in the previous chapter.

Concerning the 3D visualiser software, another comparative study with VR control methods does exist [22], indicating shorter task completion times for VR control interfaces, but this study evaluates a manipulation task which is not representative at all. However since pointing targets on a map is very intuitive and has a strong effect on the operator’s understanding of his surroundings, such a map will also allow the operator to control the robot in occluded environments, which will be difficult with AR control interfaces except if the operator is allowed to move in the scene. With this explanation, the 3D visualiser software can be expected to yield shorter task selection times for difficult or occluded navigation targets.

Aside from the XR pen device, the hands are expected to yield bad performance results, thus longer task times in general, as explained in the previous chapter [12].

This thesis also assumes a middle rating for the VR motion controllers, as they are a commercial control device, and have been designed to be comfortable for long periods of use, but are not strictly designed for pointing accuracy, such as the XR pen device.

Concerning the number of navigation requests sent, this thesis expects the 3D visualiser software to have a really low number, as a map of the room would allow the user to easily select the final destination directly, while centring this target relative to the surrounding obstacles.

All XR-based control interfaces are expected to produce higher numbers of sent requests, as it takes a really short time to select a target with them since it takes only one step (pressing on a button or making a gesture), compared to RVIZ with which you have to press a “confirmation button” first before being able to select a position on a map. It is anticipated that the participants will also most likely “spam-click” the control input for the hand-based interface, as they are expected to get frustrated easily with this control method, as explained in [12].

The mean time difference between navigation requests is expected to be inversely proportional to the number of navigation requests.

As a conclusion, the 2 hypotheses for the quantitative metrics are:

- **H3:** “The XR-based control interfaces enhance the navigation task efficiency compared to the computer-based control interface, by reducing the task selection times, but not for difficult and occluded target locations.”
- **H4:** “The control interface based on the XR pen device improves the navigation task efficiency by yielding shorter task selection times, compared to the other XR-based control interfaces.”

Chapter 4

Implementation

This chapter presents how the proposed approach was implemented. The main hardware elements are described.

4.1 Hardware Description

This section presents a brief description of the hardware used for this thesis.

4.1.1 Fetch Robot

This is the robot that was used for the implementation of the proposed approach, as well as during the experiments. Fetch ¹ is a mobile manipulation robot that operates using ROS.

This robot has a mobile base, an arm manipulator with 7 **Degrees of Freedom (DOF)**, a gripper (open/close), and a 2-DOF head (tilt and span) that can all be controlled semi-autonomously or fully manually, for example with the associated PS3 controller.

4.1.2 Meta Quest 3 Headset

The Meta Quest 3 ² headset was released in 2023. It is a standalone headset, with its own System on Chip, a Snapdragon XR2 Gen 2. It works in passthrough, with pancake lenses. The Quest 3 is praised for its large FOV of 110° (horizontal) x 96° (vertical), its high resolution and graphical capabilities. Many parameters are customizable, such as the inter-pupillary distance, in the range of 53-75 mm. However,

¹Official documentation of the Fetch robot.

²Official webpage for the Meta Quest 3 headset.

no eye tracking is available, and the headset has a battery life of approximately 2 hours.

4.1.3 MX-Ink Pen from Logitech

The MX-ink pen ³ is a pen-like control device in mixed reality, developed for the Meta Quest headset. It is a 6 DOF control device composed of 4 buttons and a pressure-sensitive tip. It can be used in the same way as a XR motion controller, but has been developed to provide a high precision as well as a low latency and ensure intuitive controls.

4.1.4 Unity

Unity ⁴ is a game development platform, allowing developers to create XR projects, while providing all the tools required to build and deploy the XR applications on an XR headset.

4.2 Implementation Details

Below is an outline of the implementation details for the proposed interface.

4.2.1 Implementation Overview

The XR application manages all the XR visual effects and XR control interactions running in the XR headset, organised in a project containing all the necessary scripts. It was an existing XR project called developed by the fellow M2 student Carl Törnberg, in which several scripts were integrated in the context of this master's thesis.

A total of 8 scripts were developed for this master's thesis, however 3 of them implement the control interfaces described in the proposed approach.

The 3 scripts consist of:

- A Topic publisher script: Allows sending ROS messages, allowing the XR application to send navigation requests to the robot via the network.
- A device handler wrapper script: Manages the connections/disconnections of the different XR devices used with the headset.

³Official Logitech webpage for the MX-Ink pen.

⁴Official Unity website.

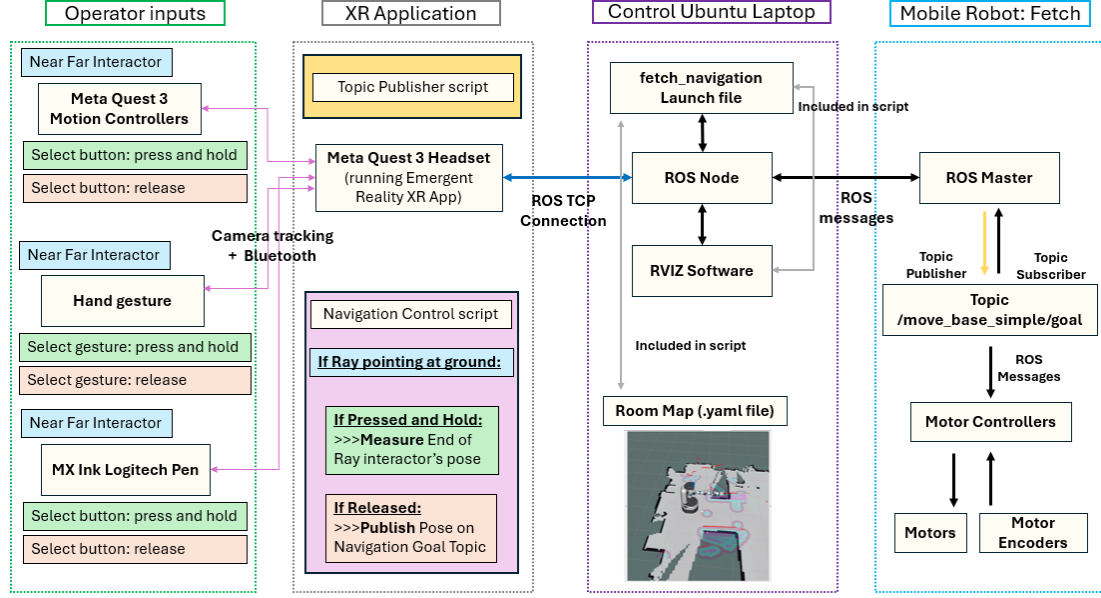


Figure 4.1: Overview diagram of the implementation details for the proposed approach.

- A Navigation Control script: Detects XR control devices' inputs to select navigation poses, to be later sent to the robot.

The Navigation Control script first detects if the connected XR control device points towards the ground (using a Near Far Interactor function that manages raycast-based and direct interactions with virtual objects). If it does, the script checks next if the input of the control device assigned to trigger the selection is activated. If it is, the script saves the 2D Cartesian position of the point at the interaction between the ground and the raycast coming from the control device in a vector, with reference to the robot's reference frame. If the input is held, the script saves the orientation of the arrow in a quaternion, with reference to the robot's reference frame. If the input is released, the position vector and orientation quaternion are combined as a navigation goal in a ROS message, which is published from the headset to a ROS topic controlling the navigation of the robot, through the network.

The XR application running in the headset is connected to the network through a ROS TCP connection ⁵ (an adapter developed to establish a connection between a Unity application and a ROS node, enabling the Unity application to keep using the ROS communication conventions). The ROS node is running on an Ubuntu

⁵ROS-TCP-Connector

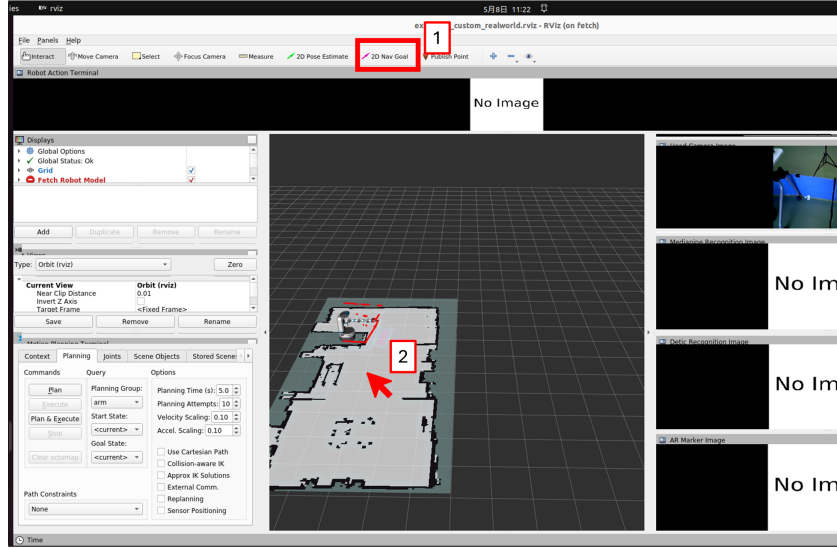


Figure 4.2: Screenshot of the RVIZ software when a target is selected, making the orientation arrow appear (picture shown to the participants during a tutorial video).

laptop, from which we can run the different launch scripts of the Fetch robot, which is connected to the ROS Network as the ROS Master. When the Navigation goal message is sent on the ROS network, it is published on the internal ROS topic of Fetch “/move_base_simple/goal”, which controls the navigation system by sending a target pose (a position and orientation) to the robot, which will manage its motor controllers accordingly to reach this pose.

When the operator uses the baseline interface RVIZ (3D visualiser software), the software is directly running on the Ubuntu laptop ensuring the ROS network connection. After pressing a button that activates the navigation control function of the software, the operator selects a position and orientation on a map, which are combined and sent in a ROS message on the same ROS topic as for the XR interfaces, as shown in Figure 4.2.

4.2.2 XR-Based Control Interfaces

In Figures 4.3, 4.4, and 4.5, the point of view of the operator wearing the headset and using the different XR-based control interfaces is shown.

The control interface based on the XR motion controllers uses the primary button of each motion controller as the activation input. Both motion controllers can trigger the navigation goal selection, but only one at a time, as shown in Figure 4.3.

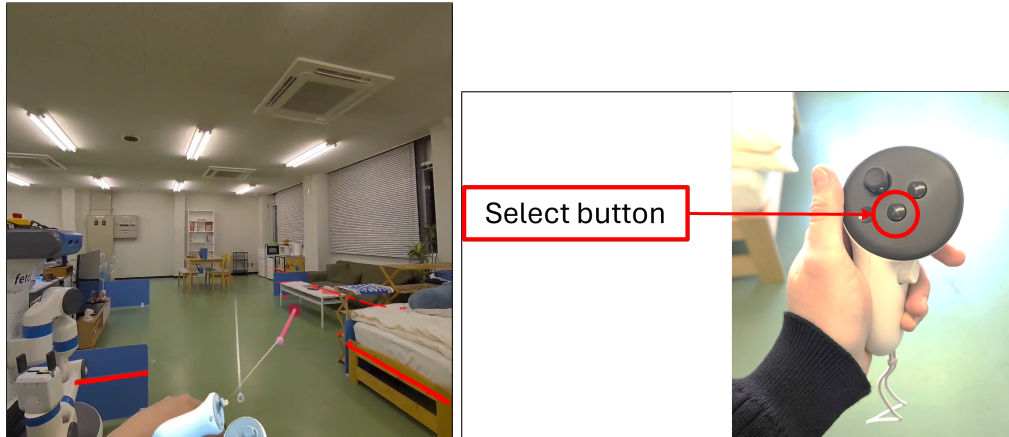


Figure 4.3: XR-based control interface using the XR motion controllers.



Figure 4.4: XR-based control interface using the operator's hands.

The control interface based on the operator's hands uses a specific hand gesture called a "pinch gesture" (thumb and index fingers must enter in contact for the gesture to be detected) that you can see in Figure 4.4 to trigger the navigation goal selection. As for the motion controllers, both hands can trigger the selection, but one at a time.

The control interface based on the XR pen uses one of the buttons on its side to trigger the input. Only one pen was used for this thesis. However, as for the motion controllers, a second XR pen device could be connected.

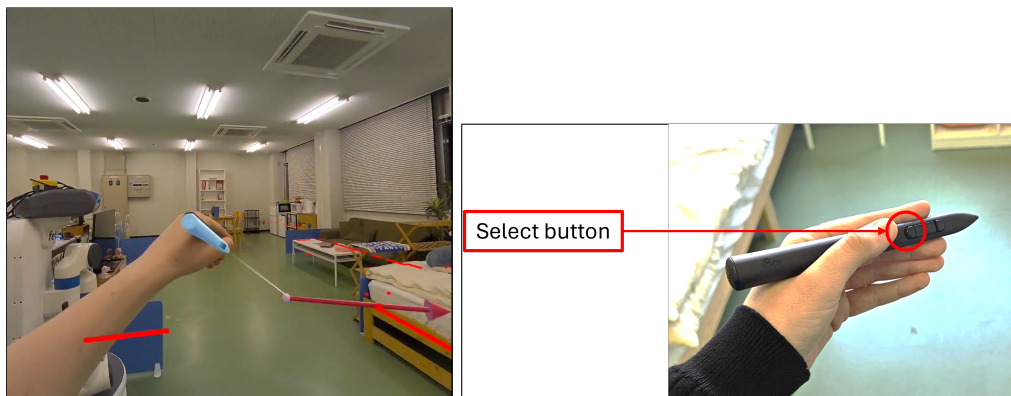


Figure 4.5: XR-based control interface using the XR pen.

Chapter 5

Experiments

This chapter presents the experiments that were conducted in order to evaluate the proposed approach of this research. In the following sections, the context of the experiments, the setup, and the evaluation metrics used are described.

5.1 Context of the Experiment

In this research, we are interested in the impact of service robotics on societal challenges. Robots could help people for household chores for example, but also to allow physically-disabled people being less dependent of their body capabilities. Robots are particularly useful in dangerous tasks, and in service robotics such a task could simply be taking food that is too hot to be handled by a human, or in the context of medical field, transporting sterile objects.

The proposed application and its XR-based design are presented as a solution for service robotics deployment, currently restrained by the control methods still being too complex for non-expert users, or users not yet able to trust a fully autonomous device running in their houses, without understanding what it does. Three XR-based interfaces were developed as a proposed approach to enhance service robotics deployment by presenting a more intuitive control method. To evaluate their performance, we compared these control methods with a reference. In our context of service robotics and semi-autonomous robots, a good candidate is the 3D visualiser RVIZ for ROS, commonly used in robotics. The latter permits to select a navigation goal on a map of the room the robot is currently in. When selecting the navigation goal, the system allows the user to move an arrow to control the robot's orientation at its destination.

In our experiments, the robot task is to grab a can of coffee, which is too hot to be handled by a human, and to transport it from one place in the apartment to a user sitting somewhere in the apartment, unable to move.



Figure 5.1: Experiment scene: a reproduction of a daily living environment.



Figure 5.2: Third object location: the kitchen table. Figure 5.3: Third object location: the living room shelf. Figure 5.4: Third object location: the bed table.

5.2 Experiment Setup

For these experiments, the participants had to operate a semi-autonomous mobile robot. Their goal was to guide the robot to a given location in the simulated apartment, where a can of coffee was placed.

The participants were not asked to perform the robot's arm manipulation to grab the item, they only had to guide the robot to the furniture where the object was located using the presented interfaces. The task was completed once the robot had reached a position close enough to the object location in order to be able to grab the object using its arm. The participants also had to control the orientation of the robot, and were asked to make the robot face the object at its destination. To introduce different levels of difficulty, we used three different object locations, one in each room of the home-like environment, as shown in Figure 5.2, Figure 5.3, and Figure 5.4.

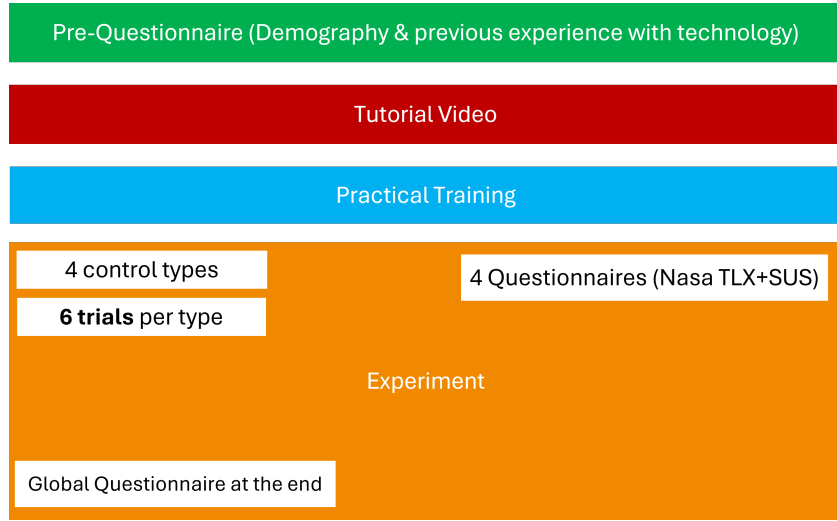


Figure 5.5: Different stages of the experiment.

5.2.1 Experiment Procedure

This section describes the different stages of the experiments, following the process shown in Figure 5.5.

Each participant started the procedure by completing a demographic questionnaire, as well as several questions about their previous experience with XR modalities and robots. Once this was done, participants were shown a tutorial video. This 10-minute video described what was the context of the experiment, the different devices involved, and the whole trial procedure. Each navigation control interface was also briefly described, while insisting on the correct gestures to perform and the input buttons to select.

When the video ended, participants could ask a first round of questions before moving on to a practical session of 20 minutes. During this time, participants were free to test the different control interfaces and see how the robot would react to them, spending as much time as they needed to feel comfortable with the devices. In parallel, the experiment conductors would repeat all the necessary information that was explained in the video concerning the experiment rules for example.

Once they were ready, we would start the experiments. The experiment went as shown in Figure 5.6: each participant had to control the robot using the four different interfaces one by one. For each of them, the participant had to navigate the robot to three different locations. The participant had to go through two trials for each of these object locations: one during which the participant had to stay at his seat, and during the other one, he could freely move in the room. Thus, for each device, the participant had to go through six different trials. Initially,

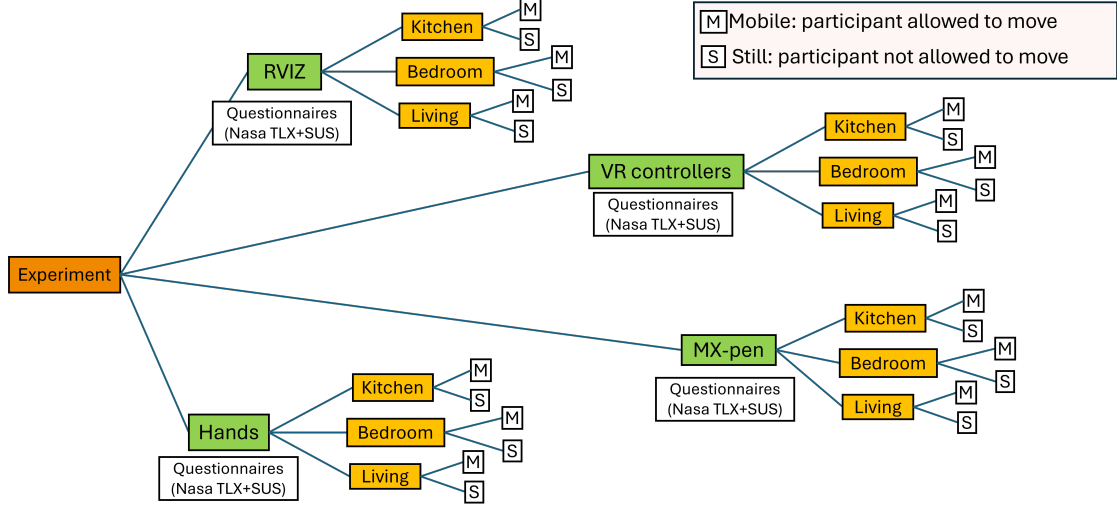


Figure 5.6: Visual representation of all the trial types tested with each participant.

every trial type was planned to be performed four times to evaluate the interface repeatability and robustness. But due to the experiment duration, it was reduced to two and later to only one trial per trial type, as with only one trial the complete experiment for one participant already lasted around two hours.

Once each device was completely tested, the participant completed one NASA-TLX [17] and one SUS [18] questionnaire about his/her experience using the tested device. After testing all four devices, the participant finally had to complete four questions about his general impression of the whole experiment.

5.2.2 Trial Procedure

Each trial started this way. The participant was asked to place himself at the operator desk in the bedroom of the simulated apartment. If the robot wasn't at its initial position, next to the operator's desk, it would be reset. Then, if everyone was ready, the experiment conductors would tell all the necessary parameters to the participant. If it was the first trial for one device, the experiment conductor would tell the participant which controller interface to use, and if it was an XR-based interface, they were asked to put on the headset. Then, and for every trial, the experiment conductor would tell the user which object location he had to navigate the robot to, in addition to whether the participant could move from his seat or not. Once this was said, a countdown was started, and at the end of it, the participant was free to control the robot using the assigned control interface. Once the robot had reached the selected destination pointed by the participant, meaning that the robot had completely stopped moving near the object location and that

the participant was satisfied with its pose, he had to shout “done”, to notify the experiment conductors: at the same time, they would record all quantitative data of the trial as well as sending a reset request to the robot. The participant then had to go back to his seat and wait for the parameters of the next trial.

5.2.3 Trial Rules

Each trial had a time limit of two minutes. If at any point there were any crash or bug during the experiment, we would restart the ongoing trial and possibly restart the control devices if it was necessary. At no point during the experiments were the participants told what to do. For example, they had to decide for themselves if the final destination they selected for the robot was valid and sufficient. It was only based on their own intuition. In the same process of thoughts, if at some point the robot would get stuck near obstacles, waiting to autonomously find a path around them, the participant was free to either wait for the robot to un-stuck itself, or re-select new positions in the room that were easier to reach. Once the trial parameters were assigned, at no point during the experiments would a participant not respect them. Moreover, participants were not allowed to leave the experimental scene during the trials.

5.3 Measurements

This section describes all measurements that were made during the experiment.

5.3.1 Qualitative Measurements

Before, during and after the experiment, the participant had to fill out various questionnaires: a demographic survey and several questions about their previous experience with the related technologies, both a NASA-TLX [17] and a SUS [18] questionnaire after each device had been tested, and several small questions at the end of the whole experiment.

Demographic Survey

The participants were asked several questions about personal information and prior experience to assess if individual differences between participants could influence the outcome of the experiment. These questions were written using questions conventionally asked in user studies (age, gender, etc), to which were added several questions taken from [23]. The personal questions asked were:

- the participant’s age

- the participant’s gender
- the participant’s vision correction type
- the participant’s country of origin
- the participant’s handedness
- if the participant had any encumbered movement of the hands
- if the participant had any neurological condition

Prior Experience with XR and Robots

As mentioned in the previous paragraph, the participant also answered questions about their prior experience with the technologies related to the study, using a 5-point Likert scale [21], going from strongly disagree to Strongly agree. They had one question asking “I am NOT familiar with robots”.

The rest was four questions about prior experience with XR modalities following this format: “I am familiar with [-]”, in which the free space was replaced by four modalities: Virtual Reality (VR), Augmented Reality (AR), Augmented Virtuality (AV), and Extended Reality (XR). Half of the questions were negated to remove the acquiescence bias.

These four questions about XR modalities were not used in this master’s thesis but were asked to the participants for Carl Törnberg’s research, who was recording the answers to these four questions, as well as an additional question about the participants’ preferred XR modality for his own experiments.

NASA-TLX

A NASA-TLX [17] questionnaire was used to evaluate the participants’ task load index with each control interface.

SUS

A SUS [18] questionnaire was also used to assess the quality and intuitiveness, and comfort of each control interface from the participant’s point of view. Half of the questions were also negated to remove the acquiescence bias.

Questions For User’s Perception of HRC

At the end of the experiment, the participants were asked four questions about their overall impressions of the human-robot interaction during the whole experiment.

The questions were designed based on this research [24]. The questions were answered using a 5-point Likert scale [21], and half of them were negated as well. These were the questions asked:

- “My interaction with the robot felt fluent.”
- “I did not trust the robot to do the right thing at the right time.”
- “The robot’s intentions were predictable.”
- “I did not feel safe.”

5.3.2 Quantitative Measurements

During each trial, the following variables were recorded:

- **Start time:** after a countdown, the experiment conductors manually sent a command to start all data recordings. The time stamp of the moment the command was sent was saved under this variable.
- **Finish time:** as soon as the participant notified the experiment conductors that he finished the task, a command to stop the timer was manually sent, saving the last time stamp and all recorded data.
- **Navigation request time:** each time a new navigation goal was requested, the time stamp was recovered.
- **Navigation goal requests from the participant’s Point of View (POV):** each time a navigation goal request was sent, the complete pose (position vector and orientation quaternion, defined from the participant’s reference frame) was recorded. All navigation pose requests were stored in a list, thus, the number of sent requests was saved as well.
- **Navigation goal requests:** similarly to the previous measure, the navigation goal requests were recorded in a list, but this time the poses were defined from the robot’s reference frame.
- **Object location:** the name and the pose of the target object in the robot’s coordinate frame were saved.
- **Baselink Final pose:** the final pose of the robot (the one at the time the “finish” request was sent) was saved.

All these data were automatically saved in a .txt file in the JSON format, with the trial’s information (trial id, object location, trial type, controller type) as well as the participant’s ID.

Similarly to the paper[12], the task completion time (finish time - start time) was recorded to detect any improvement in task efficiency between the four compared interfaces. Its purpose was also to highlight if one of the trial’s cases, for example, one object location in particular, was more difficult to achieve than the others. However, to objectively compare the four interfaces, a task selection time and a task waiting time were isolated from the global task duration, using the navigation request time measures. Because the employed robot could sometimes get stuck around obstacles due to its autonomous path planning, the task selection time was defined to exclude the navigation-only period of the trial (navigation request time - start time), which could not directly be controlled by the participant. That way, for the very first trial of each participant for each device, the time necessary to select only a pose could be compared, which could be a measure of intuitiveness.

During each trial, the navigation requests sent were recorded, from which the number of navigation requests and the time between successive requests were deduced. This information was useful to detect:

- If a participant was frustrated and selected multiple poses in a short period (spam-click)
- If a participant struggled to make the robot reach a destination and use multiple navigation requests to force the robot down a certain path.
- If a participant used way points or only pointed at one destination next to the object location.

5.3.3 Recorded Data

In addition to the measurements described in the previous sections, further recordings were made during the experiment. For each participant, each trial was recorded with an external camera pointing at the scene.

During each trial, a timer was set to stop the trial in case it took too much time, but it was not recorded since the time was also saved in the control application internally, and stopped at the same time as the timer.

The anonymous and handwritten answers of the questionnaires were transcribed into an Excel file before being processed. For the quantitative measurements, a Python code was used to extract them from the .txt files and process them.

Chapter 6

Results

This chapter presents the results of the experiments realised in the context of this master’s thesis. Starting by outlining the profile of the study population, qualitative results of the employed questionnaires are reported. Following this section, the quantitative data recorded during the trials are later detailed.

In this chapter, box plots are frequently used to graphically display the results of the experiments. Box plots are statistical charts used to visualise how numerical data is spread within a sample. In the graph, the lines that extend from the box are called the whiskers. Both ends represent the minimum and maximum values of the dataset, respectively. The box is the interquartile range (IQR), representing which interval of values half of the dataset is situated.

6.1 Participants

The study was conducted on ten participants. Two of them were female, the others male. The participants were between 22 and 61 years old, with a mean at 33 years old and a standard deviation of 14 years. All participants were right-handed, all without any encumbered movement on their hands. Two participants out of ten had normal vision: one of the eight others wore contact lenses, while all the others had glasses. All participants were free of neurological conditions.

The subject group was composed of domain experts as well as novice users, coming from various countries around the world. In the set, 3 participants were Japanese, 3 others were Belgian, and the others came from Mexico, India, France as well as Poland. When asked if they were familiar with robots, participants had a mean of 4.2 out of 5. Concerning XR modalities, participants had, on average, little or neutral familiarity with the different XR modalities.

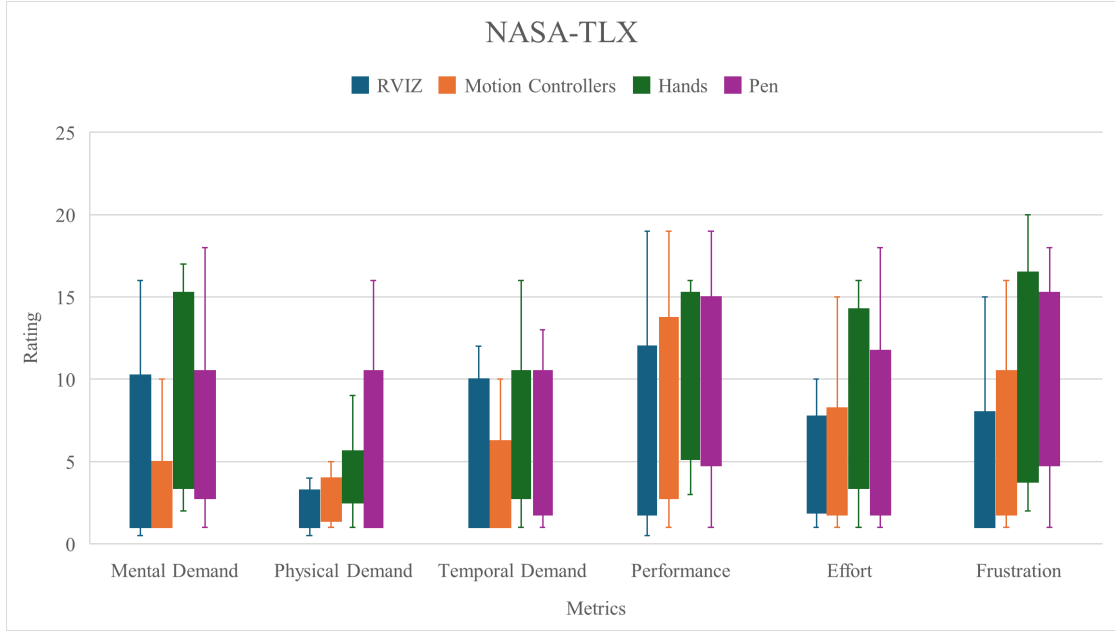


Figure 6.1: Box plot of the TLX ratings and their standard deviations, for the different devices, rated on a scale from 1-20.

6.2 Qualitative Results

This section presents the results of the NASA TLX and SUS questionnaires, given to the participants after each device had been tested.

6.2.1 NASA-TLX

This section contains the results for the NASA-TLX questionnaires.

Figure 6.1 is a detailed breakdown of the ratings of the 6 TLX factors for each control device.

A single-factor ANOVA analysis was performed to evaluate if the type of device had a significant effect on the TLX workload, separated into its different dimensions. Table 6.1 is a summary of the p-values for each dimension. We can observe that all p-values are relatively high, except for the frustration dimension, which presents a rounded p-value of 0.1, which reflects a moderate statistical difference between control devices.

If we observe Figure 6.1, we can notice that the hands and the Pen have significantly higher frustration ratings than RVIZ and the motion controllers: the hands presented the highest frustration rating, with an average of 10.6/20. RVIZ presents the smallest frustration workload, with an average of 5.1/20.

Variable	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
p-value	0.2392	0.144	0.5045	0.7089	0.1533	0.099

Table 6.1: p-values of the different NASA-TLX loads in function.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	5.1	6.25	10.6	9.9
Variance	23.211	27.181	44.933	33.878

Table 6.2: Average and Variance of the NASA-TLX ratings for the frustration, for each control device, obtained as part of the single-factor ANOVA analysis.

However, the ratings of the participants have a high variability, as the variance for the hands frustration ratings is at 44.933 (thus a standard deviation of 6.703), for example, as shown in 6.2.

For the mental demand, Table 6.3 shows that the hands also have a higher rating, with an average rating of 7.65, but a variance of 35.114. However, the motion controllers have the smallest average of 3.35, with a very low variance of 8.003.

For the physical demand, there is also a significant difference between the pen and the other devices, both in the plot and the numerical values. The motion controllers have the smallest average of 2.55 and the lowest variance of 1.914. The pen has an average of 5.8, which is the highest, but yields a high variance of 28.178 as noted in Table 6.4.

The temporal demand is more dense, but we can clearly notice that the motion controllers get the smallest average rating of all controllers (average: 4.45, variance 9.581 in Table 6.5).

The performance had the highest p-value with the ANOVA test, presenting no statistical difference between the control devices, which is observable in the box plot chart as all four boxes and whiskers are elongated along the axis, translating in a high variance in the ratings of the participants, as shown in Table 6.6 (minimum variance: 28.003, maximum: 40.803).

The effort dimension presents a noticeably lower rating for RVIZ (average: 4.85) and the motion controllers (average: 5.6) compared to the hands and the pen (however, they yield a high variance of respectively 30.003 and 32.322 in Table 6.7).

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	5.65	3.35	7.65	7
Variance	26.558	8.003	35.114	28.667

Table 6.3: Average and Variance of the NASA-TLX ratings for the mental demand, for each control device, obtained as part of the single-factor ANOVA analysis.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	2.65	2.55	3.95	5.8
Variance	12.003	1.914	5.803	28.178

Table 6.4: Average and Variance of the NASA-TLX ratings for the physical demand, for each control device, obtained as part of the single-factor ANOVA analysis.

Unweighted Workload Index

The questionnaires were unweighted as we considered each dimension of perceived workload to be equally important. In that particular case, the Raw Task Load Index (Raw-TLX, for unweighted factors) is computed as the average of the 6 ratings, multiplied by 5 to obtain a score rated on 100. A single-factor ANOVA test was also performed on these results, which led to a p-value of 0.129, indicating that the type of control device used during a session had a moderate effect on the user’s workload.

Table 6.8 presents the average and variance values of the raw workload indexes for the four control devices.

We can observe that the motion controllers obtained the smallest workload rating of 25.542/100, with the lowest variance among all four devices. RVIZ is in close second place, however, with a variance almost twice as big (variance: 331.798 compared to 171.298 for the motion controllers).

The hands have the worst average rating of 40.25, followed closely by the pen (average: 39.583), however, the latter has the highest variance, with a value of 488.619.

6.2.2 SUS

This section presents the results of the SUS questionnaires.

Based on the ratings from 1-5 of the 10 questions asked in the SUS questionnaire,

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	4.9	4.45	7.1	6.3
Variance	19.656	9.581	24.322	22.233

Table 6.5: Average and Variance of the NASA-TLX ratings for the temporal demand, for each control device, obtained as part of the single-factor ANOVA analysis.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	7.55	8.45	9.65	10.4
Variance	40.803	36.581	28.003	32.044

Table 6.6: Average and Variance of the NASA-TLX ratings for the performance, for each control device, obtained as part of the single-factor ANOVA analysis.

the Usability score is computed as shown in this equation 6.1, following the scoring standards of the SUS questionnaire as described in [18].

$$\text{SUS} = \left[(Q_1 - 1) + (5 - Q_2) + (Q_3 - 1) + (5 - Q_4) + (Q_5 - 1) + (5 - Q_6) + (Q_7 - 1) + (5 - Q_8) + (Q_9 - 1) + (5 - Q_{10}) \right] \times 2.5 \quad (6.1)$$

This equation is computed for each participant and each device, as the participants completed one SUS questionnaire as soon as they finished testing one device. The Usability score of each device was then computed as the average of the scores from the 10 participants.

The obtained score rated on 100, which should not be interpreted as a percentage but as a percentile, is evaluated based on the scale shown in Figure 6.2.

The SUS scores of each control device obtained during the experiments are shown in Figure 6.3. A single-factor ANOVA analysis has also been applied to these results, which resulted in a p-value of 0.12, taking the type of control devices as the fixed variable. This value reflects that there is a moderate difference between the control devices in the user's perceived usability of the system.

In Table 6.9, the average and variance values of the obtained SUS scores are shown. The motion controllers have the highest average SUS score of 77, significantly outperforming the second-ranked device which is RVIZ, with an average of 68.5.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	4.85	5.6	9.35	8.1
Variance	10.669	22.267	30.003	32.322

Table 6.7: Average and Variance of the NASA-TLX ratings for the effort, for each control device, obtained as part of the single-factor ANOVA analysis.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	25.583	25.542	40.25	39.583
Variance	331.798	171.298	373.773	488.619

Table 6.8: Average and Variance for the raw workload index of each control device, obtained as part of the single-factor ANOVA analysis.

The motion controllers also yield the smallest variance of 271.111. RVIZ has a variance of 271.111 in comparison, while the hands and the pen have very high variances of 706.944 and 721.181. Consequently, the pen obtains the lowest average SUS Score of 56.25, despite obtaining good results from several participants.

In Figure 6.3, the average SUS Scores for each device are summarised in a histogram plot.

Following the SUS scale shown in Figure 6.2, the control interface based on the motion controllers is in the acceptable range, rated as a system of good usability, of grade C. RVIZ is in the grade D, in the high marginal acceptability range. Both the hands and the pen are rated with the F grade, but still in the low marginal acceptability range, which is still categorised as an “OK usability”.

6.2.3 Global Questions

At the end of the whole experiment, the participants were given 4 global questions centred on their interaction with the robot. The results are presented in Table 6.10. The participants first commonly agreed that their interaction with the robot was fluent, with an average grade of 4.1/5 and a standard deviation of 0.8756. Their answer concerning their trust in the robot’s action reliability was rated neutral, and the standard deviation was high (1.333). The user’s ability to predict the robot’s intentions yielded a mixed average result of 3.9. It is rather positive, however, the standard deviation is high as well (1.1972).

The 10 participants finally all agreed that they felt safe during the experiment, despite a human-sized robot moving around them.

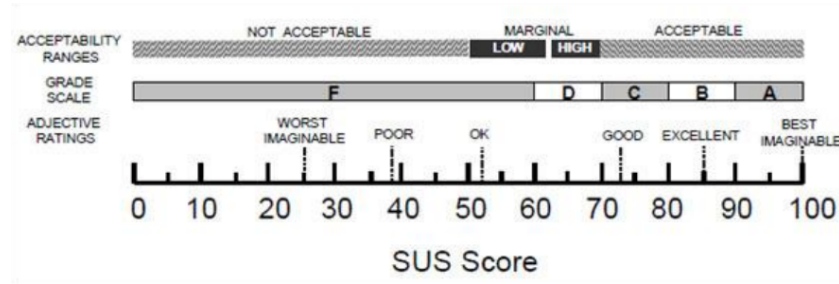


Figure 6.2: Scale used to evaluate the obtained usability score for the SUS questionnaire, from [18].

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	68.5	77	57.5	56.25
Variance	271.111	152.5	706.944	721.181

Table 6.9: Average and Variance of the participants' SUS Scores, for each control device, obtained as part of the single-factor ANOVA analysis.

6.3 Quantitative Results

In this section, the measures of objective task efficiency are presented. As detailed earlier in the manuscript, 3 time durations were measured: the task completion time, the task interaction time, and the task waiting time. Aside from these metrics, the number of navigation requests sent by the participant was recorded, as well as the elapsed time between consecutive requests, presented as an average duration for each trial.

6.3.1 Task Completion Time

Figure 6.4 displays the task completion time of each control device and for each object location. A difference between the difference control devices is rather difficult to observe, however, we can notice a clear vertical shift in the values for the kitchen table compared to the ones associated with the 2 other object locations.

Fixed Variable: Type of Control Device

A first single-factor ANOVA analysis confirms that there is no statistical difference in task completion time between the control devices, as the p-value has a value of 0.73. This is also confirmed in Table 6.11, as the different averages in completion

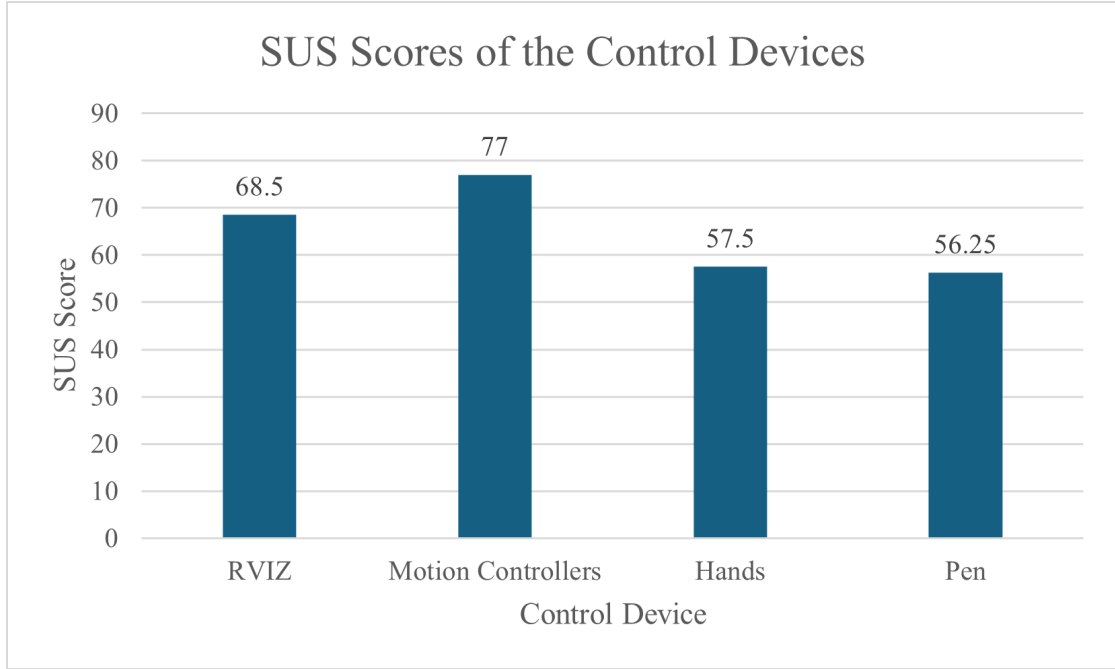


Figure 6.3: Histogram of the average SUS score for each control device.

time are really close to one another (there is only a standard deviation of 3.519 seconds between the four averages). The variances are also high, with the lowest variance value of 1284.177 obtained for RVIZ.

Fixed Variable: Object Location

In reaction to the high p-value of the first ANOVA analysis and after observing a noticeable shift in the data repartition of Figure 6.4 between the three object locations, another ANOVA test was taken on the same dataset, but now using the object location of each trial as the fixed variable.

Compared to the average values of the task completion times computed for each control device, we can observe a distinct difference between the average completion time for an object placed on the kitchen table with the values for the 2 other object locations.

The kitchen table, considered as a difficult location by the experiment conductors, yielded an average duration of 73.949 seconds as shown in Table 6.12 (however, with a high variance of 1589.28, corresponding to a standard deviation of 39.866 seconds).

This shift between average values is confirmed with the p-value obtained with the analysis, resulting in a value of 0.0233. This result states that the object

Variable	“My interaction with the robot felt fluent.”	“I did not trust the robot to do the right thing at the right time.”	The robot’s intentions were predictable.”	“I did not feel safe.”
Average	4.1	3	3.9	1.5
Std Dev.	0.8756	1.3333	1.1972	0.7071

Table 6.10: Participants’ answers to the global questions concerning the whole experiment.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	59.635	62.791	63.901	56.070
Variance	1284.177	2453.886	1708.211	1368.845

Table 6.11: Average and Variance of the task completion time, for each control device, obtained as part of the single-factor ANOVA analysis.

location has a significant effect on the task completion time.

The living room shelf appeared to be the most consistent location between participants, as it has the lowest variance value (however, still high since it corresponds to a standard deviation of 28.385 seconds), with the lowest average time of 44.176 seconds.

6.3.2 Task Selection Time

For this metric, isolating the selection time of other perturbations not related to the developed system, Figure 6.5 seems to strongly highlight a longer selection time for the hands control interface.

This is confirmed with an ANOVA test, resulting in a p-value of 0.0075, indicating that the type of control device has a strong effect on the task selection time. This information is illustrated in Table 6.13 as well, with the average and variance values of the selection time of each device.

As we can clearly notice in Figure 6.5 with the boxes being higher and elongated than the ones of the other devices, the hands as a control device yield the highest selection time of 4.670 seconds. However, RVIZ has the highest variance with a value of 29.987 (corresponding to a standard deviation of 5.476 seconds).

The pen is the fastest selection device among all four, with an average selection

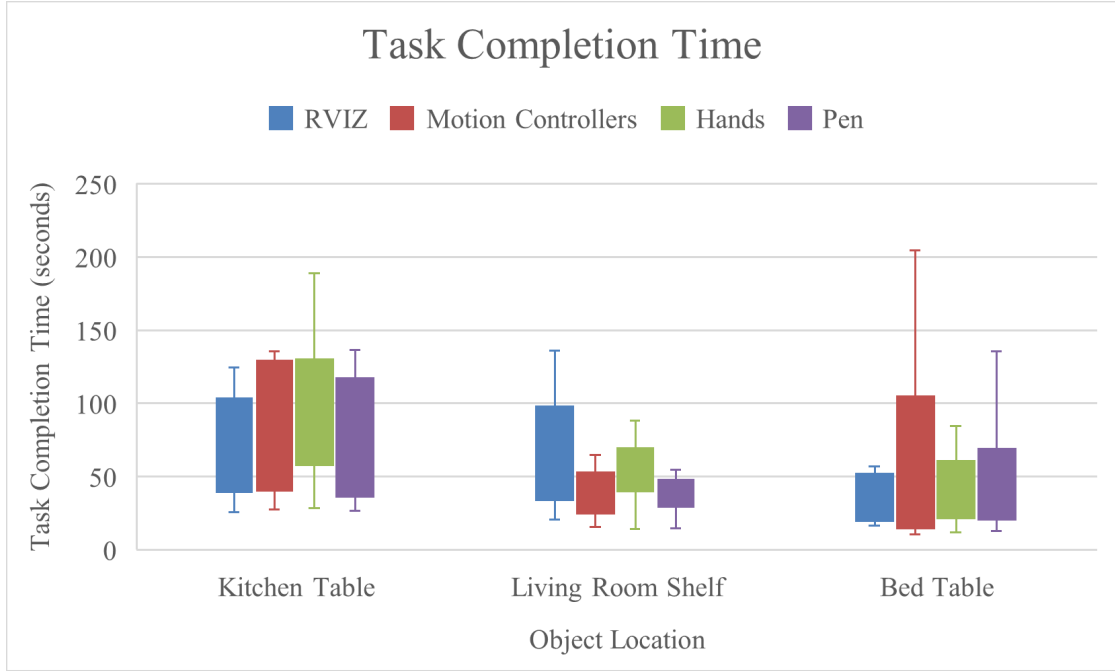


Figure 6.4: Box plots of the task completion times of each control device, for each object location.

Variable	Kitchen table	Living room shelf	Bed table
Average	73.949	44.176	49.770
Variance	1589.28	805.692	1328.458

Table 6.12: Average and Variance of the task completion time, for each object location, obtained as part of the single-factor ANOVA analysis.

time of 2.438 seconds and the lowest variance, corresponding to a standard deviation of 2.558 seconds.

Another ANOVA test was conducted to check if the object location had any effect on the selection time, and this test resulted in a p-value of 0.2013, indicating a low effect.

6.3.3 Task Waiting Time

In Figure 6.6, RVIZ distinctly produces the highest task waiting time. This is confirmed in Table 6.14, summarising the average and variance values of the task waiting time for the different control devices. The use of RVIZ during the experiment resulted in an average waiting time of 16.231 seconds with a variance

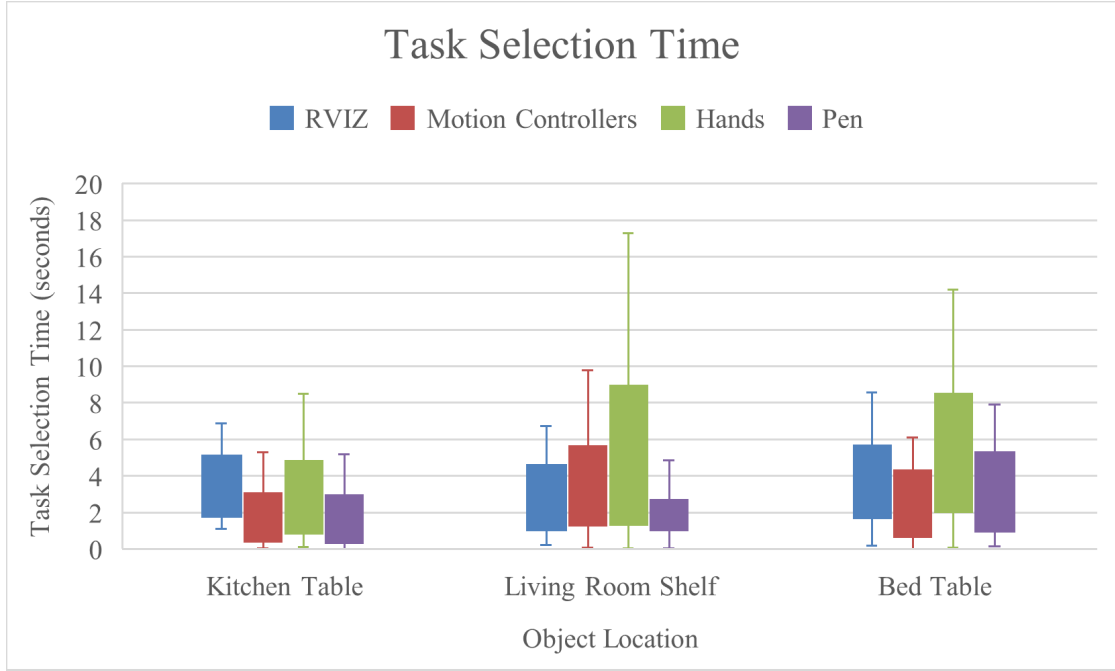


Figure 6.5: Box plots of the task selection times of each control device, for each object location.

of 188.673 (or a standard deviation of 13.736 seconds). This is, however, not the highest variance, as the motion controllers led to a value of 328.868, or a standard deviation of 18.135 seconds.

However, despite this rather pronounced observation, the p-value generated by the ANOVA test had a value of 0.446, which indicates no significant difference between devices.

6.3.4 Number of Navigation Requests

From Figure 6.7 we can easily deduce that RVIZ will yield the smallest number of navigation requests: its boxplots and whiskers are short and closer to 0 than the boxplots of the other devices. This observation is confirmed numerically with Table 6.15, in which RVIZ has the smallest average and variance values (4.6101 and 16.897, respectively).

The ANOVA test confirms that there is a significant difference in the number of navigation requests between devices, as the p-value is 0.0094.

However, the device that is associated with the highest number of navigation requests is not really distinctive between the hands and the pen.

This hesitation is explained in Table 6.15, as the hands are associated with

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	4.127	3.058	4.970	2.438
Variance	29.988	10.619	26.589	6.542

Table 6.13: Average and Variance of the task selection time, for each control device, obtained as part of the single-factor ANOVA analysis.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	16.231	15.645	12.564	12.904
Variance	188.673	328.868	263.891	148.596

Table 6.14: Average and variance of the task waiting time, for each control device, obtained as part of the single-factor ANOVA analysis.

the highest average number of requests with a value of 11.016, but have a high variance (287.305), while the pen has the second biggest average number of requests (average: 9.542), but also a rather small variance in comparison to the hands (variance: 71.632).

From the same graph, we can also expect that the number of requests sent has a dependency to the object location, as the data distribution for the Kitchen table seems to be vertically shifted, compared to the two other object locations.

Another ANOVA test was run, using the object location as a fixed variable, and it obtained a p-value of 0.008, which indicates that the number of target requests depends on the object location as well.

Table 6.16 indeed highlights that the kitchen table location was associated with a higher number of sent requests, with an average of 11.413 requests sent (variance: 120.600). The trials conducted with the living room shelf produced the lowest average number of navigation requests, with a mean of 6.575 sent requests, with the lowest variance among the three object locations (variance: 46.602).

6.3.5 Mean of the Time Differences Between Navigation Requests

The mean of time differences, or the average time the participants spent waiting before re-selecting another navigation target, was measured as well. The following graph 6.8 was obtained, on which we can notice a tendency of RVIZ to produce higher values, as the corresponding boxes are shifted compared to the ones of the

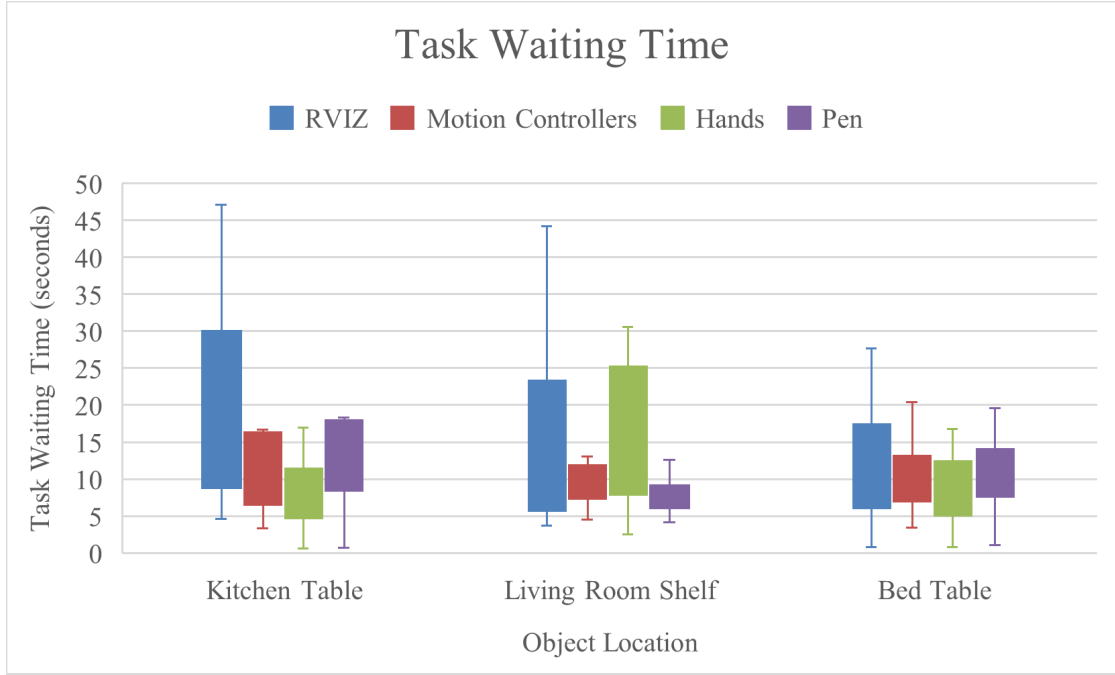


Figure 6.6: Box plots of the task waiting times of each control device, for each object location.

other devices.

The ANOVA test revealed a p-value of 0.0045, highlighting a significant difference in this metric between control devices.

Table 6.17 indeed shows that RVIZ led to the longest time differences between requests, with an average of 14.233 seconds. However, it had the highest variance among all four devices (value: 129.965). The pen achieved the smallest average of mean time differences of 7.911 seconds, with the smallest variance of 57.490.

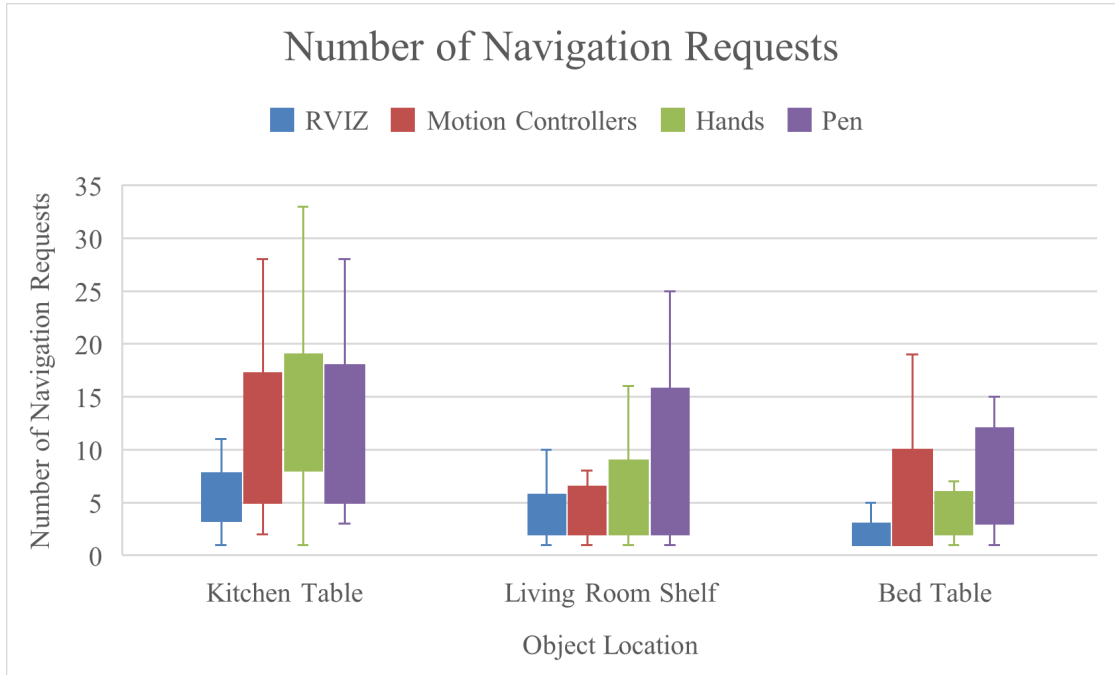


Figure 6.7: Box plots of the number of navigation requests sent for each control device, for each object location.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	4.6101	8.383	11.016	9.542
Variance	16.897	76.512	287.305	71.632

Table 6.15: Average and variance of the number of navigation requests sent, for each control device, obtained as part of the single-factor ANOVA analysis.

Variable	Kitchen table	Living room shelf	Bed table
Average	11.413	6.575	7.179
Variance	120.600	46.602	176.850

Table 6.16: Average and variance of the number of navigation requests sent, for each object location, obtained as part of the single-factor ANOVA analysis.

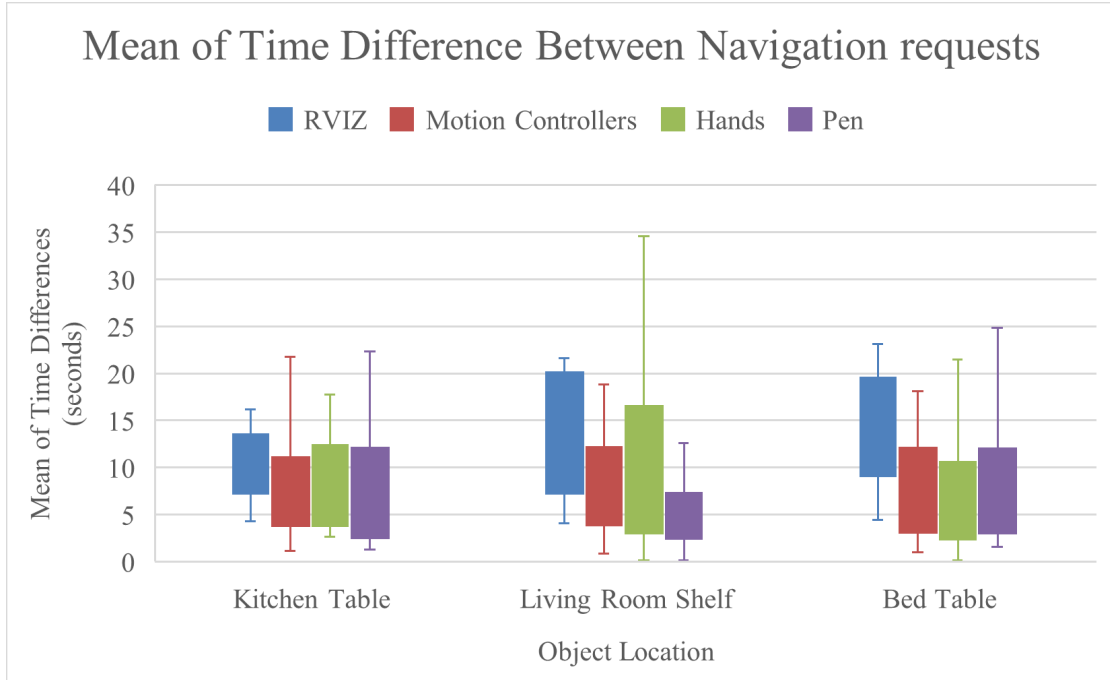


Figure 6.8: Box plots of the mean time differences between navigation requests, for each control device and each object location.

Variable	RVIZ	Motion Controllers	Hands	Pen
Average	14.233	10.084	8.554	7.911
Variance	129.965	122.8513	70.908	57.490

Table 6.17: Average and variance of the mean time difference between navigation requests, for each control device, obtained as part of the single-factor ANOVA analysis.

Chapter 7

Discussion

This chapter presents an objective discussion of the results presented in the previous chapter, while also considering the observations made during the experiments by the experiment conductors as well as the different issues encountered, which will be described in this chapter before moving on to the results discussion.

7.1 Observations during Experiments

This section presents a summary of the observations taken during the experiments. When a participant was performing a trial, the experiment conductors took notes of his overall behaviour, hints of his possible reasoning strategies as well as the decisions he made to optimise the robot's navigation.

7.1.1 Behaviour Towards the Robot

One of the first things we noticed was that no participant was visibly afraid or uncomfortable with the robot. All seemed to have understood of how much in control of the robot they were and looked like they mostly expected the robot to perform the action correctly, however with a certain delay. By looking back at the trials during which the participants were allowed to move, the experimenters often detected the participants that were clearly comfortable with the robot's presence, as they were usually walking around the scene without taking a look at the robot moving behind them, except when the robot was close to its destination.

However many would turn their back or stay behind the robot to simply check that it was performing the right actions without getting stuck around obstacles for example. This issue would however frequently happen and generate a lot of frustration to the user, especially when participants were trying to get the robot very close to the target destination that was close to different obstacles. Several

participants were so frustrated by this robot’s behaviour that they mentioned by themselves that this issue would probably affect their questionnaire answers, regardless of the control device used.

Several participants were caught talking in a friendly way to the robot, with questions such as “where are you going?”, “what are you doing”, etc. This demonstrates the clear importance of developing multi-modality control interfaces, as speaking in our most natural, intuitive language.

7.1.2 Task Efficiency

As mentioned in the experimental setup, for each object location in the room, the participants performed one trial while staying at their desk (stationary trial) and one trial during which they were allowed to move in the room freely (dynamic trial). For the dynamic trials, most participants would stand up directly to move close to the navigation destination. However, several participants did not want to move at all and stayed seated, except for some trials that seemed too difficult. Also some of them would just stand up and catch a small insight of the robot’s position and destination, instead of moving far in the room. Another interesting observation we made was that for one participant in particular, who did not stand at any time during the whole experiment, stood for the very first and only time when using the hands to make the robot reach the kitchen table. Among all participants, we could notice that the kitchen table location was more difficult than the other two, as the robot had to turn around half of a table and place itself in a relatively narrow space. However for this participant, despite having navigated the robot to that destination several times before with other devices, he only stood up while using the hands to point at that target. This indicates that the hands might not be as efficient and easy as other devices to point at far destinations. This struggle with the hand-based interface for far destinations was also observed with other participants, and is quite confirmed in Figure 6.4 shown in the previous chapter, where we can notice that the boxplot related to the hands for the kitchen table location is visibly higher than other devices.

This clearly indicates that users can simply get lazy really fast with such navigation tasks, enhancing the need to develop interfaces yielding a low user’s workload, or the technology would not be adopted or used correctly. Hopefully, it also indicates that the interfaces developed in this master’s thesis were suitable for participants unable to move, as the participants staying seated at all times were performing as well as others.

To solve the problem of the robot getting stuck around obstacles, there were two main behaviours. Some participants would just wait for the robot to find a new path without doing anything. Such participants were sometimes just not sure if they could intervene to help the robot, but as soon as they were aware that

they could, they directly selected a new target position for the robot. A few of these participants were also well aware that they could help the robot, but chose to just wait for it to move by itself. The majority of the participants would have the second behaviour of directly selecting new navigation targets. Some users did not understand why the robot was stuck and would just try to select new positions very close to the problematic one, which would not solve the problem. However others, especially participants that had a high previous experience with robots, would understand that the best way to help the robot was to select a target far from the obstacle, to make the robot move away from it, and later reselect other intermediate targets to force the robot to use a simpler path.

This method was indeed clearly observed in a few participants, which yielded significantly better results than the others. Such users were using waypoints, or intermediate navigation targets, to force the robot to go down a path using mostly straight lines, avoiding close turns around obstacles that were the problematic regions of the map for the robot's autonomous path planning, and thus the places in the room where it would often get stuck. We could observe that for each object location, these participants used almost the exact same path with all four devices, proving that they had a clear navigation strategy in their head.

As the participants were also well aware of the time limit they had to reach the destination during the trials, for one specific object location that was on the other side of the room (the kitchen table), in dynamic trials, two participants had the habit of running a little to the target destination and point at it directly, then wait for the robot to come to them. This solution was not really efficient as the robot was navigating around the whole room in full autonomous mode, which would make it get stuck in turns near obstacles. Thus, these two users had sometimes to go back near the robot to select closer targets for it to be reach more easily.

7.1.3 The Control Devices

Several observations recorded during the experiments were very specific to the type of control device.

During the Practice

After they were shown a scheme of how to activate the target selection with RVIZ, as in Figure 4.2, the participants didn't express any concern while learning how to use RVIZ to control the robot during the practical training. Some of them just mistakenly pressed the button that was right next to the correct one at the very first try during the practice. Most of the participants were very quick to catch and remember the correct button associated with the target selection for the motion controllers. For the hands as well, the participants learned easily the activation

gesture needed to select a target. However, many of them struggled to remember the selection button of the pen, despite it having only 4 possible buttons on one device, compared to the motion controller having 5 buttons and 1 joystick on each handheld device.

The experimenters first believed that this learnability difference was possibly due to a difference of clarity between the two control schemes presented in the tutorial video, shown in the right picture of 4.3 for the VR motion controllers, and in the right picture of 4.5 for the XR pen. Even if these pictures were accompanied by zoomed-in videos of the control device being correctly used, for the XR pen it seems to be still difficult to isolate the selection button from the two other buttons next to it, as they are pretty close to each other. The experimenters also thought that this memory difference might be due to the tags put on the buttons of the VR motion controllers: each button on the front side of the device has its letter, which is not the case for the pen.

During the Trials

When using RVIZ, as the scene was in their back, most of the participants frequently checked the scene behind them to confirm that the robot was correctly following the requested actions and to assess its position in real time in the room, which clearly demonstrates how AR-based interfaces are helpful to allow the operator to focus on only one field of view for the whole task, reducing his mental load.

It was already predictable that the motion controllers would be preferred by the participants in usability and workload surveys. Indeed, most participants looked very confident and comfortable with the motion controllers in their hands, compared to the use of the pen or the hands with which several participants were frequently shaking their arms to rest them. While pointing at far distances with one arm, using only their hands or the pen, several participants rested their inactive arm on the desk, seemingly trying to stay stable while pointing. This observation was not seen when the participants were using the motion controllers.

The participants were already annoyed by the robot being stuck near obstacles as explained in a previous discussion of this section, however, they looked even more frustrated while pointing targets with their hands. The hands interface was subject to a lot of false positives, as the participants' hands were often wrongly detected as performing the select gesture while they were just at rest. This issue generated a lot of frustration among the users, as they frequently had to retake navigation requests for that reason. This is a topic that will be more detailed in the next session.

The XR pen was also expected to yield a bad user experience, as some difficult issues were encountered during the experiments, which led to the pen being disconnected for a certain time, and sometimes the app crashing. However, when

the trials were not affected by these bugs, we could clearly notice that the selection with the XR pen was particularly efficient compared to other devices, and the participants seemed to point more easily at distant targets compared to the VR motion controllers and especially the hands.

7.1.4 Advice From the Participants

Many participants, especially several experts who mentioned using the RVIZ software frequently, exclaimed their appreciation of the AR-based control interfaces, saying comments such as “This is easy”, “Very practical”, or also “Very cool”, especially when using the VR motion controllers. This clearly highlights the potential of using AR-based interfaces in service environments but as well in research as a daily tool. Similar use-cases of using AR for robotic development in research do exist, such as in [25], where an MR-based interface is used to allow an operator to dynamically assist a robot’s data collection process by correcting classification errors.

After watching the tutorial video, when the participants tried selecting a navigation target for the first time, many of them quickly asked the experiment conductors if there was a way to cancel the selected target. As there is no “cancel function” implemented in RVIZ for 2D Navigation goal requests, it was not implemented either for the AR-based control interfaces to maintain consistency. This advice was mentioned by many participants, and the experiment conductors imagine that if this function was implemented, it would have very likely reduced users’ perceived frustration and fatigue. This way, the participants could have stayed even more in control of the navigation, and avoid wasting time as in situations where the robot had already started moving to the wrong direction due to a false positive hand gesture detection for example.

7.2 Issues Encountered During the Experiments

During the experiments, several technical issues occurred, possibly influencing the different results.

7.2.1 Hand Gesture: False Positive

The first problem was already mentioned in the previous section and concerned the hand-based control interface. Indeed, the hand gesture used to select a target, a “pinch” gesture, was often falsely recognised when the user’s hand was at rest. This issue resulted in higher frustration ratings for the hands as a control device, while having an impact on task completion times and the number of targets sent.

What was happening was that when the user was resting his hands between the trials, the “resting gesture” of the hand was too close from the “pinch” gesture, as the latter is defined almost only by a contact between the index and the thumb. Thus, when the participant was waiting for another trial to start, the hands were recognised and still tracked as he was wearing the headset, and the resting gesture of his hands were recognised as an activation gesture. Since our hands naturally point towards the ground when they are at rest, the rays were activated, and in that condition, the activation gesture would select the position pointed by the hand-rays, which was often near the feet of the participant. The robot would change its navigation goal and try to reach that position, but failing inevitably as it was placed in the middle of detect obstacles. When that situation happened between the trials, the experiment conductors would simply reset the robot’s position to its origin. However, when it happened during a trial, the participant himself would re-select a target to overwrite this false navigation request.

Also, the experimenters observed that the participants themselves noticed this problem and forced themselves to extend all the fingers of their inactive hands to avoid them being recognised as a selection gesture, which worked well.

7.2.2 XR Pen: Disconnection

The second problem was related to the XR pen control device. After several participants had already taken part in the experiments, we encountered a strange bug while a participant was using the pen. What happened was: the participant was waiting for the start of the trial, sitting on his chair, the pen in his hand. As soon as he started, he pointed at a position in the room and tried to select it with the pen: the target visual and the arrow appeared, but froze instantaneously. After further investigations, we discovered that while the participant was waiting at his seat with the pen in his hand, the pen got disconnected as it was detected as inactive. The AR application directly recognised the hands motion instead and was tracking them. When the participant turned to the scene and tried pointing with the pen, the hands were still detected, so the rays coming from the hands were interacting with the ground. When the participant pressed the pen’s button to select the target, as he didn’t notice that the pen got disconnected because it happened very fast, the “pinch gesture” was somehow detected through the motion made to press the pen’s button. However, since the button had been pressed, the pen reactivated itself. The issue is that if the button is not released, the visual target and arrow will not disappear. As the hands got disconnected when the pen’s button was pressed, the AR application never saw the “pinch gesture” from the hands being released, and the appeared target and arrow visuals could only disappear with that gesture being released, not any other button.

The experiment conductors made further research to find what was causing

the pen to get disconnected so easily and inspected the related scripts to find a possible issue. Even after revising the related scripts, the issue was still present. Thus, the experimenters looked for a parameter that could control the activation time of the XR pen. In the headset settings, they found a parameter controlling the threshold of activation force necessary to keep the pen active (this activation force was the force with which a finger was pressing on a certain button of the pen). However, even setting this parameter to the lowest value possible, it didn't solve the problem. After further investigations, we discovered that other users of the XR pen from Logitech with the Meta Quest 3 headset had a similar issue. Some of them contacted the technical support of Meta, who confirmed that this problem was a compatibility issue between the Logitech MX pen and the Meta Quest 3, and was still under investigation. The experiment conductors thus categorised this issue as a technical limitation of the XR pen.

The experimenters noticed that gently shaking the pen while not using it would keep the inertial sensors active and keep the pen connected. Thus, in order to continue the experiments and keep recording data, the researchers asked the participants to shake the pen while they were not using it. Thankfully, this helped the pen keep being connected and we were able to record data with this control device. The same issue appeared from time to time when a participant forgot to keep the pen active, and if the pen got disconnected, we would first ask the participant to reactivate it by pushing on its buttons. If it was not working or if the target and arrow visuals had appeared and frozen, we would first ask the participant to put the pen back in its support, to force its deactivation, while asking him to connect his hands or the motion controllers to make the frozen visuals disappear before reconnecting the pen. If the issue was still appearing, we would take the headset and restart the application, which always solved the issue.

Due to this technical limitation, we could easily expect that the usability and users' workload questionnaires would be greatly influenced, which was the case. However, the numerical data recordings proved to us that, when the pen did work without bugs, it yielded very good task efficiency results compared to the other devices. But this will be discussed in the following section.

7.2.3 Robot's Path Planning Parameters

As mentioned in the previous section concerning the observations made during the experiments, the robot's autonomous path planning was not the most optimal and impacted the user's frustration during the task. Even if the room floor plan had been enlarged to help the robot navigate smoothly between obstacles and furniture, its path planning parameters had to be fine-tuned. The parameters that could be modified were the robot radius and inflation radius for the local and global cost maps. Despite taking the default values recommended by the robot's

manufacturer, when the robot was working in autonomous navigation to reach the position requested, the robot would take very tight turns around obstacles, especially near the corner of a bed or a table, and would stop moving. When it happened, the robot turned on itself to update its local cost map by scanning its surroundings, until it found a new path to reach its destination and start moving again. For novice users, this behaviour was frustrating as they could not really understand why it was happening and did not know what to do at first. However, as explained earlier, they could deduce that the robot was too close to an object, and after some time, they usually selected another navigation target to help the robot find another path. For users with extensive previous experience with robots, this behaviour was still frustrating for them but they had a better idea of how to help the robot navigate efficiently.

Even though this technical limitation clearly influenced the participants' frustration ratings, it was an interesting and realistic challenge for a navigation task in service robotics, as this situation, even with better path planning parameters, could still happen in a home-like environment. This is already a use case of small autonomous vacuum cleaners deployed in many households nowadays. If the autonomous vacuum cleaner gets stuck somewhere in the living space, the robot notifies the user via an application, for example, that it got stuck in some place and requires assistance.

7.3 Discussion of the Results

Based on the observations and issues encountered during the experiments, the results presented in the previous section are discussed.

7.3.1 The Hypotheses

This is a summary of the hypotheses yielded during the development of the proposed approach.

These are the two hypotheses for the user experience:

- “The AR-based control interfaces improve the user experience and decrease the user’s workload, compared to the computer-based control interface.”
- “The control interface based on the XR pen device enhances the user experience and decreases the user’s workload, compared to the other AR-based control devices.”

These are the two hypotheses for the navigation task efficiency:

- “The AR-based control interfaces enhance the navigation task efficiency compared to the computer-based control interface, by reducing the task selection times, but not for difficult and occluded target locations.”
- “The control interface based on the XR pen device improves the navigation task efficiency by yielding shorter task selection times, compared to the other AR-based control interfaces.”

7.3.2 User Experience

The user experience of each control device has been evaluated through two questionnaires that resulted in two ratings: the user’s task load index from the NASA-TLX questionnaire and the user’s perceived usability of the system.

In the previous chapter, results showed that the VR motion controllers obtained the lowest average raw task load index of 25.542/100, while the hands yielded the highest average of 40.25/100. The XR pen had the second-worst TLX rating of 39.583.

The SUS scale presented the motion controllers as the control interface with the highest user’s perceived usability score among all four devices with a rating of 77, while the XR pen had the worst rating of 56.25, and the hands had the second worst rating of 57.5.

Both these indexes confirms that the computer-based control interface is outscored by an AR-based control interface, which confirms partially our first hypothesis, since both the hands and the XR pen have worst ratings than the computer-based interface.

Our second hypothesis is not confirmed, following the XR pen bad ratings of system usability and task load index. However this thesis states that the user experience of the pen interface has been strongly influenced by technical limitations of the hardware and integration conflicts. Following the observations made during the experiment, this thesis suggests that the pen would have yielded better results if the disconnection issue mentioned in the previous section had been discovered prior to the start of the experiments. Interestingly, the experimenters have observed that the disconnection problem never occurred during the practical training. It would have been interesting to have asked the participants to ask a first simplified version of the SUS and TLX questionnaires just after the practical training, to evaluate their very first impression of the control devices.

Concerning the hands, the bad rating for the user experience was predicted, following this research’s results [12]. However, this thesis hypothesises from observations made during the practical training and during the experiment that the use of another selection gesture, one that would be especially less close to the

natural resting pose of the hand, would increase the user’s perceived usability of this hand-based interface.

7.3.3 Objective Task Efficiency

Our first hypothesis is partially confirmed, as both the VR motion controllers and the XR pen have produced shorter task selection times than the computer-based control interface RVIZ. However, the hand-based interface has resulted in a higher task selection time compared to RVIZ. Our second hypothesis is however directly validated, as the pen produced the smallest average task selection time of 2.438 seconds and the lowest variance.

Concerning the other metrics, our hypothesis concerning the number of navigation requests is confirmed, as all three AR-based interfaces have produced significantly higher average numbers of requests compared to the computer-based interface.

We can also observe that the hand-based interface had the highest number of requests sent (with, however, a very high variance), which numerically confirms the hypothesis that some participants have clicked on the selection button multiple times out of frustration. A corresponding observation has been made by the experiment conductors during the experiments, as explained in a previous section.

The mean time difference is however, not completely inversely proportional to the number of navigation requests, as the pen here yields the shortest mean time difference (instead of the hands as it was expected). This confirms again that the XR pen had the fastest selection time out of all four devices.

Chapter 8

Conclusion

This thesis presented three control interfaces in XR, based on navigation target pointing, for HRC. Three XR-based control methods were developed for three control devices: one using the XR motion controllers, another using the operator's hands being tracked a XR controllers, and one using an XR pen control device.

In the context of my master's thesis, I developed the following interfaces during a three-month internship in Japan:

- A Topic Publisher script: permits publishing messages on the ROS network, from the Unity application to the robot.
- A Unity Logs Publisher script: permits publishing the debug logs of the Unity application to the robot.
- A Navigation Goal Control script: permits the robot to navigate to a position in the room by pointing at this position with 3 XR controllers: the XR motion controllers, the operator's hands, and the XR pen controller. This script also permits controlling the orientation of the robot by moving an arrow with the XR-based controllers.
- A Device Handler Wrapper script: Based on the device handler script for the Logitech package for Unity, permits the integration of the XR pen controller to an existing project with other XR-based controllers already included, without generating any input conflict.
- A Data Processing script Python script to process the data collected during the experiments.
- A Select Gesture script: permits to select a position with the operator's hands by making a "V" gesture with the hand. (not kept in the application).

- A Base Control script: permits to control the robot’s motion and orientation with the joysticks and the buttons of the XR motion controllers. (Not used in the proposed approach)
- A Gripper Control script: permits to control the gripper of the robot by pressing on a button of the XR motion controllers. (Not used in the proposed approach)
- A Head Control script: permits to control the robot’s head motion (tilt and span), with the joysticks and the buttons of the XR motion controllers. (Not used in the proposed approach).

These programs were systematically reviewed by the M2 student Carl Törnberg before being integrated into the Unity XR project that he developed for his own master’s thesis.

The purpose of my thesis was to evaluate the XR pen-based control interface in terms of objective task efficiency and user experience. In that context, this thesis developed the following hypotheses:

Concerning user experience:

- **H1:** “The XR-based control interfaces improve the user experience and decrease the user’s workload, compared to the computer-based control interface.”
- **H2:** “The control interface based on the XR pen device enhances the user experience and decreases the user’s workload, compared to the other XR-based control devices.”

Concerning objective task efficiency:

- **H3:** “The XR-based control interfaces enhance the navigation task efficiency compared to the computer-based control interface, by reducing the task selection times, but not for difficult and occluded target locations.”
- **H4:** “The control interface based on the XR pen device improves the navigation task efficiency by yielding shorter task selection times, compared to the other XR-based control interfaces.”

To validate these hypotheses, a comparative study between these three XR-based interfaces and a computer-based control interface serving as a baseline was performed on a set of 10 participants who were asked to perform a simple navigation task to control a semi-autonomous mobile robot. The experiments were conducted by Carl Törnberg and myself, during which he gathered preliminary results for his own thesis at the same time. In this context, the script to collect the data

generated during the participants' experiments was written by him. This script also permitted to reset the robot's position in the room and set the trial parameters (controllers, object location, etc).

To evaluate user experience, participants were asked to fill out the NASA-TLX and SUS questionnaires for each device tested, to measure their task load index, and their perceived usability of the system. During the experiments, the task completion times and isolated selection times and waiting times were measured, alongside all the navigation requests sent.

The hypotheses H1 and H3 have been partially confirmed as the computer-based interface RVIZ was always outscored by at least one XR-based control interface, both in user experience and task efficiency metrics, however, the hand-based interface had both worse user ratings and efficiency results than RVIZ.

The hypothesis H2 has been rejected, as the XR pen had the worst system usability score (SUS score = 56.25) among control devices and obtained the second-worst TLX rating after the hand-based interface. However, a technical limitation of the device greatly influenced the user experience, as several technical issues requiring experimenters' intervention occurred during the experiments with the XR pen.

However, the hypothesis H4 is validated as the XR pen raised the lowest task selection time among all devices, with an average of 2.438 seconds, which was significantly different among devices (p -value = 0.0075). It also yielded the smallest variance, showing that it was the most consistent control device among participants.

The type of control device had a significant effect on the mean time difference between navigation requests (p -value=0.0045). The XR pen also produced the smallest mean time difference between navigation requests, with an average of 7.911 seconds and the smallest variance among all control devices (variance = 57.490).

In the end, the preferred device in terms of user's workload and system's usability was the XR motion controllers, with scores of 25.542 (TLX score) and 77 (SUS score) respectively. The hands were the worst device in terms of task load index as they yielded the worst frustration ratings, which were moderately different among devices (frustration average rating = 10.6, variance = 44.933), with a raw TLX score of 40.25. These results are explained by a technical limitation of the hand gesture used to trigger the selection of a position, as it was impacted by false positives during the experiments. Both the hands and the XR pen had the highest variance among the control devices concerning the TLX and SUS results, which indicates that the obtained scores were not consistent among participants. This is confirmed as these scores were significantly better for the participants who did not encounter any bugs during their trials.

More in-depth studies will have to be conducted to evaluate the user's perceived usability and workload of the XR pen control device, after resolving several technical

limitations concerning the XR pen frequent disconnections and its compatibility in applications that integrate other XR controllers. This thesis confirms as well the need for further investigations concerning hand-gestures used to trigger inputs, as mentioned by [6]. The hand-free characteristic of such a control interface is motivated by various use cases, both in industrial and service environments. Based on observations made during the experiments for this thesis, extending pointing-based control interfaces with a method to cancel a target selected by mistake, such as how [12] implemented a confirmation widget appearing at target selection, is expected to enhance users' frustration levels. Multimodal interaction methods, including the detection of natural languages, could enhance the fluidity of the HRC.

Current trends in the field of Extended Reality for Human-Robot Collaboration include the development of control techniques to enhance task efficiency with Multi-Robot System (MRS) [5], [26] and the use of hand-free control inputs that yield an even lower perceived mental load for the user, using EEG signals [20] and eye-tracking [12] to trigger inputs.

Appendix A

Results

A.1 Questionnaires

A.1.1 NASA-TLX

Participant	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
1	1	1	1	4	4	6
2	10	2	4	10	10	3
3	11	12	10	19	7	7
4	6	1	2	2	2	5
5	0.5	0.5	1	0.5	1.5	1
6	5	1	1	15	5	1
7	1	1	1	1	1	1
8	3	1	12	11	3	11
9	3	3	7	3	5	1
10	16	4	10	10	10	15

Table A.1: Participants' answers to the NASA-TLX questionnaire for the RVIZ software.

Participant	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
1	1	4	1	7	6	7
2	10	2	7	13	12	2
3	4	4	6	16	7	12
4	2	2	2	2	2	2
5	1.5	1.5	4.5	5.5	2	3.5
6	1	1	1	19	1	1
7	1	1	1	1	1	1
8	5	2	6	3	15	16
9	3	3	6	8	5	8
10	5	5	10	10	5	10

Table A.2: Participants' answers to the NASA-TLX questionnaire for the XR motion controllers.

Participant	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
1	5	3	3	7	7	10
2	7	3	10	9	10	2
3	15	9	16	16	14	13
4	2	1	2	4	1	4
5	3.5	5.5	6	5.5	3.5	5
6	16	1	1	16	16	16
7	4	6	3	3	3	3
8	17	4	12	6	14	18
9	3	3	8	15	15	20
10	4	4	10	15	10	15

Table A.3: Participants' answers to the NASA-TLX questionnaire for the hands.

Participant	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
1	10	10	5	14	14	15
2	10	1	10	11	10	4
3	12	12	12	19	8	8
4	4	2	2	4	2	5
5	2	7	2	10	6	7
6	5	1	1	15	1	16
7	1	1	1	1	1	1
8	18	16	13	15	18	18
9	3	3	7	10	11	10
10	5	5	10	5	10	15

Table A.4: Participants' answers to the NASA-TLX questionnaire for the XR pen.

A.1.2 SUS

Participant	Use Frequency	Unnecessarily complex	System easy to use	Need of support	Functions well integrated
1	4	4	5	3	4
2	4	2	3	2	4
3	4	2	4	2	4
4	2	3	2	4	4
5	5	1	5	4	5
6	5	2	4	5	3
7	5	1	5	1	5
8	4	1	5	4	5
9	2	3	2	1	3
10	3	3	4	4	4

Table A.5: SUS questionnaire for RVIZ: first part.

Participant	Too much inconsis- tency	Most would learn very quickly	Very cum- bersome	Operator's confi- dence	Need to learn a lot before
1	3	4	2	5	1
2	3	4	2	4	3
3	3	4	3	4	3
4	1	2	3	3	3
5	1	1	2	5	1
6	3	4	3	4	2
7	1	5	1	5	1
8	3	4	1	5	1
9	2	4	3	4	2
10	3	2	4	2	3

Table A.6: SUS questionnaire for RVIZ: second part.

Participant	Use Frequency	Unneces- sarily complex	System easy to use	Need of support	Functions well integrated
1	4	2	4	3	4
2	3	2	3	3	3
3	4	2	4	2	4
4	5	1	5	2	4
5	3	1	5	2	2
6	5	1	4	2	4
7	5	1	5	1	5
8	3	4	4	2	3
9	4	3	4	2	4
10	5	5	5	4	3

Table A.7: SUS questionnaire for XR Motion controllers: first part.

Participant	Too much inconsis- tency	Most would learn very quickly	Very cum- bersome	Operator's confi- dence	Need to learn a lot before
1	2	5	3	4	3
2	3	3	2	3	2
3	2	5	2	4	2
4	2	4	1	5	2
5	1	5	1	4	3
6	2	5	2	5	1
7	1	5	1	5	1
8	4	5	2	4	1
9	3	4	3	4	1
10	1	5	1	5	1

Table A.8: SUS questionnaire for XR Motion controllers: second part.

Participant	Use Frequency	Unneces- sarily complex	System easy to use	Need of support	Functions well integrated
1	3	4	4	2	4
2	4	2	4	2	4
3	3	4	5	4	4
4	3	1	4	1	4
5	3	1	4	2	4
6	1	5	2	5	2
7	5	1	5	1	5
8	1	4	1	2	1
9	1	3	2	2	1
10	2	3	5	2	5

Table A.9: SUS questionnaire for Hands: first part.

Participant	Too much inconsis- tency	Most would learn very quickly	Very cum- bersome	Operator's confi- dence	Need to learn a lot before
1	3	4	3	4	2
2	3	3	2	3	3
3	3	2	4	2	4
4	3	5	2	3	1
5	1	5	1	4	3
6	5	1	5	3	3
7	1	5	1	5	1
8	5	4	5	2	1
9	4	2	4	2	2
10	1	5	4	2	1

Table A.10: SUS questionnaire for Hands: second part.

Participant	Use Frequency	Unneces- sarily complex	System easy to use	Need of support	Functions well integrated
1	1	4	2	3	2
2	3	2	4	2	4
3	3	3	4	2	4
4	3	1	4	2	4
5	3	1	3	4	3
6	2	3	4	4	3
7	5	1	5	1	5
8	1	5	1	5	1
9	3	2	4	3	3
10	3	5	5	3	4

Table A.11: SUS questionnaire for Pen: first part.

Participant	Too much inconsistency	Most would learn very quickly	Very cumbersome	Operator's confidence	Need to learn a lot before
1	4	3	3	2	4
2	3	3	2	4	2
3	3	4	3	4	3
4	3	4	2	3	1
5	1	4	3	3	4
6	5	2	4	2	4
7	1	5	1	5	1
8	5	3	5	1	5
9	2	4	2	3	2
10	1	5	2	4	2

Table A.12: SUS questionnaire for Pen: second part.

A.1.3 Global Questionnaire

Participant	Fluent interaction with robot	Did not Trust robot	Predictable robot's intentions	Did not feel safe
1	5	5	5	2
2	4	3	4	1
3	4	3	4	3
4	4	2	4	1
5	4	2	5	1
6	4	2	4	1
7	5	1	5	1
8	2	5	1	2
9	4	3	4	1
10	5	4	3	2

Table A.13: Global questionnaire about the whole experiment.

A.2 Raw Quantitative Data

A.2.1 Time Duration

VR

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:37:52.686	11:38:35.137	42.451614
2	17:06:23.000	17:08:32.443	129.442964
3	20:22:41.216	20:24:51.082	129.866202
4	14:28:36.197	14:29:19.269	43.072004
5	19:28:43.643	19:30:52.716	129.072781
6	13:05:12.064	13:07:23.718	131.653737
7	18:38:10.293	18:38:42.377	32.084137
8	17:08:52.750	17:10:04.809	72.058769
9	20:27:39.650	20:28:07.283	27.633231
10	11:15:02.119	11:17:17.704	135.585415

Table A.14: Participant timings for the trial type: kitchen table in stationary, using XR motion controllers.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:39:05.640	11:39:45.668	40.028364
2	17:09:21.049	17:11:36.308	135.258647
3	20:19:16.370	20:21:20.679	124.309742
4	14:30:15.450	14:31:42.474	87.023576
5	19:27:17.653	19:27:55.358	37.705163
6	13:03:31.949	13:04:25.256	53.307000
7	18:36:19.089	18:37:25.999	66.909436
8	17:06:48.291	17:08:21.635	93.343918
9	20:28:43.692	20:29:36.191	52.499518
10	11:18:12.160	11:18:50.095	37.935365

Table A.15: Participant timings for the trial type: kitchen table in dynamic, using XR motion controllers.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:28:03.581	11:30:45.032	161.450650
2	16:59:56.582	17:00:12.225	15.643142
3	20:16:40.049	20:17:03.087	23.038268
4	14:26:12.239	14:26:51.600	39.360728
5	19:25:43.569	19:26:36.946	53.377180
6	13:09:14.613	13:09:49.165	34.552605
7	18:40:39.701	18:41:10.270	30.569678
8	17:14:17.026	17:14:47.595	30.569194
9	20:30:23.395	20:30:39.026	15.631541
10	11:03:48.016	11:06:17.406	149.389683

Table A.16: Participant timings for the trial type: living room shelf in stationary, using XR motion controllers.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:19:54.294	11:20:44.949	50.654665
2	16:57:25.004	16:59:27.370	122.366014
3	20:15:36.894	20:16:07.679	30.784145
4	14:27:27.406	14:27:52.423	25.017081
5	19:24:26.841	19:25:10.910	44.069171
6	13:10:24.647	13:10:51.296	26.648867
7	18:39:37.022	18:40:01.691	24.669277
8	17:15:25.006	17:15:46.559	21.553392
9	20:31:14.916	20:32:19.720	64.803841
10	11:07:01.806	11:07:35.981	34.175144

Table A.17: Participant timings for the trial type: living room shelf in dynamic, using XR motion controllers.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:36:04.396	11:37:19.690	75.293562
2	17:00:36.382	17:03:32.027	175.645519
3	20:17:35.281	20:17:50.026	14.744950
4	14:33:32.910	14:33:55.087	22.176976
5	19:21:45.658	19:23:39.017	113.358303
6	13:12:37.116	13:12:51.248	14.132249
7	18:44:13.590	18:44:26.331	12.740757
8	17:13:20.106	17:13:38.398	18.291986
9	20:25:18.762	20:25:29.427	10.664544
10	11:08:29.613	11:10:19.565	109.952027

Table A.18: Participant timings for the trial type: bed table in stationary, using XR motion controllers.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:32:18.280	11:35:42.770	204.490024
2	17:03:52.571	17:06:01.586	129.015272
3	20:18:12.071	20:18:27.097	15.026360
4	14:32:44.004	14:33:10.653	26.649574
5	19:21:01.254	19:21:19.993	18.738752
6	13:11:55.620	13:12:08.839	13.218452
7	18:42:19.695	18:43:07.870	48.174295
8	17:11:13.041	17:12:43.251	90.209999
9	20:26:33.832	20:27:10.699	36.866118
10	11:10:39.140	11:10:57.672	18.531650

Table A.19: Participant timings for the trial type: bed table in dynamic, using xR motion controllers.

Hands

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	12:15:35.938	12:16:25.003	49.064665
2	15:47:16.194	15:50:25.157	188.963595
3	19:21:31.584	19:23:36.321	124.737147
4	14:44:05.879	14:44:34.358	28.478969
5	18:56:46.200	18:59:02.792	136.592735
6	12:52:25.020	12:54:53.110	148.089387
7	18:00:11.304	18:00:57.581	46.277169
8	17:27:28.332	17:28:27.645	59.313372
9	20:38:53.323	20:41:05.108	131.785336
10	12:02:09.465	12:03:44.921	95.455854

Table A.20: Participant timings for the trial type: kitchen table in stationary, using the hands.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	12:13:48.233	12:15:06.677	78.443429
2	15:53:25.114	15:53:59.465	34.351146
3	19:24:05.664	19:26:11.889	126.225341
4	14:45:44.079	14:46:42.306	58.227029
5	18:59:55.412	19:01:21.433	86.020751
6	12:50:18.480	12:51:24.795	66.315119
7	18:01:30.526	18:03:21.609	111.082867
8	17:24:53.857	17:26:44.109	110.252213
9	20:36:11.045	20:38:22.127	131.082577
10	11:59:29.426	12:00:27.364	57.938059

Table A.21: Participant timings for the trial type: kitchen table in dynamic, using the hands.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	12:11:53.048	12:13:05.464	72.415577
2	15:57:02.834	15:57:45.416	42.581738
3	19:14:51.805	19:15:36.105	44.299552
4	14:41:10.660	14:41:49.895	39.234715
5	19:07:08.579	19:07:58.361	49.782284
6	12:39:52.313	12:40:53.200	60.886790
7	18:07:31.581	18:08:05.338	33.757533
8	17:22:21.275	17:22:35.397	14.122085
9	20:45:17.251	20:46:26.347	69.096873
10	11:55:00.444	11:56:09.759	69.314794

Table A.22: Participant timings for the trial type: living room shelf in stationary, using the hands.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	12:10:19.577	12:11:16.501	56.924134
2	15:55:02.072	15:56:17.534	75.462566
3	19:13:08.428	19:14:17.138	68.709813
4	14:42:19.892	14:43:15.518	55.626427
5	19:06:19.639	19:06:37.718	18.078405
6	12:41:14.380	12:42:42.644	88.264158
7	18:04:27.390	18:05:30.185	62.794970
8	17:23:08.563	17:24:18.798	70.235622
9	20:44:06.978	20:44:49.572	42.594264
10	11:53:55.957	11:54:26.957	30.999955

Table A.23: Participant timings for the trial type: living room shelf in dynamic, using the hands.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	12:07:37.931	12:08:11.043	33.111882
2	15:59:15.943	16:02:25.107	189.164817
3	19:28:37.897	19:29:02.040	24.143615
4	14:39:37.297	14:39:55.933	18.635869
5	19:04:05.689	19:05:19.627	73.938182
6	12:47:57.425	12:48:18.769	21.343980
7	17:58:13.246	17:59:16.316	63.070085
8	17:30:29.191	17:31:24.389	55.198289
9	20:42:20.188	20:43:44.809	84.621112
10	11:56:43.911	11:56:59.434	15.522860

Table A.24: Participant timings for the trial type: bed table in stationary, using the hands.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	12:08:29.397	12:08:56.077	26.679794
2	15:58:23.967	15:58:55.815	31.847915
3	19:27:32.060	19:27:57.810	25.750642
4	14:38:30.836	14:39:11.238	40.401837
5	19:03:06.725	19:03:37.777	31.051908
6	12:48:39.623	12:49:41.807	62.183944
7	17:56:24.277	17:57:19.673	55.395943
8	17:29:38.039	17:29:59.705	21.665656
9	20:41:49.015	20:42:01.016	12.001436
10	11:57:21.269	11:57:35.714	14.444605

Table A.25: Participant timings for the trial type: bed table in stationary, using the hands.

RVIZ

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:11:19.695	11:13:07.811	108.11677
2	16:05:09.646	16:07:09.379	119.732558
3	20:05:49.064	20:07:53.828	124.76442
4	14:12:04.685	14:12:43.307	38.622733
5	19:42:49.935	19:44:00.064	70.129118
6	12:10:13.705	12:11:29.770	76.064514
7	18:23:11.425	18:23:40.375	28.949967
8	17:58:53.669	17:59:19.441	25.771783
9	19:49:04.823	19:49:43.197	38.374073
10	11:45:39.881	11:47:09.728	89.846876

Table A.26: Participant timings for the trial type: kitchen table in stationary, using RVIZ.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:08:19.593	11:10:18.454	118.861121
2	16:03:29.506	16:04:19.098	49.591894
3	20:02:29.825	20:04:22.408	112.582163
4	14:10:11.006	14:10:54.082	43.076408
5	19:40:46.736	19:41:57.751	71.014821
6	12:08:02.599	12:09:29.926	87.327663
7	18:21:45.261	18:22:30.033	44.772283
8	17:59:55.424	18:01:12.346	76.922024
9	19:51:47.470	19:52:59.504	72.033829
10	11:47:52.627	11:48:22.583	29.955446

Table A.27: Participant timings for the trial type: kitchen table in dynamic, using RVIZ.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:17:23.903	11:17:51.373	27.470312
2	16:14:28.899	16:16:44.428	135.529245
3	20:00:40.012	20:01:38.412	58.400607
4	14:20:55.327	14:21:34.850	39.523280
5	19:35:53.072	19:36:41.444	48.372026
6	12:16:39.699	12:17:41.990	62.291567
7	18:24:17.582	18:24:59.339	41.757127
8	18:01:46.996	18:03:45.167	118.170716
9	19:46:04.300	19:48:20.202	135.901915
10	11:41:58.183	11:43:47.847	109.663761

Table A.28: Participant timings for the trial type: living room shelf in stationary, using RVIZ.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:14:44.533	11:15:42.219	57.686679
2	16:11:43.289	16:13:46.464	123.174840
3	19:58:56.821	19:59:59.721	62.899860
4	14:18:21.339	14:18:52.021	30.681835
5	19:37:07.299	19:38:08.706	61.407237
6	12:03:56.351	12:04:50.433	54.082146
7	18:25:34.011	18:26:06.263	32.252218
8	18:04:22.087	18:05:06.153	44.066042
9	19:45:12.793	19:45:40.740	27.946275
10	11:44:15.617	11:44:36.356	20.739469

Table A.29: Participant timings for the trial type: living room shelf in dynamic, using RVIZ.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:05:18.371	11:05:38.132	19.761350
2	16:08:48.586	16:10:51.170	122.583803
3	20:08:52.770	20:09:12.193	19.422327
4	14:15:24.700	14:16:16.739	52.039006
5	19:39:52.115	19:40:14.249	22.133607
6	12:02:04.915	12:02:39.262	34.347539
7	18:26:44.267	18:27:04.188	19.921715
8	17:56:37.607	17:57:22.807	45.200317
9	19:44:06.252	19:44:24.193	17.941200
10	11:39:42.512	11:40:15.365	32.852620

Table A.30: Participant timings for the trial type: bed table in stationary, using RVIZ.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:06:55.363	11:07:24.714	29.350980
2	/	/	/
3	20:09:35.039	20:10:31.922	56.883792
4	14:17:06.247	14:17:49.277	43.030034
5	19:39:02.903	19:39:28.367	25.464182
6	12:00:15.409	12:01:07.809	52.399852
7	18:27:47.396	18:29:46.786	119.390494
8	17:58:09.324	17:58:25.810	16.486477
9	19:42:47.100	19:43:38.082	50.982825
10	11:41:02.655	11:41:22.398	19.742717

Table A.31: Participant timings for the trial type: bed table in dynamic, using RVIZ.

Pen

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:58:35.272	12:00:08.322	93.049754
2	16:49:04.140	16:51:11.410	127.270088
3	19:37:10.772	19:39:11.282	120.510186
4	13:57:29.126	13:58:50.342	81.215286
5	20:24:39.712	20:26:56.302	136.589809
6	13:40:41.519	13:42:07.062	85.543256
7	18:59:31.745	19:00:09.925	38.179450
8	17:40:31.202	17:41:25.665	54.462978
9	20:10:21.964	20:11:16.588	54.624164
10	11:32:12.377	11:33:41.743	89.365166

Table A.32: Participant timings for the trial type: kitchen table in stationary, using the XR pen.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	12:00:53.367	12:01:27.150	33.783066
2	16:39:32.543	16:41:19.640	107.097400
3	19:34:33.973	19:36:34.411	120.438609
4	13:55:53.316	13:56:29.666	36.349934
5	20:18:55.019	20:19:34.577	39.557968
6	13:33:20.392	13:35:35.290	134.898306
7	19:00:38.685	19:01:09.752	31.067187
8	17:42:03.944	17:42:40.678	36.733925
9	20:11:50.170	20:12:21.650	31.480290
10	11:34:30.425	11:34:57.183	26.758101

Table A.33: Participant timings for the trial type: kitchen table in dynamic, using the XR pen.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:44:42.432	11:45:11.835	29.402944
2	16:37:09.110	16:38:56.492	107.382027
3	19:51:12.019	19:51:48.064	36.044586
4	13:54:27.037	13:54:56.771	29.733776
5	20:17:05.068	20:17:34.425	29.356946
6	13:30:59.753	13:31:42.452	42.698593
7	18:58:22.411	18:58:54.153	31.742357
8	17:37:05.392	17:37:51.608	46.215771
9	20:01:32.378	20:02:20.657	48.279008
10	11:31:16.183	11:31:30.818	14.635320

Table A.34: Participant timings for the trial type: living room shelf in stationary, using the XR pen.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:46:00.299	11:46:30.251	29.952596
2	16:26:38.893	16:27:59.901	81.007978
3	19:52:35.176	19:53:30.038	54.861869
4	13:52:18.706	13:52:54.563	35.857358
5	20:17:59.576	20:18:19.929	20.353233
6	13:32:25.443	13:32:48.097	22.653951
7	18:57:28.184	18:57:47.996	19.812313
8	17:36:03.004	17:36:41.204	38.199883
9	19:58:55.913	20:00:58.485	122.571560
10	11:30:12.268	11:30:55.029	42.760971

Table A.35: Participant timings for the trial type: living room shelf in dynamic, using the XR pen.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:48:15.308	11:49:35.068	79.759617
2	16:17:32.346	16:19:48.042	135.696163
3	19:32:06.263	19:32:25.712	19.448724
4	13:59:28.253	14:00:04.744	36.490884
5	20:10:58.323	20:11:19.659	21.335947
6	13:30:16.851	13:30:29.677	12.825655
7	18:56:12.214	18:56:55.564	43.349796
8	17:52:35.893	17:52:53.943	18.050066
9	20:21:06.528	20:22:03.300	56.772449
10	11:27:24.857	11:28:57.100	92.242996

Table A.36: Participant timings for the trial type: bed table in stationary, using the XR pen.

participantID	startTime (hh:mm:ss.ms)	finishTime (hh:mm:ss.ms)	duration (s)
1	11:47:26.245	11:47:56.191	29.945398
2	/	/	/
3	19:31:06.541	19:31:46.515	39.97479
4	14:00:54.283	14:01:37.248	42.964789
5	20:15:29.197	20:16:18.078	48.881035
6	13:29:13.518	13:29:48.691	35.173249
7	18:55:21.303	18:55:41.749	20.446836
8	17:45:32.610	17:46:41.577	68.966778
9	20:18:27.766	20:20:36.538	128.772055
10	11:29:15.444	11:29:29.982	14.538198

Table A.37: Participant timings for the trial type: bed table in dynamic, using the XR pen.

Bibliography

- [1] H. Ishiguro, “The Realisation of an Avatar-Symbiotic Society Where Everyone can Perform Active Roles without Constraint,” *RSJ Advanced Robotics (AR)*, vol. 35, no. 11, pp. 650–656, Jun. 2021. [Online]. Available: <https://doi.org/10.1080/01691864.2021.1928548>
- [2] J. A. Gonzalez-Aguirre, R. Osorio-Oliveros, K. L. Rodríguez-Hernández, J. Lizárraga-Iturralde, R. Morales Menendez *et al.*, “Service robots: Trends and technology,” *Applied Sciences*, vol. 11, no. 22, p. 10702, 2021.
- [3] R. Palmarini, I. F. del Amo, G. Bertolino, G. Dini, J. A. Erkoyuncu *et al.*, “Designing an ar interface to improve trust in human-robots collaboration,” *Procedia CIRP*, vol. 70, pp. 350–355, 2018.
- [4] G. A. Zachiotis, G. Andrikopoulos, R. Gornez, K. Nakamura, and G. Nikolakopoulos, “A survey on the application trends of home service robotics,” in *2018 IEEE international conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 1999–2006.
- [5] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt, “Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces,” in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–33.
- [6] F. Kennel-Maushart, R. Poranne, and S. Coros, “Interacting with multi-robot systems via mixed reality,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 633–11 639.
- [7] A. Cleaver, R. M. Aronson, and J. Sinapov, “Enhancing users’ predictions of robotic pouring behaviors using augmented reality: A case study,” in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. IEEE, 2024, pp. 1886–1891.

- [8] M. Walker, H. Hedayati, J. Lee, and D. Szafir, “Communicating robot motion intent with augmented reality,” in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 316–324.
- [9] C. Tornberg, L. E. Hafi, P. M. U. Eljuri, M. Yamamoto, G. A. G. Ricardez *et al.*, “Mixed reality-based 6d-pose annotation system for robot manipulation in retail environments,” in *2024 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2024, pp. 1425–1432.
- [10] R. Luo, M. Zolotas, D. Moore, and T. Padir, “User-customizable shared control for robot teleoperation via virtual reality,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 196–12 203.
- [11] J. Nakanishi, S. Itadera, T. Aoyama, and Y. Hasegawa, “Towards the development of an intuitive teleoperation system for human support robot using a vr device,” *Advanced Robotics*, vol. 34, no. 19, pp. 1239–1253, 2020.
- [12] S. K. Tadeja, T. Zhou, M. Capponi, K. Walas, T. Bohné *et al.*, “Using augmented reality in human-robot assembly: A comparative study of eye-gaze and hand-ray pointing methods,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 8786–8793.
- [13] D. Yu, T. Dingler, E. Velloso, and J. Goncalves, “Object selection and manipulation in vr headsets: Research challenges, solutions, and success measurements,” *ACM Computing Surveys*, vol. 57, no. 4, pp. 1–34, 2024.
- [14] S.-K. Ong, A. Yew, N. K. Thanigaivel, and A. Y. Nee, “Augmented reality-assisted robot programming system for industrial applications,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101820, 2020.
- [15] D.-M. Pham and W. Stuerzlinger, “Is the pen mightier than the controller? a comparison of input devices for selection in virtual and augmented reality,” in *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*, 2019, pp. 1–11.
- [16] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote *et al.*, “ROS: An Open-Source Robot Operating System,” in *Proceedings of 2009 IEEE Workshop on Open Source Software*, Kobe, Japan, May 2009. [Online]. Available: <http://www.willowgarage.com/papers/ros-open-source-robot-operating-system>
- [17] S. Hart, “Development of nasa-tlx (task load index): Results of empirical and theoretical research,” *Human mental workload/Elsevier*, 1988.

- [18] J. Brooke, “SUS: A ‘Quick and Dirty’ Usability Scale,” in *Usability Evaluation in Industry*. Taylor & Francis, Jun. 1996, pp. 189–194. [Online]. Available: <https://doi.org/10.1201/9781498710411>
- [19] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, “Augmented reality: A class of displays on the reality-virtuality continuum,” in *Telem manipulator and telepresence technologies*, vol. 2351. Spie, 1995, pp. 282–292.
- [20] X. Xu, H. Zhang, H. Zeng, A. Song, Y. Xing *et al.*, “Combination of augmented reality based brain-computer interface and eyetracking for control of a multi-robot system,” in *Proceedings of the 2024 4th International Conference on Robotics and Control Engineering*, 2024, pp. 90–95.
- [21] R. Likert, “A technique for the measurement of attitudes,” *Archives of Psychology*, 1932.
- [22] E. Sheetz, M. Kian, M. Savchenko, S. J. Jorgensen, E. Laske *et al.*, “Comparing human input in rviz and virtual reality interfaces for 3d object manipulation tasks,” in *First International Workshop on Horizons of an Extended Robotics Reality at IEEE IROS 2022*, 2022.
- [23] V. Ortenzi, M. Filipovica, D. Abdllkarim, T. Pardi, C. Takahashi *et al.*, “Robot, pass me the tool: Handle visibility facilitates task-oriented handovers,” in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2022, pp. 256–264.
- [24] R. Newbury, A. Cosgun, T. Crowley-Davis, W. P. Chan, T. Drummond *et al.*, “Visualizing robot intent for object handovers with augmented reality,” in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2022, pp. 1264–1270.
- [25] C. Salvato, V. Bomnale, J. Wang, Y. Xiang *et al.*, “iteach: Interactive teaching for robot perception using mixed reality,” *arXiv preprint arXiv:2410.09072*, 2024.
- [26] L. Zhang, D. Liang, M. Li, W. Yang, and S. Yang, “Coalition formation game approach for task allocation in heterogeneous multi-robot systems under resource constraints,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 3439–3446.

List of Figures

1.1	A UR5 from Universal Robotics used as a collaborative robot on production lines in a car factory (reference: Picture link).	11
2.1	Several examples of visual effects in Augmented Reality for Robotics, from [5].	15
2.2	Different hardware placement for Augmented Reality in Robotics, from [5].	16
2.3	A presentation of several types of interaction techniques of Augmented Reality in robotics, from [5].	16
2.4	4 different AR-based visual effects to improve human-robot collaboration in the case of an aerial robot, proposed by the paper [8].	17
2.5	Experimental setup proposed by the paper [8].	17
2.6	The three interfaces proposed by this paper [14] to program an industrial manipulator robot.	18
2.7	XR-based control interface using eye-gaze, proposed by [12].	19
2.8	XR-based control interface using hand-ray pointing, proposed by [12].	19
2.9	The three different tasks used by [6] to evaluate their interface.	20
2.10	Experimental setup to evaluate pointing performances of several devices, from [15].	22
3.1	System overview of the proposed approach.	26
4.1	Overview diagram of the implementation details for the proposed approach.	33
4.2	Screenshot of the RVIZ software when a target is selected, making the orientation arrow appear (picture shown to the participants during a tutorial video).	34
4.3	XR-based control interface using the XR motion controllers.	35
4.4	XR-based control interface using the operator's hands.	35
4.5	XR-based control interface using the XR pen.	36

5.1	Experiment scene: a reproduction of a daily living environment. . .	38
5.2	Third object location: the kitchen table.	38
5.3	Third object location: the living room shelf.	38
5.4	Third object location: the bed table.	38
5.5	Different stages of the experiment.	39
5.6	Visual representation of all the trial types tested with each participant.	40
6.1	Box plot of the TLX ratings and their standard deviations, for the different devices, rated on a scale from 1-20.	46
6.2	Scale used to evaluate the obtained usability score for the SUS questionnaire, from [18].	51
6.3	Histogram of the average SUS score for each control device.	52
6.4	Box plots of the task completion times of each control device, for each object location.	54
6.5	Box plots of the task selection times of each control device, for each object location.	55
6.6	Box plots of the task waiting times of each control device, for each object location.	57
6.7	Box plots of the number of navigation requests sent for each control device, for each object location.	58
6.8	Box plots of the mean time differences between navigation requests, for each control device and each object location.	59

List of Tables

6.1	p-values of the different NASA-TLX loads in function.	47
6.2	Average and Variance of the NASA-TLX ratings for the frustration, for each control device, obtained as part of the single-factor ANOVA analysis.	47
6.3	Average and Variance of the NASA-TLX ratings for the mental demand, for each control device, obtained as part of the single-factor ANOVA analysis.	48
6.4	Average and Variance of the NASA-TLX ratings for the physical demand, for each control device, obtained as part of the single-factor ANOVA analysis.	48
6.5	Average and Variance of the NASA-TLX ratings for the temporal demand, for each control device, obtained as part of the single-factor ANOVA analysis.	49
6.6	Average and Variance of the NASA-TLX ratings for the performance, for each control device, obtained as part of the single-factor ANOVA analysis.	49
6.7	Average and Variance of the NASA-TLX ratings for the effort, for each control device, obtained as part of the single-factor ANOVA analysis.	50
6.8	Average and Variance for the raw workload index of each control device, obtained as part of the single-factor ANOVA analysis. . . .	50
6.9	Average and Variance of the participants' SUS Scores, for each control device, obtained as part of the single-factor ANOVA analysis.	51
6.10	Participants' answers to the global questions concerning the whole experiment.	53
6.11	Average and Variance of the task completion time, for each control device, obtained as part of the single-factor ANOVA analysis. . . .	53
6.12	Average and Variance of the task completion time, for each object location, obtained as part of the single-factor ANOVA analysis. . .	54
6.13	Average and Variance of the task selection time, for each control device, obtained as part of the single-factor ANOVA analysis. . . .	56

6.14	Average and variance of the task waiting time, for each control device, obtained as part of the single-factor ANOVA analysis.	56
6.15	Average and variance of the number of navigation requests sent, for each control device, obtained as part of the single-factor ANOVA analysis.	58
6.16	Average and variance of the number of navigation requests sent, for each object location, obtained as part of the single-factor ANOVA analysis.	58
6.17	Average and variance of the mean time difference between navigation requests, for each control device, obtained as part of the single-factor ANOVA analysis.	59
A.1	Participants' answers to the NASA-TLX questionnaire for the RVIZ software.	74
A.2	Participants' answers to the NASA-TLX questionnaire for the XR motion controllers.	75
A.3	Participants' answers to the NASA-TLX questionnaire for the hands.	75
A.4	Participants' answers to the NASA-TLX questionnaire for the XR pen.	76
A.5	SUS questionnaire for RVIZ: first part.	76
A.6	SUS questionnaire for RVIZ: second part.	77
A.7	SUS questionnaire for XR Motion controllers: first part.	77
A.8	SUS questionnaire for XR Motion controllers: second part.	78
A.9	SUS questionnaire for Hands: first part.	78
A.10	SUS questionnaire for Hands: second part.	79
A.11	SUS questionnaire for Pen: first part.	79
A.12	SUS questionnaire for Pen: second part.	80
A.13	Global questionnaire about the whole experiment.	80
A.14	Participant timings for the trial type: kitchen table in stationary, using XR motion controllers.	81
A.15	Participant timings for the trial type: kitchen table in dynamic, using XR motion controllers.	81
A.16	Participant timings for the trial type: living room shelf in stationary, using XR motion controllers.	82
A.17	Participant timings for the trial type: living room shelf in dynamic, using XR motion controllers.	82
A.18	Participant timings for the trial type: bed table in stationary, using XR motion controllers.	83
A.19	Participant timings for the trial type: bed table in dynamic, using xR motion controllers.	83

A.20 Participant timings for the trial type: kitchen table in stationary, using the hands.	84
A.21 Participant timings for the trial type: kitchen table in dynamic, using the hands.	84
A.22 Participant timings for the trial type: living room shelf in stationary, using the hands.	85
A.23 Participant timings for the trial type: living room shelf in dynamic, using the hands.	85
A.24 Participant timings for the trial type: bed table in stationary, using the hands.	86
A.25 Participant timings for the trial type: bed table in stationary, using the hands.	86
A.26 Participant timings for the trial type: kitchen table in stationary, using RVIZ.	87
A.27 Participant timings for the trial type: kitchen table in dynamic, using RVIZ.	87
A.28 Participant timings for the trial type: living room shelf in stationary, using RVIZ.	88
A.29 Participant timings for the trial type: living room shelf in dynamic, using RVIZ.	88
A.30 Participant timings for the trial type: bed table in stationary, using RVIZ.	89
A.31 Participant timings for the trial type: bed table in dynamic, using RVIZ.	89
A.32 Participant timings for the trial type: kitchen table in stationary, using the XR pen.	90
A.33 Participant timings for the trial type: kitchen table in dynamic, using the XR pen.	90
A.34 Participant timings for the trial type: living room shelf in stationary, using the XR pen.	91
A.35 Participant timings for the trial type: living room shelf in dynamic, using the XR pen.	91
A.36 Participant timings for the trial type: bed table in stationary, using the XR pen.	92
A.37 Participant timings for the trial type: bed table in dynamic, using the XR pen.	92

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl