TU WIEN Informatics

Laborautomatisierung: Benutzerschnittstelle für die Teleoperation eines Mobilen Manipulators

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Media and Human-Centered Computing

eingereicht von

Janek Janßen, B.Sc.
Matrikelnummer 51829460

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Markus Vincze
Mitwirkung: Ádám Wolf, M.Sc.

Wien, 31. August 2022

Janek Janßen   Markus Vincze

TU WIEN Informatics

Lab-Automation: Building a User-Interface for a teleoperated Mobile Robot platform

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Media and Human-Centered Computing

by

Janek Janßen, B.Sc.
Registration Number 51829460

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Markus Vincze
Assistance: Ádám Wolf, M.Sc.

Vienna, 31st August, 2022

Janek Janßen   Markus Vincze

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Declaration of Authorship

Janek Janßen, B.Sc.

I hereby declare that I have written this work independently, have given full details of the sources and aids used, and have marked places in the work—including tables, maps and illustrations—which are taken from other works or from the Internet, either verbatim or in spirit, as borrowed, in any case indicating the source.

Vienna, 31st August, 2022

Janek Janßen

Kurzfassung

Es wurde ein Virtual-Reality-System entwickelt, um die Machbarkeit der Virtual-Reality-Teleoperation von mobilen zweibeinigen Robotern in einer Laborumgebung zu testen. Unter Verwendung eines Standalone-Headsets (Quest2), eines ROS-basierten Roboters (Tiago++) und der Unity Game Engine zeigte dieser Prototyp vielversprechende Ergebnisse im Hinblick darauf, wie diese Software-Pipeline in Zukunft eingesetzt werden könnte. Der Prototyp wurde mit Hilfe der iterativen Prototyping-Methode entwickelt und durch Benutzertests evaluiert. Die Testergebnisse zeigten, dass unerfahrene Benutzer in der Lage waren, den Roboter ohne vorheriges Training zu bedienen, und dass die negativen Auswirkungen einer längeren Verwendung von VR-Headsets durch die Verwendung des Homunculus-Steuerungsmodells weitgehend ausgeglichen werden konnten. Das Homunculus-Modell wurde erweitert und ein neues Steuerungsmodell, das Augmented Humunculus Model (AHM), wurde konzipiert und realisiert.

Abstract

A virtual reality system is developed to test the feasibility of virtual reality teleoperation of mobile bipedal robots in a laboratory environment. Using a standalone headset (Quest2), a ROS based robot (Tiago++) and the Unity Game Engine, this prototype showed promising results in regards of how this software pipeline could be used in the future. The prototype was developed using the iterative prototyping method and was evaluated through user tests. The tests results showed that Novice users were able to teleoperate the robot without prior training, negative effects of prolonged VR headset use could be mostly negated through the use of the Homunculus control model. The Homunculus model was expanded upon and a new control model, the Augmented Humunculus Model (AHM) was conceptualized and realized.

Contents

CHAPTER 1

Introduction

Seamlessly teleoperating a mobile robot from a remote location has been a goal for researchers spanning back decades. The dawn of commercially available Extended Reality (XR) technologies might enable a paradigm shift by the fusing XR technologies with robotics. One field of research where mobile robotics are gaining in importance is laboratory automation in the pharmaceutical sector, which has distinctly different prerequisites for enabling mobile robotics compared to other sectors conventionally closer to automation. Non linear processes and changing circumstances drive the need for adaptable robots. Teleoperation is an important step to enable fully automated laboratories.

1.1 Vision

Compared to other industries - most of which are currently outperforming pharmaceutical production in their level of automation - laboratory automation has not experienced widespread adaptation yet. To paint a picture of the situation, a recent study concludes that 89% of biomedical research papers are using methods that could already be automated using existing solutions [GC17]. This suggests that the lack of adaption has manifold reasons and cannot be pinpointed to be based in the unavailability of technologies. This lack of adaptation has to be overcome to uncover unfulfilled potential, since laboratory automation provides considerable benefits such as reproducibility of laboratory research [Kit+19], reduction in human errors, safety and efficiency [HD20].

One of the most integral parts of automation are robots. Stationary robots are already in widespread use to automate low level processes in laboratories, but are not interconnected to enable sophisticated chains of processes to work together. A human still has to operate between robots and other laboratory equipment. This is exactly the point at which mobile robots become of interest, because processes are not linearly connected automation cannot be organized as an assembly line. Mobile robots can move the product from one point to multiple different points and be controlled via a scheduling software and take over other manual tasks usually done by humans. This dynamic organization of laboratories offers an environment with multiple use cases for mobile robots. As a first step towards laboratory automation, this thesis is concerned with enabling and researching the ability to teleoperate a mobile robot in laboratory environment. Completely enabling autonomous mobile robots will not be achieved in the near future [HD20], which is why teleoperation is of immediate value and will stay of value in the future - not only intrinsically - but also for enabling easy demonstration capabilities and robot teaching. A user friendly way of precisely controlling a mobile robot is an essential enabling technology for future paths in laboratory automation. If an automated process does not immediately present itself with an acceptable success rate, teleoperation can bridge early development set backs and continue to uphold the trust in the process. Specifically, development of automation solutions concerning mobile robots, which would otherwise been abandoned due to their failure rate will be seen through, because teleoperation can work as a backup solution. Fully automated processes always need backup solutions, otherwise a single point of failure would end operation entirely. In current laboratory environments the backup solution is a human operator acting in the same space as the robot. Teleoperation completely eliminates the necessity to design the workspace for human needs in the future. Instead of designing the robots to work around environments created or humans, laboratories will be designed for robots while humans can control the robot when necessary.

1.2 Challenge

To achieve this vision multiple hurdles will have to be overcome. There is currently neither standardized software nor hardware for teleoperation. Since teleoperation can simply be defined extending the human capability to manipulate remotely [HS06], this could mean anywhere between mechanical linkages to operating a mars rover from earth. Unsurprisingly, this leads to closed and highly specialized system depending on the use-case. To specify, this thesis examines the ability to teleoperate over the internet, which tends to be the prevailing scenario in applicable domains such as telesurgery or industrial applications. Also, we are concerned with bilateral teleoperation, communication in both directions, see figure 1.1.

Furthermore, there is an array of input and output devices on the operators end that could be used for bilateral teleoperation and equally so there are a plethora of robot configurations on the other end. The choice of which is not only determined by distance between operator and robot, but also by the expertise level of the operator, the number of different operators and costs. All but the most basic devices come with a considerable cost, which is why the researchers hand is mostly forced on which devices to work with. In this case the mobile robot platform used is a Tiago++ bi-manual robot developed by Pal Robotics [PAL19]. This mobile manipulator is controlled through an on-board computer running Ubuntu and the Robot Operating System (ROS) [ROSa]. Despite its name ROS is no operating system but an open-source competitor to proprietary software

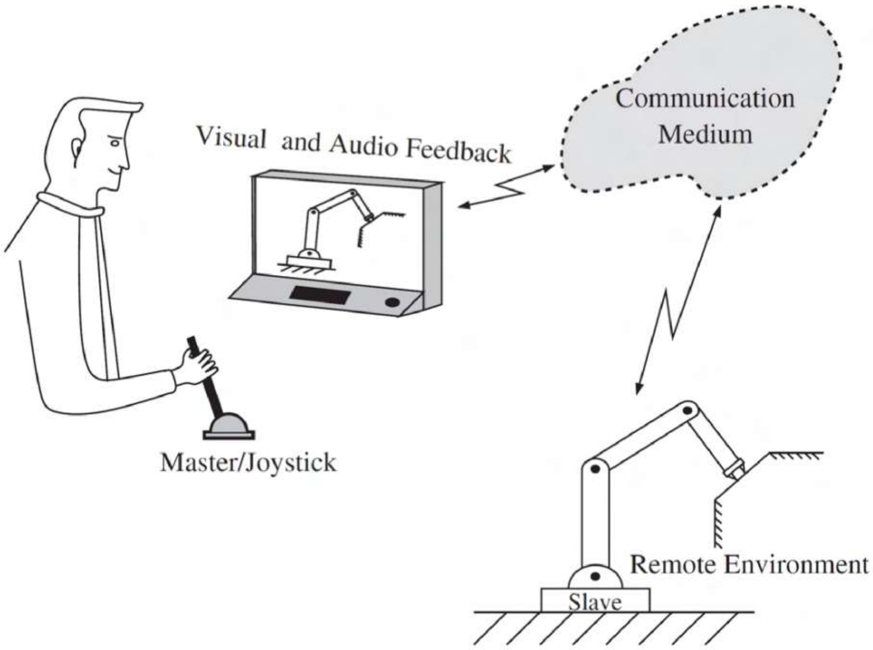

Figure 1.1: Bilateral teleoperation [HS06].

solutions for robot programming. Being ROS based has the advantage of developed solutions being applicable to other robot configurations running ROS and lessening the aforementioned problem of researchers working with different hardware.

It comes with another upside concerning the hardware chosen to track the operators input and communicate the output generated by the robot to the operator. Virtual Reality (VR) Headsets are most often used in a video game context, therefore the most convenient way to interface with them is a video game engine such as Unity [Unia]. Since the release of the Unity Robotics Hub (URH) [Unib], Unity is able to communicate with ROS, which enables the complete integration between VR headset and robot.

The VR headset used is an Oculus Quest 2 [Ocu], which offers multiple advantages over other headsets for this use case. First and foremost it is a stand-alone headset, allowing direct communication between the headset and the robot without the need of a connected computer. Secondly, it uses inside out tracking and works without installing additional tracking beacons in the environment it is used in.

The challenge lies within creating a prototype that is capable of interfacing between a standalone VR headset and a mobile bipedal robot, that is capable to be used for manipulation tasks by non expert users. This thesis should serve as a blueprint on how to use the chosen technologies to develop such a prototype and evaluate the current state of technology to enable future researchers to further explore possibilities regarding VR and mobile robotics. The usability of the user interface is of central importance to allow

Figure 1.2: Tiago++, Pal Robotics [Vil19].

a broader spectrum of operators and researchers access to teleoperation. Other research as well as implementations in generally focus on one or more expert users teleoperating a single robot for a specific use case (e.g. [Kla+20], [Che+21]). To develop an interface with little to no overhead that allows non expert users to teleoperate is another challenge entirely.

The studies overarching goal is concerned with the feasibility of the proposed solution, to determine if what is proposed can work and which capabilities it fulfills and which it lacks. The technologies needed to develop a VR interface to control a mobile robot are all available in principle, it is the researchers aim to evaluate how compatible they are and if they can be made to work together. Other solutions based on similar technologies will be explored in section 2.2. As a conclusive goal, a picture of what is possible will form that highlights which steps will need to follow to develop a mature solution that can withstand the actual use case of a working teleoperated mobile robot in laboratory robustly.

Regarding the choice of using a VR headset, this could be seen as an arbitrary choice of device if researching teleoperation in general. However, VR headsets do have many characteristics that make them particularly interesting to be researched, this is further explored in subsection 2.2.4. Which is why this thesis is concerned exclusively with VR headsets as input/output devices to evaluate the potential of this technology. As VR headsets also come with some negative characteristics, that will have to be negated as much as possible.

The use case has influence over the research goals, since the robot is bound to physically be in the laboratory, as well as feedback during the process coming from personnel of a pharmaceutical company. This will supposedly work in favor for this specific solution but might effect generalisability to other settings, this bias is recognized part of the research and is neither actively embraced nor resisted. Nonetheless the solution should easily be adaptable to other settings.

From these goals the following research questions have been identified that will guide the following research process:

- How well is the implemented prototype suited to control a mobile robot in laboratory environment?
- How well can the negative effects of a VR headset used in teleoperation be negated through the developed solution?
- What are the missing links to enable robust, usable teleoperation through a VR headset?

1.3 Contributions

In accordance to these research questions, a multitude of contributions have been achieved, these range from a comprehensive overview about the current environment of VR teleoperation of mobile robots, to a functioning prototype usable even by inexperienced operators. Additionally, the work provides a blueprint on how to construct this prototoype, as well as a new control mode for VR teleoperation, the Augmented Homunculus Model (AHM). On top of this user tests with 13 participants were conducted to validate the performance and usability of AHM in actual use, the analysis of which generated data that makes the prototype comparable to other studies.

Out of these contributions, most centrally, the theoretical basis of VR teleoperation was extended upon by the AHM control mode introduced and developed in this work. AHM is a complete control interface using a VR headset allowing the operator to teleoperate a bipedal mobile robot. This includes manipulation, observation and locomotion. These tasks have been shown to be achieved by novice operators with minimal adjustment time. The known side effects of prolonged VR use, especially while using VR teleoperation, have been shown to not apply to almost all operators using AHM.

Thus, AHM was not only conceptualized but also realised and tested. The results of the user tests as well as the data generated by it are a significant part of this works contributions towards making VR teleoperation more accessible. The user, guide that was created to ease new operators into being able to learn how to use the system, as well as the user interface design, enabled non expert users to teleoperate a mobile robot without any prior knowledge and experience. AHM is therefor a leap in usability for VR teleoperation, which was no central aim of prior research. This usability can be achieved for a exceptionally low price point on the operators end. A consumer grade standalone VR headset is the only hardware needed. This opens up the possibility for multiple operators teleoperating the robot from different locations without the need to install particular setups at the operators location.

As a consequence of this research a tangible artefact that will act as a basis for future research is now available. Development used the iterative prototyping method described by Beaudouin-Lafon and Mackay [BM12]. Between and during iterations the prototype was subject to constant feedback from the researcher as well participants of user tests. This feedback drove the prototype development up to a point were the feasibility of the study was shown. The resulting artifact extends on previous research regarding VR teleoperation. The use of a stand-alone VR headset as well as the laboratory use case and the bipedal mobile robot are mostly novel in this field on their own, more so in combination. The successful use of the iterative prototyping method established the method in VR teleoperation and therefore allows other researchers to cross over into other fields.

Furthermore, the artefact was tested by novice users to identify to what degree they are capable of solving teleoperation tasks. The generated data as well as the participants' answers to the following questionnaire was put into context with knowledge gathered throughout development through interviews. Based on these empiric findings the research questions regarding usability, likeability and identification of missing links towards usable VR teleoperation were advanced.

Beyond that the established VR teleoperation control models were extended upon and potential for a novel fusion of models was identified. Constructing a complete teleoperation system for a mobile bipedal robot usually focuses around a single operator achieving the highest possible precision neglecting usability, simplicity and cost. Shifting the focus towards usability and accessibility is a new perspective that enabled teleoperation to enter into a much wider spectrum of use cases.

It has been shown that novice users can teleoperate a mobile bipedal robot, achieving manipulation tasks all while using consumer grade hardware. These users reported high enjoyment and intuition ratings and no disorientation. From these findings the AHM and VR teleoperation in general is not only competitive in performance to other teleoperation solutions, but it offers upsides that are hugely advantageous.

1.4 Results preview

The developed artefact was able to demonstrate its capabilities in the hands of novel users. Almost all were able to complete locomotion tasks inside a laboratory environment. More complex manipulation task could be solved by some, without any training beforehand. The computing performance of the standalone headset was sufficient for this application. Users generally rated usability and likeability positively. The general user interface chosen was deemed very suitable. One factor holding the prototype back was a lack of robustness. However this does not compromise the study in principle and can be improved for future studies. Potential improvements for the user interface in general and the AHM specifically where identified.

1.5 Thesis outline

To establish the context and background of the thesis, the following chapter 2 on page 9 will cover all relevant areas. Lab automation, the umbrella field, as well as mobile robotics and Mixed Reality (MR) are introduced. Then the current state of the art (section 2.2, page 11) of VR controlled robots, the theoretical background and current day examples are presented to manifest what research has been done up to this point, how this topic has come to be historically and what conventions are already established. This also specifically covers the control models for VR teleoperation (section 2.2.3, page 13), from which the AHM is derived.

After presenting the related work and context, the succeeding chapter will follow the implementation of the artefact in two steps. First of all, it starts with an analysis of potential software and hardware components that where identified to be capable to be used for the purposes of the thesis. It is shown why the choices were made and also which other choices could have been made. The second step will present the implementation of the prototype in such detail that readers can comprehend what was done and why it was done.

The following analysis of the user tests is split in multiple parts according to the method used to acquire the data. The results of the analysis are then used to formulate potential improvement that can be made to the prototype.

Finally, the thesis concludes with a summary that illustrates potential future developments in VR teleoperation, followed by an conclusion that evaluates the results in context of the research goals.

CHAPTER 2

Related work

To examine the current state of teleoperating mobile robot through the use of VR headsets in a laboratory environment, multiple domains will have to be explored. This chapter will put the thesis into the context of lab automation, which in turn will be put in context to mobile robotics and the relation between mobile robots and MR. Further on, the current state of the art and present day practices and challenges will be displayed after a brief introduction to the history of teleoperating mobile robots. Finally, the current day challenges will be presented and recent examples of recent prototype developments will be shown.

2.1 Background

The research field of teleoperated mobile robotics intersects with multiple other research fields which has been truncated in the Introduction 1. Through the laboratory setting an additional research area is relevant. In order to establish the overall context the following will illuminate the related fields and tie them into this research.

2.1.1 Lab automation

Lab-Automation can be considered the umbrella field for this research and a reason why teleoperation is of interest in laboratories in general. The key thought being that a teleoperated mobile robot acts as a in between step to the goal of fully automated laboratories. Without going into the benefits of automation, which are similar to other industries, the following will highlight the issues holding back lab automation adaptation and the importance of mobile robots to counteract them.

One of the aspects holding back the level of automation is a lacking standard for lab instrument integration, however efforts are being made to counteract this situation. As such, Standardization in Lab Automation (SiLA), a not-for-profit open consortium is developing an open standard for lab device communication [Sil]. Other limitations include the risk of incorrect application, that potentially results in less efficiency [Zie+14], innovation inhibition where already automated processes are less likely to be changed [HD20] and resistance by the workforce [Aut15], but these are not domain specific and research in this regard is plentiful.

Regarding lab automation specifically, the central point in the way of adaptation can be identified as an interplay between cost, equipment obsolescence and changing procedures. Whereas in other applications in automation processes are linear, predictable and calculable so that total costs can be planned for, life science research labs require flexibility because of changes that are not foreseeable. A factory can justify investing into automation that will yield expected profit in a certain amount of time. A research lab is unable to predict direct profits, especially when it is unable to predict how long the equipment will last before being obsolete [HD20]. This situation is especially emphasized through the existence of widespread automation of processes in labs that are not suspect to change and that can be easily automated such as liquid handling (liquid handling robots) and sample transportation (Selective Compliance Assembly Robot Arm (SCARA)) [CB15]. Overall, adoption of automation in lab scenarios will probably be slower than what is desirable, since most of the discussed limitations are inherent to the field and high level automation solutions will only be attainable by the wealthiest of laboratories. In this context, low level automation is a desirable interim goal, where certain fixed processes are automated, eliminating the need for repetitive tasks by humans. Leading to an incremental increase in automation over time. These low level solutions are partially created in-house by laboratories, by adapting commercially available hard- and software to their needs, because available solutions are either too expensive or do not fit for their specific use-case. The necessity of this is a general indicator about the state of lab automation [HD20].

2.1.2 Mobile robotics

These circumstances have led to different robotic solutions being used for automation, most of them being stationary. Generally, one has to differentiate between mobile robots and mobile manipulators because mobile only refers to the robot not being stationary, i.e. vacuum robots and most logistics robots fall into this category. Mobile manipulators integrate one or more robotic arms onto the mobile platform, which allows for manipulation of objects and handling of devices without them being conceptualized and build with robotics in mind.

Mobile manipulators in life science laboratories have become subjects of research only recently, with case studies literature being available as well as some commercial mobile manipulators that are able to be controlled through scheduling software (See section 2.2).

In laboratories, mobile manipulators can act as a bridge between different low level automation solutions. Mobile manipulators are inherently flexible and can be used in different scenarios. Most importantly in the context of this thesis is the topic of remote troubleshooting and error handling, the process of evaluating and fixing an error remotely without the need for direct human interaction. A example of this would be a process in a lab that is low level automated, where one of the devices responsible fails and needs to be restarted. Since the devices cannot be reset remotely an on-call employee has to travel on site to take care of the problem. A mobile manipulator can be used to resolve the error, saving time and cost.

Considering the strict regulations in the pharmaceutical industry [CB15] and the potential danger associated with working next to manipulators [GM20] as well as the challenge to develop a human machine interface capable of allowing precise movement while maintaining usability, remote error handling represents a challenge on multiple levels.

2.1.3 Mixed Reality

As mentioned before and further discussed in Subsection 2.2.4, the space of potential devices to display robot output and receive operator input is large and only becomes larger over time. However the focus of this research lies on VR headsets specifically. To specify the terminology for the oncoming passages, VR headsets are headsets most often used to display virtual environments. Opposite on Milgram's Reality-Virtuality Continuum Figure 2.1 lie real environments. Every mixture of these two is called MR, which means that the prototype developed also lies within the MR spectrum even though it uses a VR headset. Specifically the prototype produces an Augmented Reality (AR) experience, integrating real environment visual feedback into a VR setting. Depending on your viewpoint, you could also argue that the robot is an extension of virtuality into the real environment, making it Augmented Virtuality. Generally, teleoperation in this context might transcend the continuum defined by Milgram in 1994.

Figure 2.1: Milgram's Reality-Virtuality Continuum [Mil+95].

2.2 State of the art

Teleoperated mobile robots in life science labs are a very recent development. Therefore other related domains are consulted to construct an accurate picture of the current state of the art, after a brief history of the field of teleoperated mobile robotics outside of the life science domain is established.

2.2.1 History

Teleoperation and mobile robotics were traditionally two distinct fields. The first mechanically controlled teleoperator was build by Goertz in the mid 1940s to handle nuclear material [HS06], whereas Moravec proposed the first autonomous wheeled robot in 1977 [Han76]. The two fields consecutively merged, first for underwater exploration (1970s, 1980s) then space exploration (1990s) and then mobile robotics in 1999 [SJS99; SR99; Kaw+99]. Mobile robots are additionally found as military robots, rescue robots in unsafe environments and service robots, whereas teleoperation is used in medical operations [CHB07; HS06].

2.2.2 Present-day challenges

The field of teleoperation is primarily concerned with the interaction between human and machine. The first teleoperation systems where mechanically linked master-slave systems, the operator using the input controls (master) to control a manipulation output device (slave) (see Figure 1.1, 3). These systems avoided two central limitations of current telerobotics: Firstly, these system did not have to deal with latency and secondly the operator did not see the system as a different entity but as a tool. As the distance between master and slave increased so did the perception of the operator, as one could not see the slave directly but through displays, which inadvertently only generate a fraction of possible human sensory stimulation. Output as well as input devices massively improved over time, but they are inherently unable to overcome the limitation of latency.

Telemanipulation concepts: telepresence vs. tool approach

This development still remains relevant today as it culminates into two opposing working concepts: On the one hand, in a perfect world, an interface could be developed where the slave captures a perfect replica of all human sensory input while the master simultaneously captures the operators demands exactly, which in turn are executed perfectly by the slave. This creates the feeling for the operator of being inside the remote world. This concept is called telepresence and is expandable to the concept of teleexistance, where other people feel the presence of the operator in the remote world. Implementation of these concepts is however limited. When trying to achieve telepresence the goal is easy to formulate: Capture as much information about the remote world as possible and transfer this information as fast and accurately as possible to the operator and vice versa. This is however hindered through the existence of interfaces, as stimuli cannot be communicated directly. In both directions information is therefore always distorted [Che10].

On the other hand, the limitations created through interfaces and network communication speed can be accepted to be inevitable and instead of minimizing these limitations, efforts can be directed to solve the task at hand. Instead of trying to duplicate the operators perception into telepresence, the human-robot interaction can be developed so the robot is perceived as a tool as it did in the beginning of teleoperation. When the interface is perceived as a tool and not as one's extension of reality it is easier to accept discrepancies such as delay and distortion.

These two concepts have corresponding control paradigms that are primarily relevant to them. While telepresence allows *direct control* - the robot does exactly what the operator does - the tool concept allows for *semi-autonomous control*, meaning the operator can send a command which the robot then fulfills. It has to be mentioned at this point that these two hypothesis are not to be perceived on a spectrum or as counteracting forces per-se. One teleoperation solution can implement approaches of both or switch between the two during operation, which is not explored in this thesis, but was discussed during user tests and interviews. For a more in depth examination of the topic, see Bohren et al.'s paper "Do What I Intend, Not What I Do" [Boh+11].

Direct control / coupling effects

When using direct control, the distortions of sensory and motoric in- and output are not only of direct concern: Even when you can replicate the operators motions exactly onto the robot, this is still not enough to execute certain motoric actions. Motoric actions are cross related to sensoric input, when one of them is defective the other one is greatly compromised. Catching for example, is therefore extremely difficult using teleoperated robots. This so called coupling effects are to be considered when designing teleoperated systems. [Che10; CHB07].

Semi-autonomous control

To completely escape effects caused by sensory and motoric distortions, the robot can be controlled semi-autonomously. Instead of manually driving to a desired location, issuing a single command to move to the desired location has the potential to increase usability [Ann+19]. Similarly, instead of manually grabbing something, just issuing the command to grab is easier for the operator [BLB14]. These hybrid control schemes need computer vision technologies such as collision avoidance and object detection and thous bring different concerns with them [CHB07; Thu+19]. This type of control is therefore dependant on the accuracy and robustness of the robots autonomous perception and/or the the accuracy of pre-programmed actions. Semi-autonomous control is especially useful for routine, repeating tasks. This is mostly irrelevant for the use-case of remote-error handling, since errors will most likely not be identical but rather different from each other. However, if identical errors repeat and cannot be solved by a different solutions (i.e. opening doors or autonomously moving between rooms) semi-autonomous control is advisable.

2.2.3 Control models

The aforementioned control paradigms manifest themselves in different control models for VR teleoperation, which are always at least implicitly implemented in every VR teleoperation system developed. No naming convention has been established for these models, however they can be easily identified once one is aware of them. For the sake of unambiguousness this text is going to use the names used by Lipton et al. in one of their papers [LFR18]. Similar to the hypotheses not being mutually exclusive, these resulting models can be combined. One does not have to chose between them permanently, but rather switch according to circumstances [Bau+18]. Even though literature suggests that there is currently no system taken full advantage of switching between control models.

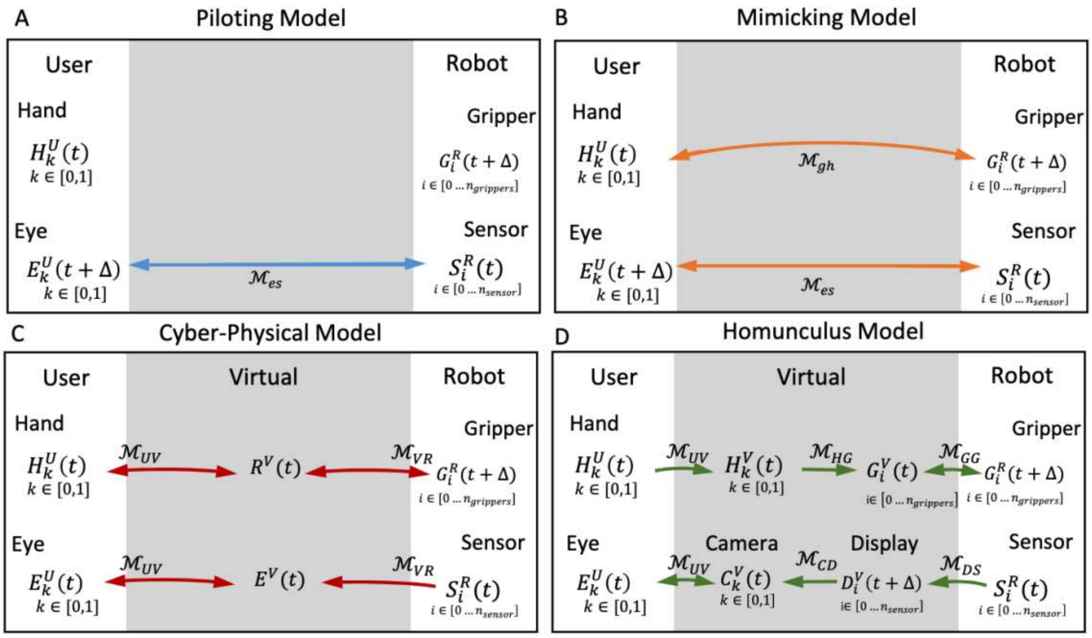

Figure 2.2: VR teleoperation control models [LFR18].

Figure 2.2 shows the control models. To emphasize their underlying differences up to three different spaces are shown for each. The user space U consists of the user's hands state (H_k^U) and eyes' state (E_k^U), while the robot space R consists out of the grippers' states (G_i^R) and sensors' state (S_i^R). The letters H,E,G,S correspond to Hand, Eyes, Grippers and Sensors while the the superscript (U,R,V) corresponds to the space (User, Robot or Virtual). The user space states have two elements [0,1] (two eyes and two hands), while the robot space states can have any number of grippers and sensors. The Cyber-Physical Model and Homunculus Model define an additional Virtual Space V that consists out of the virtual robotic system state (R^V) and the virtual environment state (E^V) for the Cyber-Physical Model. The Homunculus Model consists out of the virtual hands' state (H_k^V), the virtual grippers' state (G_i^V), the virtual camera's state (H_k^V) and the virtual display's state (D_i^V).

Every model maps the user's hands state to the robot's grippers' state and the user's eyes state to the robot's sensors state except the Piloting Model which only maps eyes to sensors. Theses mapping are shown in figure 2.2 as arrows marked with M, e.g. (M_es) means mapping between eyes and sensors. Depending on the model these mapping can be direct and indirect as well as directed and bi-directed.

The Piloting Model represents a simple mapping from the robot's cameras to a display, while the robot is controlled using hardware such as a keyboard or a joystick. This model is naturally hindered by its simplicity and does not provide enough capabilities for most tasks.

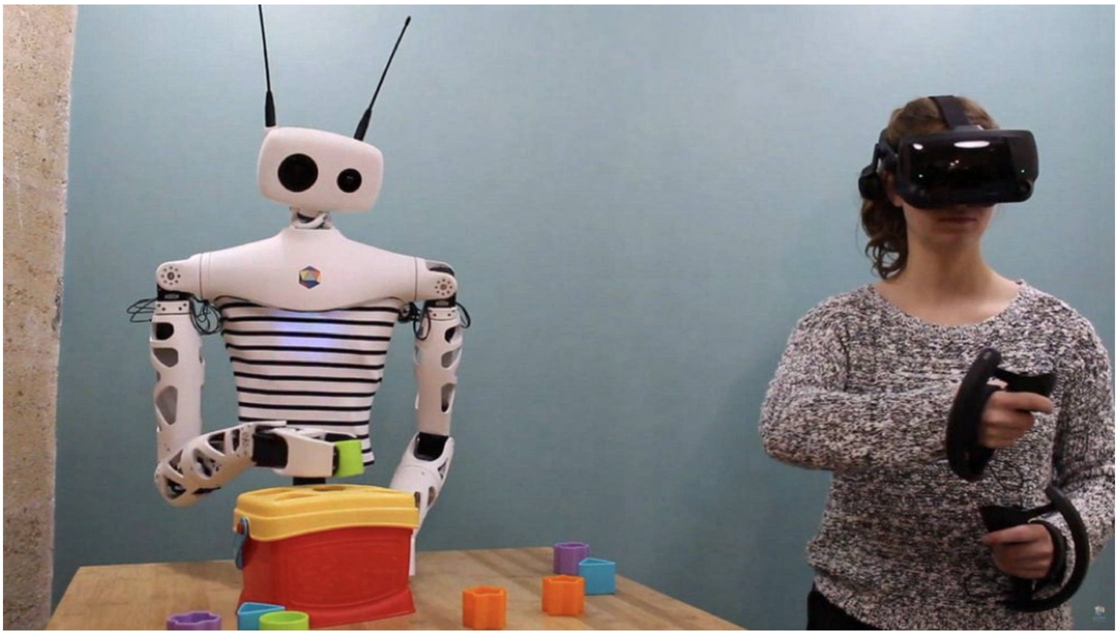

Figure 2.3: Reachy Mimicking Model [Pol].

The Mimicking Model represents a telepresence system where the user's state and the robot's state are directly linked, figure 2.3 shows an example. The user's head movement is tracked and directly applied to the robot's head, this either includes orientation and position (pose) or only orientation. This can anyhow lead to inconsistencies as the user has to either hold the head position as to not get disorientated or be careful to keep the head in a pose that is reachable by the robot. Most robots only feature two Degrees of Freedom (DoF) for the head while humans have six [Zha+18]. On top of this, as mentioned before, this mapping will always suffer from delay ($t + \triangle$). Zhang et al. [Zha+18] however, offered a solution: Instead of using the real video feed, they used the robot's color depth camera to render the 3D point cloud into a virtual environment (figure 2.4). In this virtual environment a virtual camera can be mapped to directly correspond to the operators head movements without delay.

Similar problems arise for the mapping of body movements. Possible human body movements are most likely not congruent to the abilities of the robot, so either some movements of the robots cannot be used or movements of the human body cannot be realised by the robot. Body movements for this model can be tracked by hardware such as VR controllers and force-feedback arms, while head movement is most easily tracked

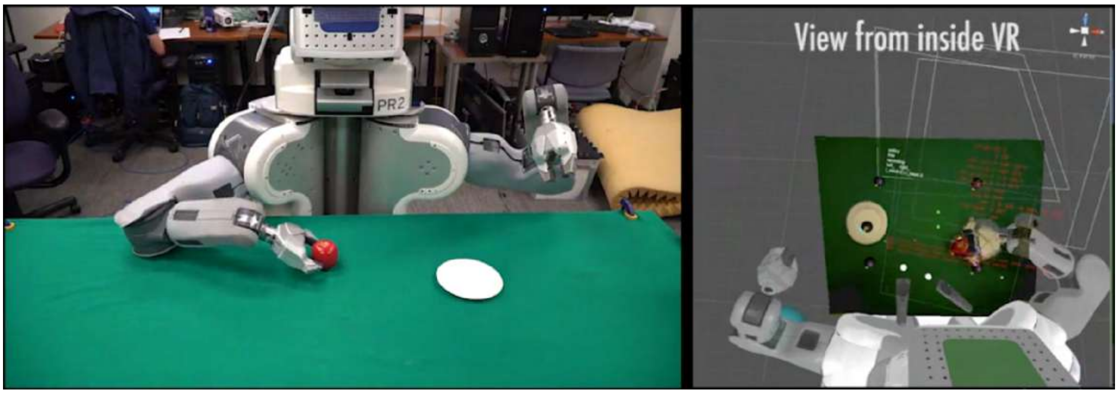

Figure 2.4: PR2 Mimicking Model [Zha+18]

by VR headsets.

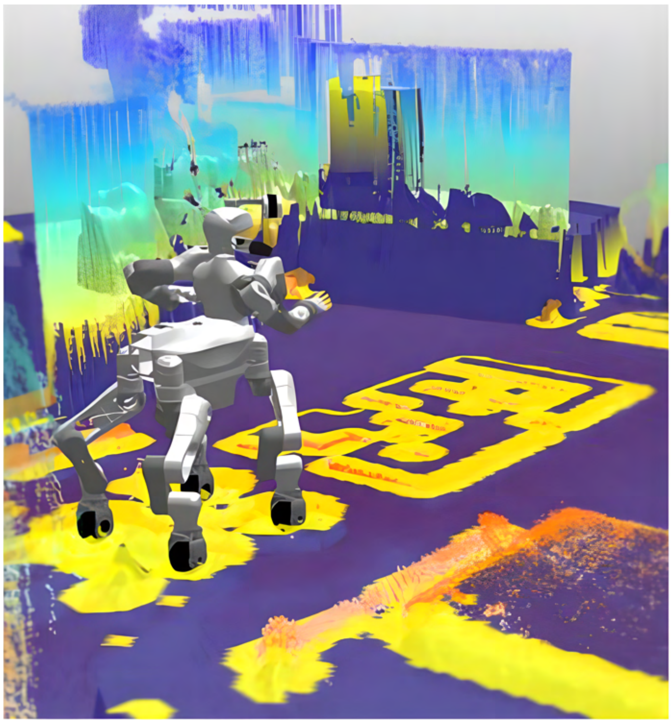
Figure 2.5: Centauro Cyber-Physical Model [Kla+20].

As hinted on by the presence of a virtual hands' state (H_k^V) a the virtual grippers' state (G_i^V) the Cyber-Physical Model uses a virtual representation of the robot and its environment as well as a virtual representation of the user. Both of these entities share the same virtual room where the state of the robot and its environment is constantly updated which then can be manipulated by the users through their representation. Figure 2.5 shows a implementation of a Cyber-Physical Model. The user is separate from the robot which allows the system to be more lenient on imprecise mappings. It is capable to be used for teaching or repetitive tasks while it is more difficult to react to changing tasks, especially if they require reactions. Furthermore, the constant updating of both robot state and environment state uses considerable amounts of bandwidth or a-priori knowledge of robot and environment.

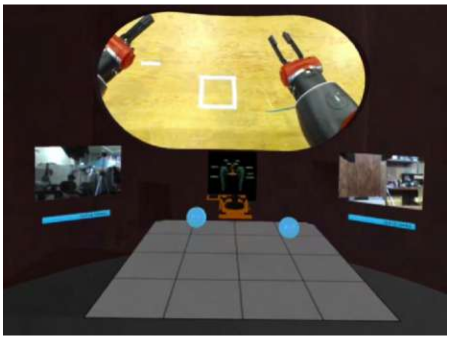

Figure 2.6: Baxter Homunculus Model [LFR18].

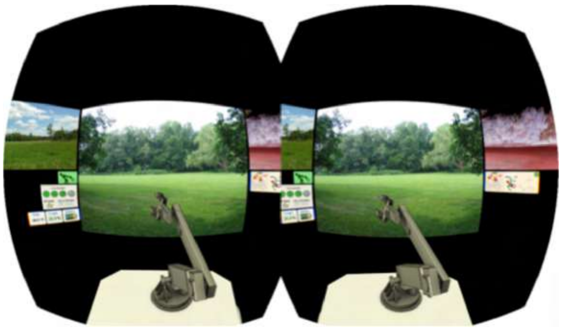

Figure 2.7: TAROS Homunculus Model [KN18].

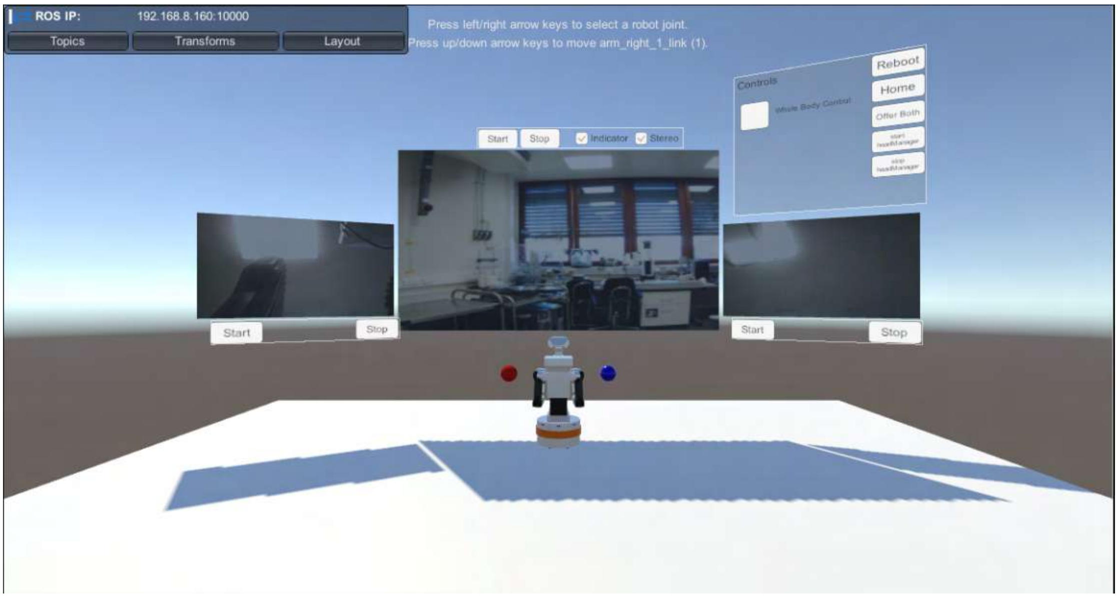

Figure 2.8: Tiago++ Homunculus Model.

The Homunculus Model introduces a Virtual Reality Control Room (VRCR) in which informations are visualized and controls for the robot are presented. Robot and user do not share the same virtual space, but the user is represented inside a virtual space from which the robot can be controlled and information about the robot is displayed. Figure 2.6 shows the implementation by Lipton et al. , while figure 2.7 shows another implementation that introduces a 3D representation of the robot that displays its current state. This 3D representation is not included in Lipton et al.'s description of the model, but can just be understood as an additional visual aid. Figure 2.8 shows one of the first functional iterations of the prototype developed in this work, which also includes a 3D representation of the Tiago++ robot.

In the Homunculus Model the user does not directly interact with the robot but chooses to engage with virtual tools that control the robot. Figure 2.6 and 2.8 show two spheres each that represent the current poses of the robot arms' end-effectors. These can be moved and rotated by the user to control the robot. So while selected the user's movements are coupled with the robots movements. This interaction can be implemented in different ways. What is important in principle is that the user actively chooses when and how their movements are coupled. So the possibility to implement multiple different tools in the same VRCR can also be explored.

What can be seen in figures 2.6, 2.7 and 2.8 is that all of them implement a central virtual screen that is capable of displaying a stereo image from the robots point of view to the user. Importantly, rather than having the stereo image be computed by combining the video streams of two separate cameras this computation can be shifted to the human brain by just showing one video stream to one eye respectively. This either lowers the amount of data to be streamed or lowers the computation load on the output VR device.

Unlike the Cyber-Physical Model, which tries to duplicate all of the sensory data provided by the robot, the Homunculus Model provides the opportunity to choose how to layout different sensor data. The user can freely shift their attention to the different visualizations. The previously mentioned 3D representation of the robot is one of those visualizations, another one are the video streams of additional cameras on the robot as can be seen left and right from the main screen in figures 2.6, 2.7 and 2.8.

2.2.4 Input/output devices

As previously established, a teleoperation system must be able to transfer sensory input to the operator and motoric input from the operator to the system. The amount of devices available for these tasks is vast and the choice which to chose has to be evaluated when designing such systems. When direct motoric movement is wished to be duplicated, Force Feedback arms are the most studied devices for bilateral control, since they are able to reproduce exact movement while being able to give motoric feedback as well [SJS99; HS06]. Other hand or body tracking devices using IR sensor or cameras, without a mechanical linkage, are easier to use but lack motoric feedback. Classical computer input devices such as keyboard and mouse or game controllers are unable to replicate motions but have lower hurdles of entry because of their widespread use.

Virtual and augmented reality are inherently linked to teleoperation and are a major component in teleoperation, especially since their widespread availability. VR is a cost effective way to stimulate human senses, but also suffers from distortion and other undesirable effects such as motion sickness. They are especially relevant because of their ability to transfer depth perception, build in hand-tracking and head movement tracking [Che10]. Multiple non-depth cameras can be conveyed onto many devices with 2D screens and are therefore worth exploring [AG10]. Their ability for hand-tracking can be combined with with robotic hands that could replicate the entire movement of the human hand, which is not explored in this thesis for lack of the specific hardware. Hand tracking can also be enabled through specific force-feedback gloves, but these would add considerable costs as well as another hurdle of entry for users.

2.2.5 Mobile robots in labs

A study by Thurow K, Zhang L, Liu H et al.[Thu+19] already reports a success rate 91% in lab ware transportation using autonomous navigation, collision avoidance across multiple floors and automated object recognition. This study was conducted at the Center for Life Science Automation (CELISCA) at the University Rostock [Cen] using H20 mobile robots.

Another mobile manipulator is being developed by the Fraunhofer Institute for Manufacturing Engineering and Automation IPA. This mobile lab robot is a combination of a already established mobile base being extended by an robotic arm. The robot can be controlled through scheduling software and autonomously grab and place labware [TB19].

Lab automation company Biosero is researching a similar combination of base and arm. The mobile manipulator called *Yoda* is able to be controlled by Bioseros scheduler software, though not much information is in the public domain [Lim21].

An article by Burger B, Maffettone P, Gusev V et al. [Bur+20] demonstrates how a mobile manipulator was able to autonomously perform experiments usually done manually by a researcher using the same devices and instruments.

In general, mobile robots have not seen widespread adaptation and no works regarding VR teleoperated robots in life-science labs could be found by the author.

2.2.6 Related work

The mobile manipulators in laboratories listed above are not being teleoperated and teleoperation does not appear to be a primary concern, but rather traditional automation. Therefore, research regarding teleoperated mobile robots from other domains or research concerning solely teleoperation are being discussed in the following. Additionally, an example of a market-ready robot that can be acquired already is presented.

VR teleoperation of mobile robots

Most of the robotic research community has settled on using ROS as the software component for their work, which was released in 2009. The first generation of consumer grade VR headsets (*HTC Vive*, *Oculus Rift*) were starting to become available from 2013 on wards. Generally, teleoperation for ROS systems were using the included visualization package (*RViz*, 2015 [Kam+15]) and the *Interactive Manipulation stack*.

Resulting in a point and click interface on a traditional screen. Multiple VR teleoperation packages([ROSb; HG] were released on the basis of a *RViz* VR plugin developed by Willow Garage [Ser11]. In the beginning the *Oculus Rift* did not include any hand or head tracking abilities, so additional hardware had to be used. *RViz* is conceptualized to be used on the same network as the robot, so inherently introduces latency negatively impacting teleoperation tasks, especially when large amounts of data are being transmitted [Whi+20].

Following these early steps, most research can be tracked back to Whitney et al. who researched a topic inline with this thesis trying to develop a framework to enable VR teleoperation of ROS robots which culminated in the *ROS Reality* framework [Whi+20] using *ROS#* the predecessor of URH. Their first paper [Whi+20] established the integration of the technologies used for this research through the *ROS Reality* framework.

	Aerial Robot	Mobile Robot	Manipulator	Mobile Robot w/ Manipulator	Dual-Arm Manipulator	Bipedal Humanoid
Visualization	2	2	4	2	1	1
Robot Control and Planning	2	0	5	0	4	1
Interaction	1	3	0	0	3	1
Usability	1	1	1	2	1	0

Figure 2.9: Heatmap of papers, VR and Robots [WP20], see Appendix C (pp. 101).

Wonsick and Padir published a review of scientific work concerned VR interfaces for controlling and interacting with robots in 2020 [WP20]. They identified 41 records of papers between 2016 and 2020 concerned with VR robot interfaces. They then categorized the papers into 4 categories: *Visualization*, *Robot Control and Planning*, *Interaction* and *Usability*. *Robot Control and Planning* is a category specifically concerned with connecting human input to robot movement, i.e. teleoperation. Figure 2.9 shows that they neither identified a single paper that focuses on teleoperating a mobile robot using a VR headset nor one concerned with a mobile robot with atleaste one manipulator attached. Thus the pool of similar work is correspondingly small. They did however identify 4 papers concerned with VR teleoperating Dual-Arm Manipulators. The *Usability* category encompasses papers that evaluate usability of VR interfaces for teleoperation, which is also related to this thesis. One of which will be presented in the following section 2.2.6.

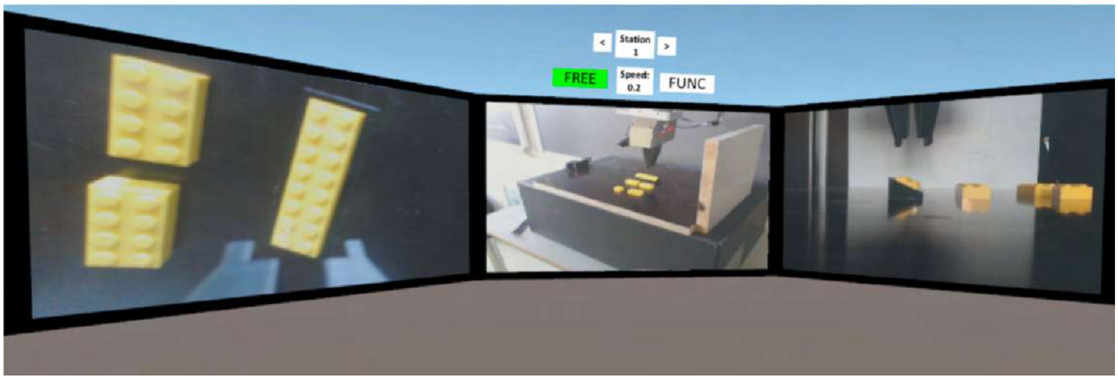

Figure 2.10: Bejczy et al. Homunculus Model [Bej+20].

Examples

Bejczy et al. [Bej+20] developed a prototype that is closest in its use case to the prototype developed in this work. A mobile base with a singular robotic arm was controlled using a VR headset to be teleoperated in a cleanroom. The system architecture used is similar to the ones mentioned above, using Unity and ROS, though the webcams used for the stereo virtual screen where mounted on a non mobile position instead of on the robot and another 2D webcam was installed in a fixed position. The only mobile camera was installed on the gripper. They report the ability of a trained operator to repeatedly suceed in high-precision assembly, namely the assembly of simple structures from LEGO bricks.

Figure 2.11: Maciaś et al. usability study [Mac+20].

Maciaś et al. [Mac+20] compared traditional teleoperation methods using multiple camera streams on a screen and controlling the robot with a gamepad, with a setup where a VR headset was displaying a stereo image while still using the gamepad for controlling the robot. They used a mobile robot with a single manipulator which was also controlled using the gamepad. Figure 2.11 shows the setup of the usability tests. They conclude that for manipulation tasks the stereo vision setup using the VR headset participants performed significantly better, while for driving tasks the traditional setup performed marginally better.

Kot and Novák [KN18] developed their Taros system based on a mobile robot using the homunculus model which can be seen in figure 2.7. Again, they did not take advantage of the tracking abilities of the VR headset, but used traditional control mechanism. They concluded that the use of the homunculus model reduced motion sickness considerably.

Zhang et al. [Zha+18] developed a VR teleoperation system controlling a mobile PR2 3.5 which can be seen in figure 2.4. As previosuly mentioned they used a virtual camera and a virtual environment rendered from a rgb-d camera. They used a homunculus control model. The aim of their study was to evaluate if human demonstration through teleoperation can be used to teach the robot to solve tasks autonomously. More on imitation learning in section 5.1.4. Their system is capable to repeatedly solve tasks such as grasping a tool, attaching wheels to a toy plane and picking up a piece of disheveled cloth through teleoperation to such a degree, that the collected data can be used for imitation learning.

Market-ready solutions

Figure 2.12: Reachy robot with mobile base [Pol].

In terms of market-ready solutions that enable a VR teleoperation that is comparable to the prototype developed for this thesis, only the Reachy robot by Pollen Robotics [Pol] could be identified. Reachy has similar capabilities to the prototype developed here and although previously developed as a stationary robot it has been upgraded with a mobile platform in 2022. It has compatibility with multiple headsets, although all of them are tethered and therefore only work with an attached computer. Information from their website suggests quite a functional teleoperation when in the same network, the remote functionality is reduced not using a stereo video transmission. It is in principle very similar since its using Unity in combination with ROS.

CHAPTER 3

Augmented Homunculus Model

Having described the theoretical basis and motivation for developing a VR teleoperation system, as well as having examined working examples, the following will accompany the development of the AHM. This control scheme was developed through the prototyping method introduced below. Furthermore, the reasoning for the chosen technologies is presented as well as a principle explanation on how to develop a system using AHM.

3.1 Prototyping method

To establish prototyping as a method it is necessary to define where this prototype lies in the field of prototyping. Following the definition of a prototype by Beaudouin-Lafon and Mackay, a prototype can be defined "as a concrete representation of part or all of an interactive system", further stating that "A prototype is a tangible artifact, not an abstract description that requires interpretation", [BM12]. They define Human-Computer Interaction (HCI) as a multidisciplinary field primarily occupied with design, combining elements of design with elements from science and engineering [BM12]. Prototyping offers a wide range of tools and strategies that have to be carefully matched to the aspired task. Generally, one can differentiate between four dimensions of prototypes, see Table 3.1. The *Representation* Dimension distinguishes between offline and online prototypes. Offline prototypes - offline meaning no use of software - are superior in early development stages, as they allow for rapid prototyping and are able to be produced by a wide array of people not only engineers. However, they struggle to enable real *interactivity*, which is a key aspect of user-centered design in HCI. Additionally, *precision* describes what the prototypes states as relevant details and what is left open as irrelevant details. The former is subject to evaluation the latter to design space exploration. *Evolution* is a critical dimension for this work, it describes the longevity of the prototype. Rapid prototypes are developed for a specific purpose and then discarded, whereas iterative prototypes evolve to increase *precision* or explore alternatives. Evolutionary prototypes eventually

Dimension	lower bound		upper bound
Representation	offline	-	online
Precision	rough	-	highly polished
Interactivity	non-interactive	-	interactive
Evolution	throw away	iterative	evolutionary

Table 3.1: Prototyping dimensions [BM12].

evolve into the final product and can be linked to the approach of extreme programming advocated by Beck [Bec99].

Typically, a rough paper prototype enables the best way of exploring the design space and can be easily discarded when it doesn't fulfill its expectations. Even though there are paper prototyping tools for VR applications [NM19], this prototype has to deal with robotics additionally which complicates the use of paper prototypes beyond designing the user interface. Also, for feasibility purposes, the user interfaces is not of the highest priority, which is why paper prototypes were passed over in favor of an established control scheme for VR teleoperation as an online software prototype (see section 2.2). In terms of precision, the prototype will be as precise as necessary to demonstrate its capabilities. Most importantly, the prototype will follow an iterative approach, where every iteration improves its predecessor. A full-on evolutionary prototype is out of scope for this thesis, as developing an expandable software architecture would take away too much time from the established goals.

The development paths and especially the undertaken changes of the user interface portrayed in section 3.3 are always a result of direct feedback provided by the researcher himself or colleagues that were mostly directly or indirectly involved with the project. Some were also separate from the project all together. The provided feedback was integral part of the decision making, however these feedbacks are difficult to depict and are subjective in nature, which is why they are not described in detail. Not every interaction can be meticulously described in a fluid prototyping process, but the conclusions made from interactions are portrayed to be comprehensible.

3.2 Choice of technologies

The space of possibilities to implement a system that enables teleoperation is substantial and software as hardware provide a realm of choices. This section will illustrate the reasoning for the choices made.

3.2.1 Input and output hardware

As discussed in section 2.2.4 there are potentially many input and output devices to choose from to build a teleoperation system. Table 3.2 shows a simple choice matrix to emphasize the choice of a VR headset. To elaborate on what has been mentioned in

Input device	Cost	Availability	1 to 1 Mapping	Ease-of-use
Force-Feedback Arm	high	no	yes	high
Gloves	high	no	yes	high
Keyboard/Mouse	low	yes	no	very high
Joystick/Gamepad	low	no	no	intermediate
Exoskeleton/Full-body	very high	no	yes	low
VR controllers/ hand tracking	low	no	yes	intermediate
Output device	**Cost**	**Availability**	**Depth**	**Ease-of-use**
Screen	low	yes	no	very high
3D screen	intermediate	no	yes	high
VR headset	low	no	yes	intermediate

Table 3.2: I/O hardware choice matrix.

chapter 1, the robot should be able to be controlled over the internet but not only that, it should also be able to be controlled by different operators from different locations. The columns *Cost* and *Availability* are therefore important criteria. If the hardware is already available at the location, no cost, no additional effort to provide the hardware and no overhead in troubleshooting the hardware is generated. In case it is not available, the cost are the next criteria to minimize. Exoskeletons or other elaborated configurations of hardware build around the human body are only feasible if the relation between operator and robot is one to one and the location is fixed. Providing these hardware setups to multiple locations is simply too impractical. The only hardware that comes into question is something that does not have high costs and takes up little space. From the listed input devices only Keyboard/Mouse, Joystick/Gamepad and VR controllers/hand tracking fulfill these requirements. VR controllers/hand tracking are the only devices offering a 1 to 1 *Mapping* meaning that the operator has access to three dimension to control the robot arms, so the operators movement can be mapped to the robots more or less directly.

Concerning the output device, a traditional screen has the advantage of already being at the workplace. It does not offer any depth perception which is highly useful in perceiving and manipulating the robots environment so the output device should be able to provide depth perception to the operator. Considering that VR controller and VR headset come as one product it makes sense to use both. When using a different input device, such as a camera or other hand tracking devices, a 3D screen is a valid alternative.

Having concluded that a VR headset fulfills all requirements needed for an input and output device the following will discuss which headset is most suitable. First of all, headsets can be differentiated between standalone and plug-in devices. The advantages and disadvantages of each category are quite self-explanatory. While plug-in devices are capable of using the connected computer's gpu and cpu they are also required to do so. They enable far more advanced real-time graphics compared to standalone devices that use the identical or similar chips to smartphones. However, attaching a VR headset to a computer is first of all cumbersome and second of all requires a computer that is probably more expensive than the headset itself. Since the teleoperation application to be developed is conceptualized to not need exceptional computing capabilities (section 2.2.3) the major downside of standalone headsets is therefore marginalized. A resulting downside is that due to lesser computing capabilities, the refresh rates and resolutions of the headsets' screens tend to be lower, although this cannot be said generally. Standalone headset are nevertheless capable of producing convincing immersive experiences and are far more approachable and usable to inexperienced users. They are also always using inside-out tracking which enables hand-tracking and can be interesting for future work, though not explored in this thesis.

Tables 3.4 and 3.3 combined show all standalone headsets released prior to 2022 which were actually available on the market at the end of 2021. Some devices offer only 3 DoF, meaning they are only capable of tracking orientation not position. As discussed in section 2.2.3, positional tracking is required if the user is allowed to freely move and not keep their head stationary. Since the user is in a VRCR positional tracking is necessary. All of the headset in table 3.3 are therefore ill-suited.

Scrutinizing the remaining headsets in table 3.4 reveals that only four headsets are considerable. The *HTC Vive Focus 3*, *Pico Neo 3 Pro*, *Pico Neo 3 Pro Eye* and the *Oculus Quest 2*. These headsets offer a refresh rate of 90hz or above and a resolution of 1832x1920 per eye or above. They also all include two 6 DoF controllers in their retail price, offer 6 or more GB of ram, have manually adjustable interpupillary distance (IPD) and feature the Qualcomm Snapdragon XR2 chipset.

The *Pico Neo 3 Pro Eye* and *Pico Neo 3 Pro* are both suitable devices, the eye model features eye-tracking, which is a feature not necessary of our prototype, but will be discussed in section 5.1.2. Generally, the Pico headsets are very similar to the *Oculus Quest 2* and are also capable of running applications developed in Unity. In this regard however the Quest 2 is superior, it has been available for longer, the user base is larger and the documentation is superior. Even if only slightly the Quest is also cheaper. If price is of no concern the *HTC Vice Focus* should be the device of choice, even though it offers the same chipset it features 8GB of ram and more elaborate active cooling and should therefore be more capable, while being significantly heavier than the Quest (785g vs 503g). Overall the $1000 premium is not worth the extra performance for our purposes. The only downfall of the Quest being its integration with facebook and the requirement of using a facebook account, which makes it unsuitable for many companies to use. In that case, the Neo 3 Pro should be selected. To sum it up, the *Oculus Quest 2* was chosen for the prototype for the reasons mentioned above, however all of the devices discussed in this paragraph are well suited.

Name	Refresh Rate	Retail Price	Display Type	Visible FoV	Resolution
		IPD Range	Chipset	Memory	Tracking Type
DPVR P1 Pro		$399	1xLCD	-	1280x1440
Light	90 Hz	-	Sd 821	3 GB	3 DoF
DPVR P1 Ultra		$599	1xLCD	90°d	1920x2160
4K 90 Hz	-	Sd 845	4 GB	3 DoF	
Lenovo Mirage		$450	1xLCD	101°h 101°v	1920x2160
VR S3	75 Hz	-	Sd 835	-	3 DoF
QWR VRone		$550	-	-	-
4K	72 Hz	-	Sd XR1	3 GB	3 DoF
DPVR P1 Pro		$349	1xLCD	100°d	1920x2160
4K	72 Hz	54-74 mm sw	Sd XR1	2 GB	3 DoF
Pico G2 4K		$399	1xLCD	101°d	1920x2160
	75 Hz	54-71 mm sw	Sd 835	4 GB	3 DoF
3Glasses X1		$550	2xLCD	105°h 88.6°v	1200x1200
	90 Hz	59-71 mm sw	Sd XR1	4 GB	3 DoF
DPVR P1		$199	1xLCD	100°d	1280x1440
	672 Hz	-	Allwiner VR9	2 GB	3 DoF
DPVR P1 Pro		$299	1xLCD	100°d	1280x1440
	672 Hz	-	Sd XR1	2 GB	3 DoF
Pico G2		$249	1xLCD	92°h 92°v	1440x1600
	690 Hz	54-71 mm sw	Sd 835	4 GB	3 DoF

Table 3.3: Standalone VR 3DoF headsets [Bro]. Sd = Qualcomm Snapdragon, hw/sw = hardware/software, FS-LCD = fast switch LCD, d/h/v = diagonal/horizontal/vertical.

Name	Refresh Rate	Retail Price	IPD Range	Display Type	Chipset	Visible FoV	Memory	Resolution	Tracking Type
Oculus Quest 2	120 Hz	$299	58-68 mm hw	1xFS-LCD	Sd XR2	97°h 93°v	6 GB	1832x1920	6 DoF
HTC Vive Focus 3	90 Hz	$1,300	57-72 mm hw	1xLCD	Sd XR2	116°h 96°v 113°d	8 GB	2448x2448	6 DoF
Pico Neo 3 Pro Eye	90 Hz	$699	58-69 mm hw	1yLCD	Sd XR2	98°h 90°v	6 GB	1832x1920	6 Dof
Pico Neo 3 Pro	90 Hz	$390	58-69 mm hw	1xLCD	Sd XR2	98°h 90°v	6 GB	1832x1920	6 DoF
HTC Vive Flow	75 Hz	$499	-	2xLCD	-	100° d	4 GB	1600x1600	6 DoF
Nolo Sonic	72 Hz	$470	-	-	Sd 845	101°h 90°v	6 GB	1920x2160	6 DoF
Nolo X1	-	$399	54-74 mm sw	-	Sd XR1	96°h 90°v	3 GB	1280x1440	6 DoF
XRSpace Manova	90 Hz	$499	-	-	Sd 845	100°h 90°v	6 GB	1440x1440	6 DoF
Pico Neo 2 Eye	75 Hz	$899	54-71 mm sw	1xLCD	Sd 845	101°h 101°v	6 GB	2048x2160	6 DoF
Pico Neo 2	75 Hz	$699	54-71 mm sw	1xLCD	Sdn 845	101°h 101°v	6 GB	2048x2160	6 DoF
HTC Vive Focus	75 Hz	$599	60.5-74 mm hw	2xAMOLED	Sd 835	-	4 GB	1440x1600	6 DoF
HTC Vive Focus Plus	75 Hz	$799	60.5-74 mm hw	2xAMOLED	Sd 835	-	4 GB	1440x1600	6 DoF
Sd 835 VR Dev Kit	60 Hz	$1,500	-	1xAMOLED	Sd 835	90°h 90°v	4 GB	1280x1440	6 DoF

Table 3.4: Standalone VR 6DoF headsets [Bro]. Sd = Qualcomm Snapdragon, hw/sw = hardware/software, FS-LCD = fast switch LCD, d/h/v = diagonal/horizontal/vertical.

Manufacturer	Name	Bi-manual	Torso lift	available
Pal Robotics	Tiago++	yes	yes	available
Pollen Robotics	Reachy with mobile base	yes	no	available
Rethink Robotics	Baxter with Mobility Base	yes	no	available
Willow Garage	PR2	yes	yes	discontinued
Kawada Industries	HiroNXO	yes	no	available
Aldebaran and Softbank Robotics	Pepper	yes	no	available
Omron Adept MobileRobots	Pioneer LX Manipulator	yes	no	discontinued
Pal Robotics	Tiago	no	yes	available
Fetch Robotics	Fetch	no	yes	available
Robotnik	RB-1	no	yes	available
Robotican	ARMadillo	no	yes	available
Neobotix	MMO-500	no	no	available
Neobotix	MMO-700	no	no	available
Carnegie Mellon University	LoCoBot	no	no	available
Iquotient Robotics	Caster Moma	no	no	available

Table 3.5: ROS enabled mobile manipulators.

3.2.2 Mobile robots

Contrary to input and output devices and especially VR headsets, mobile robots exist in a price range that restricts the freedom of choice for the researcher. In this case the mobile robot was pre-determined to be the Pal Robotics Tiago++ [PAL19]. The Tiago++ is definitely fitting and could have been chosen for the prototype developed in this thesis. For completeness sake the following will list and compare other suitable robots like the ones mentioned in section 2.2.6.

Without a guarantee for completeness, the ROS website [ROSa] offers a list of ROS enabled mobile manipulators, however this list is missing the Baxter as well as the Reachy robot mentioned in section 2.2.3 even though they both can be extended with a mobile base. It can therefore be assumed that other non-mobile manipulators that can be easily extended with mobility might be omitted from this list as well. Table 3.5 shows the robots identified in this manner.

The attributes *Bi-manual* (two arms) and *Torso lift* are of course not distinctive enough to create a complete overview of the robots. The possible criteria to differentiate between these robots would not fit in a simple table and 3.5 does not attempt to do so. This is partly because the characteristic *ROS-enabled* just describes that one can use ROS to develop for the robot, in what capacity different ROS packages are available to be used with the robot depends on each robot. Also some packages are specific and come pre-deployed they can therefore be expected to work relatively flawlessly or enable functionality that might be more difficult to achieve on other robots. This table does also not include the DoF of the arms or the quality and accuracy of the actuators installed. The table does however show two attributes that are essential to the use-case in this thesis which are enough to identify robots that are capable to be used in similar studies.

Figure 3.1: Fetch robot's torso lift [IEE].

Table 3.5 is ordered to show which robots are most capable for a teleoperation system in a life-science laboratory according to the attributes chosen. It can be argued that bi-manual is redundant and a single robotic arm is sufficient, but bi-manual teleoperation is more intuitive to the user and allows for more advanced use cases, where two arms might be needed to operate certain machines or equipment. The inclusion of the *Torso Lift* (figure 3.1) column can certainly be criticized to be artificial. However, what is relevant is the vertical reach of the robot, which in a laboratory environment has to be at least that of a human. Robots without a torso lift tend to be smaller because a taller robot would need a wider base in order to keep balance. A torso lift allows for a small footprint and high vertical reach.

Overall, the Tiago++ is the only robot to fulfill these criteria, this does not mean that it is the best or the only robot that should be used to develop a system like the one in this thesis. The Reachy robot does not incorporate a torso lift, but it still has considerable vertical reach and is built to be lightweight. Because of its pre-deployed VR teleoperation functionality it is most definitely more fitting to be used as a starting point. Because of its price point and widespread use in research the Baxter robot is also well suited. In the authors view these three robots are the most suitable for bi-manual teleoperation. If the necessity of bi-manual manipulation does not exist, any of the following can be chosen in the authors view: *Tiago*, *Fetch*, *RB-1* and *ARMadillo*. The *LoCoBot* has to mentioned for its low cost, while still encompassing the most essential characteristics necessary.

3.2.3 Software

As mentioned in chapter 1 the most convenient way to develop a VR application is to use a video game engine. Even though there are many different engines, in truth the choice lies between the two most popular engines, the Unity engine and the Unreal engine. Unity has been used to develop 60% of all content concerning VR and AR [TCP21], while the Unreal engine is of course similarly capable for the development of such applications. What was deemed most crucial is the integration of the engine with ROS, which is available for both engines based on the *rosbridge* ROS package. While the situation for the Unity engine was already discussed in section 2.2.6 and Unity was finally chosen for this system, the Unreal engine offers similar capabilities through the *ROSIntegration* plugin [MB19], while most research regarding VR teleoperation is using Unity. This is because the aforementioned URH offers supposedly better integration with ROS than the *ROSIntegration* plugin.

Instead of extending a video game engine to work withROS, another approach would be to extend graphical simulation software to work with VR headsets. Klamt et al. used VEROSIM [Kla+20] to develop their teleoperation system, see figure 2.5. In order to enable this they had to integrate SteamVR and OpenVR with VEROSIM. This approach however does not offer the same possibilities for user interfaces and in general is more appropriate for use-cases where one specific operator has to operate one specific robot. The Centauro project used the VR headset as an output device for stereoscopic vision, whereas in our case it is the central hardware the user interacts with, which is closer to use cases usually solved using game engines.

Looking at the URH website [Unib], it is primarily concerned with robotics simulation, but it offers the capabilities for our purposes in principle. First of all, the *ROS-TCP-Endpoint* package allows sending ROS messages from the Unity application to the robot. Therefore the robot can be controlled from within the application. Secondly, the *ROS-TCP-Connector* package enables receiving ROS messages from the robot, so sensor data such as the camera stream can be received in the application. On top of that, *Visualizations* package allows the visualizations of different message types. So a video stream can, for example, easily be displayed in an application. Finally, *URDF-Importer* package enables Unified Robot Description Format (URDF) files to be imported into Unity as fully physics enabled 3D character (see figure 2.8).

However, for our purposes the way an URDF is imported into Unity is problematic. The *URDF-Importer* package uses the *Articulation Body* component [Uni21] which uses a separate Featherstone solver for physics simulation, compared to the regular *Joint + Rigidbody* components which use the same physics simulation as the rest of the Unity scene. Without going into detail for now, what is important is that no physics simulation is needed at all and that the position of an *Articulation Body* component cannot be directly specified from code, just a force can be applied and then the physics engine calculates the position. Converting from *Articulation Body* to regular *Joints* is no trivial task and since the software examples in URH always use the *Rigidbody* component, the issue is further complicated.

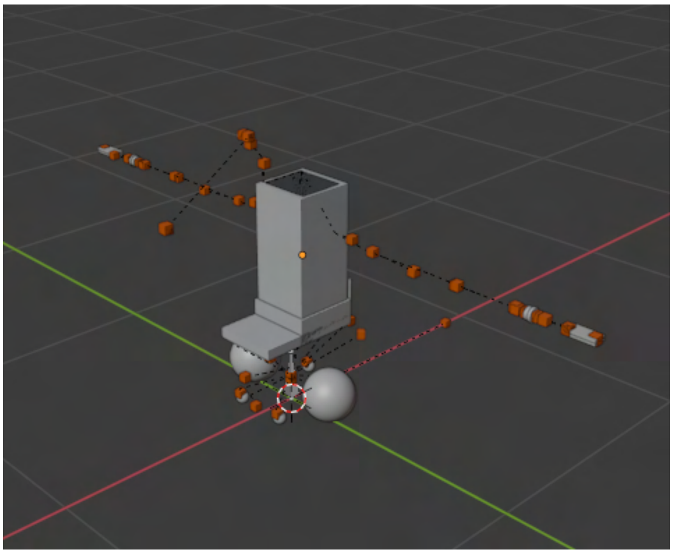

Figure 3.2: Phobos: Tiago++ URDF in Blender.

One potential solution is to import a 3D character from a 3D modelling program such as Blender, Maya or 3DS Max to emulate the usual workflow in video game development. *Phobos* [SR20] is a Blender plugin that is able to import URDF files and export them to a file in a format supported by Unity, which then can be used like any other character. Figure 3.2 shows the faulty result of importing the Tiago++ URDF file. Importing the Reachy URDF file also resulted in faults. While the results of using the *URDF-Importer* in Unity weren't flawless either, the author was at least able to fix them manually, which is why the prototype was implemented using the *URDF-Importer*. It has to be said that this decision eventually lead to problems such as Inverse Kinematics (IK) not being available for *Rigidbodies*. Generally, using *Phobos* or any method to generate a model in Unity that uses the *joint* component is advisable, however this was not clear to the author at the time of the decision.

3.3 Prototype iterations

Subsequently to the establishment of the elementary software and hardware components, all the necessary steps to create a functional prototype are described below.

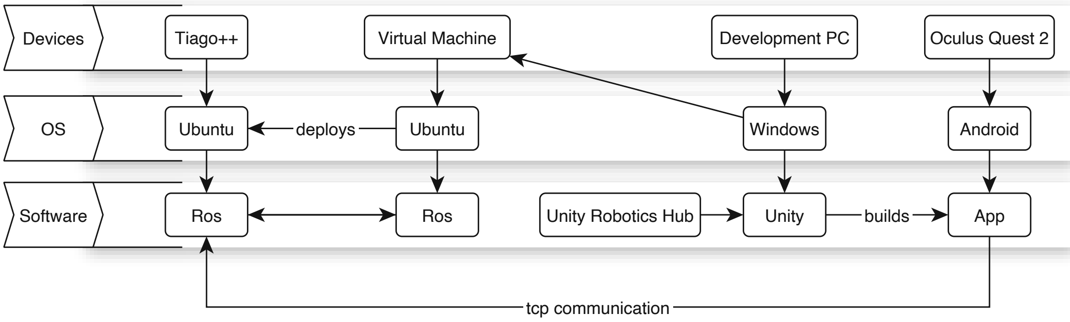

Figure 3.3: Software and Hardware Integration.

3.3.1 Software

The basic outline of which software is used has already been discussed, the baseline situation can be seen in figure 3.3. The development machine has to run Windows, as the Oculus Link software only runs on Windows, this may not apply to other VR headsets. Additionally a separate or virtual machine running Ubuntu is needed which has the same ROS Version as the robot installed. We are working with a custom ROS Melodic version specific to the Tiago++ robot, again this ROS versions is supplied by Pal Robotics and the situation may be different using another robot. The ROS installation is necessary to create and deploy packages to the robot and to test and troubleshoot, as ROS machine to ROS machine communication has far less points of errors than Unity on Windows to ROS or Android application to ROS. In order to enable VR capabilities in Unity different packages have to be installed in Unity depending on the headset. Similarly, the different URH packages have to be installed in Unity. These packages are not configured to be able to be built on Android, so they have to be manually configured in Unity to allow for Android builds. The Oculus Quest 2 development mode has to be enabled on the headset as well.

3.3.2 Camera visualization

After having completed and tested the fundamental hardware and software configuration the next step was to enable a depth image on the user interface. Since the Tiago++ does not have a stereo camera, the easiest way to achieve this is by using two webcams [LFR18]. Any webcam supporting Video4Linux (V4L) with an acceptable field of view (FOV) and resolution is suited, Foscam w41's were used in our case. Figure 3.5 shows the final configuration. Note that an unpowered hub is daisy chained between the powered hub and one of the gripper cams. Without this setup the gripper cam was unable to be recognized as a high-powered USB device and was therefore not usable.

The next problem in configuring the webcams stems from the fact that the Tiago robot which the Tiago++ robot is based on comes with a singular arm and therefore with a singular gripper cam. Since Tiago++ has two identical gripper cams they cannot

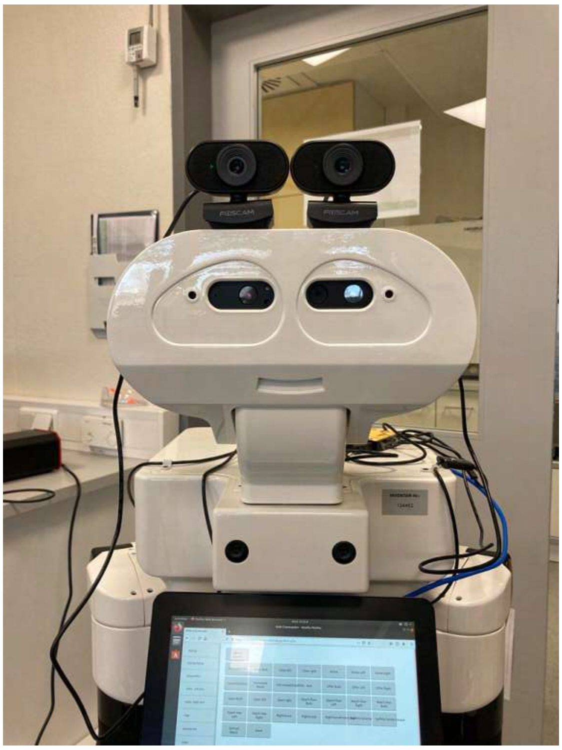

Figure 3.4: Tiago++ with webcams.

be differentiated (same product and vendor-id) through software. The script that was pre-installed differentiated between the gripper cam through order of connecting. Which was impractical, since on every startup the cameras had to be reconnected. Alternatively you can differentiate between two identical cameras through the USB port they are connected to. As long as each camera is always connected to the same port, the scripts developed could differentiate between all four cameras (webcams and gripper cams).

Once these problems were solved, there is two ROS packages that an be used for webcams. The *libuvc_camera* package was preinstalled on the robot, however *libuvc_camera* is problematic when using multiple identical cameras since it identifies them through the *serial number* which is often times empty and all other identification numbers between alike cameras are identical. Therefore, the *usb_cam* package was chosen. It identifies the device through the virtual device node, i.e. */dev/video0*, which can be identified from the USB port through a script. To sum it up: The cameras are always connected to the same USB ports, the startup script to start the usb_cam ROS node for each camera starts another script which identifies the virtual device node. Now on every startup up four ROS nodes are started, one for each camera.

These nodes can be subscribed to using the *ROS-TCP-Connector* package and visualized using the *Visualizations* package from within the Unity scene. Usually, VR scenes in

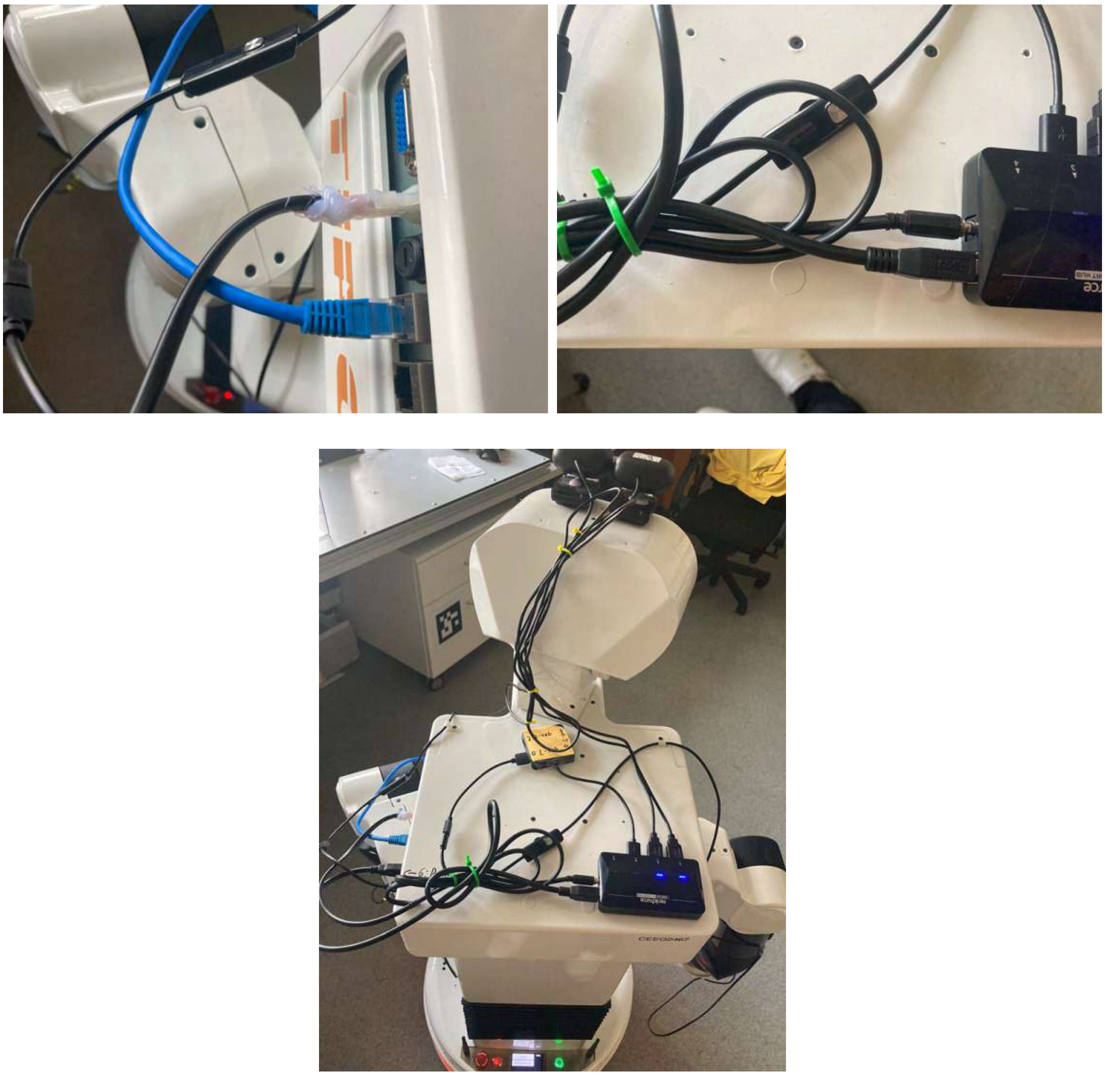

Figure 3.5: Tiago++ USB configuration.

Unity are configured to use one virtual camera, this can be changed to use two cameras, one for each eye. In this way the central screen can be configured to show the video stream of the right webcam for the right eye and the stream of the left webcam to left eye. The smaller screen left and right from the central screen are configured to show the left and right gripper cams.

```
roslaunch usb_cam
gripper_cam-test_left.launch
video:=$(v4l2-ctl --list-devices | grep -A 1 0-3 | grep video |
awk '{gsub(/^[ \t]+/,""); print $0}')
```

Listing 3.1: bash script starting left gripper ROS node

This setup worked in principle, the user was able to see all four camera streams simultaneously. The streams occupied to much bandwidth however, which resulted in the stream accumulating a significant delay over time. A similar problem was described in [LFR18]. This could be solved by lowering the resolution of each stream. Although working most of the time, sometimes the stream of the gripper cams broke down, which meant restarting the script responsible for starting the node. The source of the error appeared to be hardware related since it occurred only when the robot moved and could not be solved other then providing buttons to the user to manually restart the nodes.

Most importantly the user was able to perceive a stereo image, the depth perception however only worked perfectly at a certain distance, therefore another button was provided to switch between stereo streams and only one stream for the user to activate when working at a different distance.

3.3.3 Movement remote control

After the communication from robot to Unity was proven to be working, the next step was concerned with communication from Unity to the robot. In order to move the robot a scheme on how to control the movement had to be developed. Since the Baxter teleoperation system [LFR18] was non-mobile no inspiration could be drawn from there. The author decided to base this scheme on controller mapping known from video games, since that's what the Oculus controllers were developed for and since the application in principle moves a character in 3D space which is a common occurrence in video games.

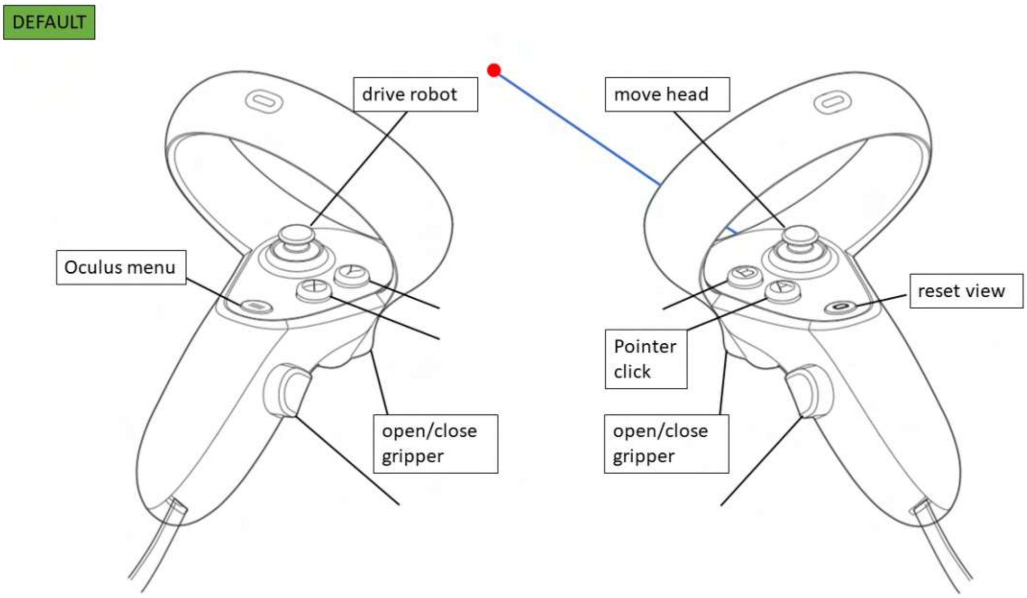

Figure 3.6: Controller mappings.

Figure 3.6 shows the controller mappings. The right controller doubles as a pointer device which enables the user to activate GUI buttons, such as the ones mentioned in the previous section 3.3.2, with the *A* button. Contrary to the controls used by the controller provided with the robot, which uses the left control stick to drive forwards and backwards and the right stick for turning, the mapping uses the left stick to control both driving and turning. The right stick is used to move the robot's head.

Using the *ROS-TCP-Endpoint* package ROS messages can be send to the robot. Tiago++ provides a topic that takes two 3D vectors (linear and angular), where only the first component of each is relevant. In order to move the robot you therefore need two values between -1 and 1 each (x-value = angular, y-value = linear). From tests conducted with the controller supplied by Pal Robotics, the ability to move straight without any turning was deemed necessary. Using two control sticks, one for angular velocity and one for linear velocity, this is easy to achieve. When using only one stick you have to introduce deadzones. Figure 3.6 shows the first iteration for the coordinate mapping of the left control stick. If the stick is anywhere between the two vertical lines (deadzone) the x-value is always 0. Once the stick exits the deadzone the x-value can be greater or lesser than 0 and the robot is enabled to drive curves.

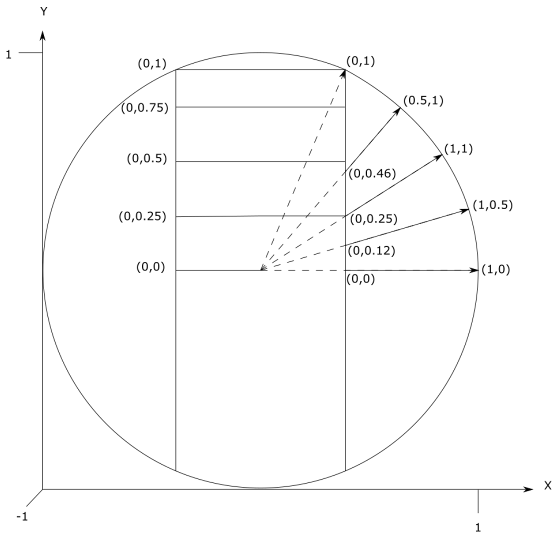

Figure 3.7: Coordinate mapping left control stick 1.

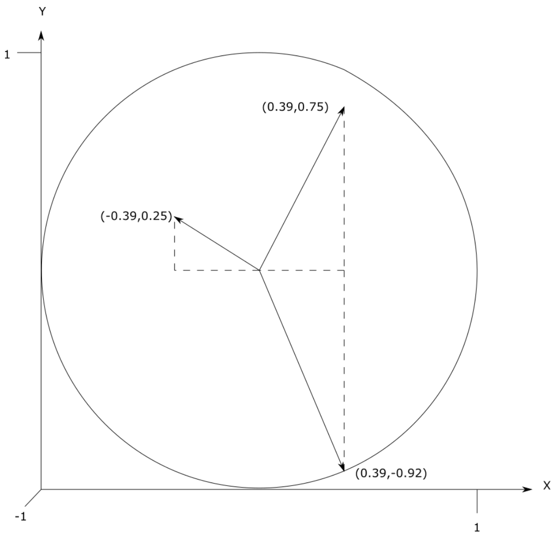

Figure 3.8: Coordinate mapping right control stick.

What is important to understand is that the robot cannot sidestep, it only has differential drive so it can turn in place. Which makes the chosen control stick mappings different to what is expected from some users familiar with video game control schemes where the character can move in all directions and the right stick not only controls where the character looks, but also defines where the forward direction is. In our case, the forwards direction is independent from the looking direction.

Figure 3.9 shows the mapping for the teleoperation package for the PR2 Robot. The PR2 robot has an omnidirectional drive therefore the right stick moves the robot in any

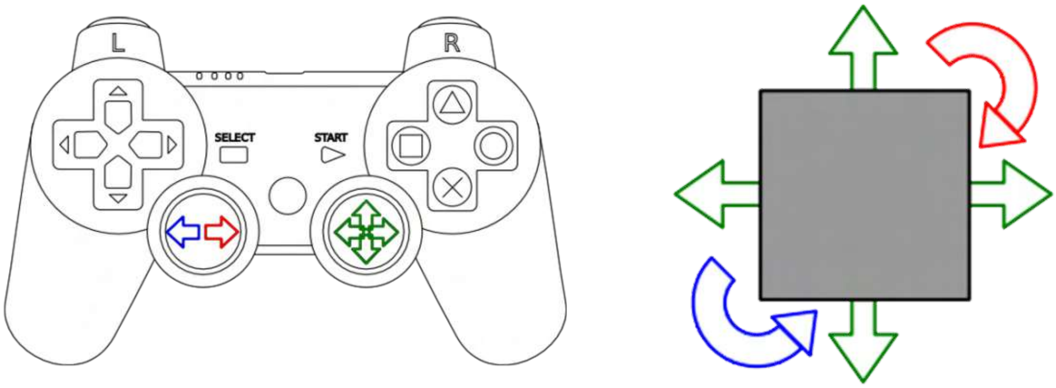

Figure 3.9: PR2 robot teleoperation joystick mapping [ROSb].

direction while the left stick rotates it. You can switch between base movement control or head movement control. The PR2's omnidirectional drive would allow for teleoperation very similar to character movement in video games. However, this was not achievable on the Tiago++ robot.

From tests it became obvious that it is not inherently obvious to the user that the looking direction of the head is not equal to the forward driving direction. Which is why pushing the right control stick down was programmed to recenter the head to look straight ahead. Tests showed that users were now aware of the fact that the head might not look in the driving direction when the function of centering the head was pointed out to them.

The stick mapping of the right stick was more straightforward otherwise. It was just mapped so that the components of the vector drawn by the control stick were send as up/down and left/right values directly, where the value corresponds to the movement speed, see figure 3.8. The head does not offer a topic to control its movement similar to the one for the robots locomotion, but rather uses its own *Head controller* to control its two joints. This is discussed in section 3.3.4.

Letting users navigate the robot in this way, they were now able to see the video streams, drive the robot and move the head and therefore the cameras looking direction, it became apparent that users were unable to finely navigate the robot. This is partly, because of the relatively small field of vision provided by the webcams as well as the robot being difficult to move in small increments. As it turned out, this was hardware related, as our particular robot had a heavier battery than standard, which meant that robot itself was heavier than originally designed. This resulted in the electric motors not having enough torque to move the robot up onto a certain threshold, e.g. when y was equal to 0.3 the robot was still stationary. Once over the threshold the robot was suddenly moving at considerable speed. As this behaviour was naturally not solvable through software alone, the coordinate mapping was still changed to allow the user to more precisely control the torque while giving up the ability to drive curves. Figure 3.10 shows the updated

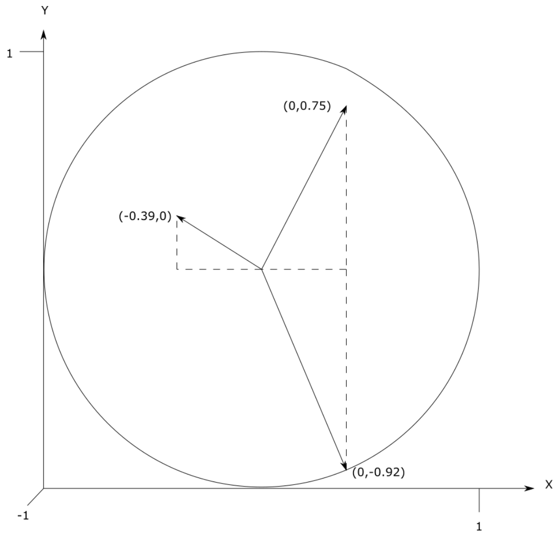

Figure 3.10: Coordinate mapping left control stick 2.

mapping for the left control stick. In this mapping only the component of the vector that has a higher absolute number, is considered, so the robot can either turn or drive.

Testing this mapping showed, that there does not seem to be a necessity to drive curves, even though it might improve the usability overall. More importantly, this implementation allowed the user to more precisely control the torque, so after a short acclimatization period every user was able to move the robot slowly and turn in small increments. For now, this solution was usable and users where now capable of moving the robot around the laboratory.

3.3.4 Body control

As alluded to in section 2.2.3 and seen in figures 2.5, 2.7 and 2.8, a 3D representation of the robot is strictly necessary in the Cyber-Physical Model while it can be included in the homunculus model. This representation will act as a mirror or digital twin to the real robot and will copy any joint movement, additionally as a first step towards controlling the robot, the robot will also mirror the movement of the representation.

Following the introduction to the URDF format in section 3.2.3, it was established that the URDF file will be imported into Unity using the *URDF-Importer* included in the URH. Thorough information about the URDF format can be found in Kang et al.'s paper [KKK19]. In general it describes all properties that are needed for a complete simulation, which means the URDF description contains more information than needed for our purposes. As previously mentioned the dual-arm Tiago++ robot is based on the single-arm Tiago robot, which in turn is also made out of separate components such as mobile base and left/right gripper. Before loading the URDF into Unity it has to be generated using the correct parameters describing the properties of the actual robot.

However, importing the URDF did not work without complications and they had to be manually resolved. As a comparison: The URDF of the Reachy robot was also imported, which worked without any complications. Nevertheless, once the URDF was imported and all simulation properties where disabled as much as possible since they were not needed, a simple controller was written that could control each joint separately (figure 3.11).

Figure 3.11: Joint controller.

Every joint has an integer value between a lower and upper bound which represents the current position of the virtual joint. This value can be read from the corresponding Articulation Body component. Sending the value to the corresponding *Torso-, Head-, Arm- and Gripper-Controller* as a *JointTrajectory Message* moves the joints of the robot to the same position as the virtual joint. Figure 3.12 shows how the virtual robot mirrors the state of the actual robot, while also allowing the joint controller to control both virtual and actual robot. Usually, the state listener sends the current joint positions of the actual robot to the joint controller to update the joint of the virtual robot. Only when the user is changing the state of the virtual robot, the state listener stops listening and the robot state is instead updated to match the virtual robots state.

This solution accomplishes two tasks. For once, it enables the user to see the current position of the robots joints, so they can see which direction the robot is looking and how far the arms are reaching out, even though they might be unable to see the arms through the webcams. The second task is to enable the user to control the joints one by one. This is done through a simple button layout where you can step through a list of all controllable joints by pressing a button. Another button steps in the other direction. The selected joint is highlighted in red as you can see in Figure 3.11. Two other buttons

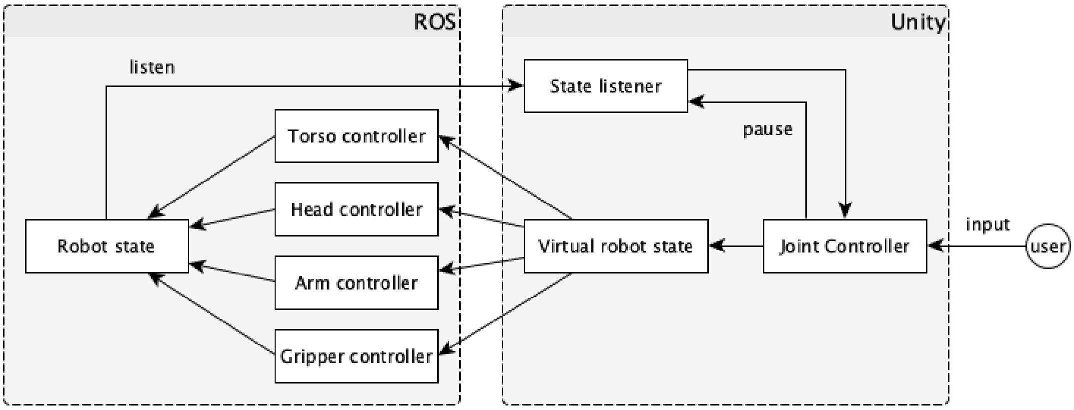

Figure 3.12: Robot state and joint control.

actuate/translate the joint through activating the virtual motor that controls it.

This solution is sub optimal in multiple ways. First of all controlling the joints one by one is cumbersome and trying to achieve any meaningful task like picking something up takes user such a high amount of time that it is basically unusable. It was however just meant as a proof-of-concept and does not represent the final mode of controlling the joints. The second complications stems from the fact that the *URDF-Importer* creates a virtual robot made for simulation, which already discussed in section 3.2.3. The Articulation Body component includes a virtual drive, so instead of just changing the position of a joint, the virtual drive is giving virtual torque which then simulates how the joint and the attached parts of the robot behave. At this point it is just unnecessary and inefficient but later on, in the upcoming sections, it will play a larger role considering the implementation of IK.

3.3.5 Inverse Kinematics

Up until this point the basic components for a teleoperated robot using a VR headset have been constructed. At this time it has been shown that all the software and hardware components can be made to work together in principle. The next step is to construct a way to control the robots arms using the pose tracking of the VR controllers. Using any software package that supports VR hardware, it is exceptionally easy to track the poses of the controllers in Unity. Using URH it is also possible to convert and transmit them to ROS. In section 3.3.4 it was shown that it is possible to mirror the state of the virtual robot to the state of the actual robot. In ROS terms this state is called a transform and it describes the coordinate frames positional and rotational relations to each other in a tree structure [Foo13]. In short it describes each joint's current pose. It is mapped to Unity through the URDF-Importer using the Articulations Body components.

This creates two principle ways to achieve teleoperation in the architecture shown in figure 3.3, which were coined *indirect* and *direct* and can be seen in figure 3.13. In the *direct* implementation the poses of the controllers is transmitted to the robot, where it is computed to change the robot's transform which then changes the virtual robot's transform. Using the *indirect* implementation, this computation takes place in Unity changes the virtual transform which then changes the transform of the actual robot.

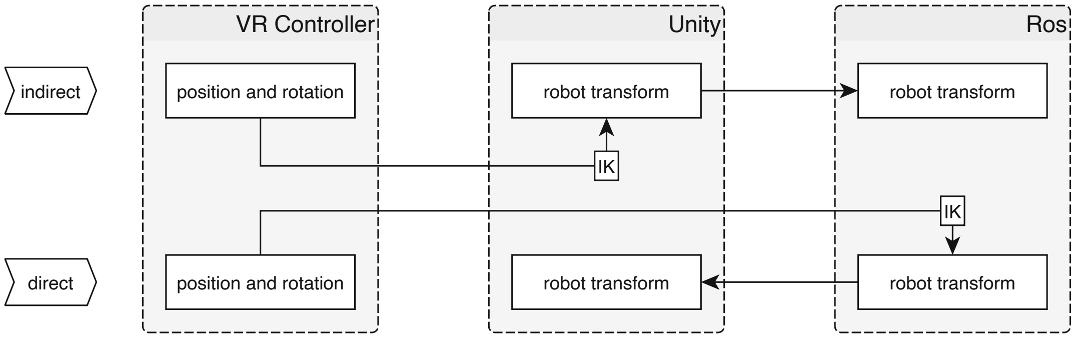

Figure 3.13: Inverse Kinematics software architecture.

The most straightforward way to implement this computation is using IK. IK allows the calculation of the joint angles needed to reach the end-effectors pose. It describes the opposite of Forward Kinematics which was shown in section 3.3.4. Using IK allows us to more or less directly transmit the pose of the controllers to the robot and the arms will reach an identical pose to the user, only the reference frame and the scaling have to be defined. This can be done in both implementations shown in figure 3.13.

To decide between these implementations availability is the most pressing concern: There are ROS packages that provide IK and Tiago++ comes with one called whole body control (WBC). On the other side, while IK is used in Unity since it is needed for character animation it is unavailable for Articulation Body components. Therefore, in our case, the *direct* implementation appears do be easier to achieve, since the WBC package just has to be integrated with Unity, no fundamental software has to be developed. If you look at other VR teleoperation solutions, you can see that the *direct* implementation is favored (Reachy, Baxter [Whi+20]). This can be tracked back to the fact that a 3D model representing the robot is not part of the Homunculus Model's definition (see section 2.2.3), but rather just a visual aid. Accordingly, the *direct* implementation was chosen for the prototype.

However, the *indirect* implementation can be argued to have many advantages over the *direct* implementation. While, as the naming suggests, *direct* minimizes the input latency to the robot, it also has a longer delay towards the visual feedback the operator receives both through camera streams and 3D model. *Indirect* also enables additional control modes, such as directly manipulating the 3D model in the VRCR. The operator could grab and move parts of the model and the robot would follow. This approach and the *indirect* implementation were briefly tested using both Reachy and Tiago++ URDF models.

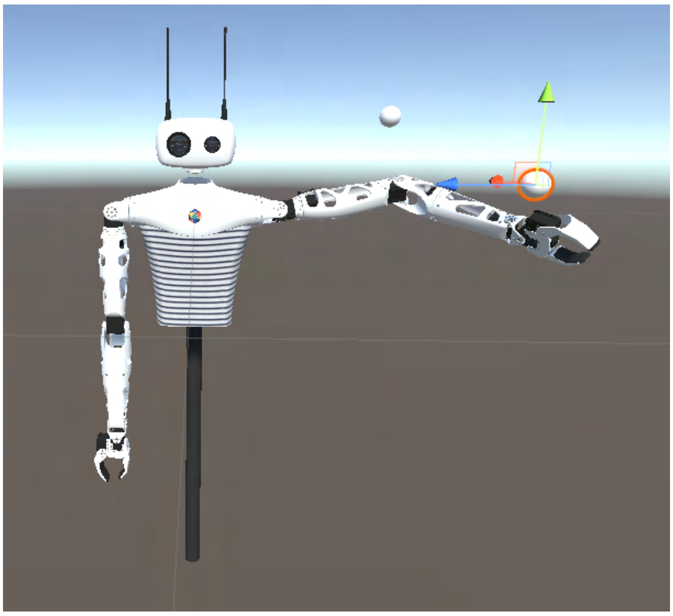

Figure 3.14: Reachy indirect control mode.

But in the end this could not be realized due to the development state of the Articulation Body component. There have been efforts of porting IK to the Articulation Body component, such as a working demonstration in post on the Unity Blog[1]. After contacting the developers it was stated that the software is not yet in a state to be shared. So it is not accessible and does not exist in a state where it is usable. There are multiple threads on the URH github stating the non-existence of IK for Articulation Bodies[2].

Since Unity Version 2022.1 in May 2022, Inverse Dynamics (ID) was implemented for Articulation Bodies, which solved most of these concerns. This will be discussed in section 5.1.1.

3.3.6 Whole Body Control

In order to enable the direct implementation, only the poses of the controllers had to be transmitted to the robot in principle, since the WBC package handles the IK. However the WBC package has certain quirks that made this task more inconvenient than expected. Once the WBC mode is active the Torso, Head and Arm controllers where deactivated, since these robot parts where now controlled by the WBC controller. This makes sense for the Arm and Torso since they move themselves according the the poses of the controllers, the Head controller however is a different situation. The head

[1] "Unity AI 2021 interns: Navigating challenges with robotics," Sept. 16, 2021. Accessed on: June 7, 2022. [Online]. Available: https://blog.unity.com/manufacturing/unity-ai-2021-interns-navigating-challenges-with-robotics

[2] "Unity-Robotics-Hub," Sept. 16, 2021. Accessed on: Jan. 25, 2021. [Online]. Available: https://github.com/Unity-Technologies/Unity-Robotics-Hub/issues/102

automatically moves out of the way when it would collide with one of the arms, which is why Head controller and WBC controller cannot be active simultaneously. The head can however be controlled through the WBC controller by transmitting a position for the head to look to, while previously the two head drives where directly controlled through the control stick axis (see section 3.3.3). Both of these control modes worked in principle, it was however confusing for the users that the feel of controlling the head changed during operation. Since general movement of the robot can be achieved without activating WBC, two separate control modes were conceived. The default control that works reliably but has no WBC function and the WBC mode that is only used when WBC is necessary. The modes can be changed through the checkbox seen in figure 2.8.

This was necessary since every time WBC control was turned on the arms moved into a default position, where the end-effectors where positioned above the head. If the arms where in a unsafe starting position, they would collide with the head and damage the grippers, because collision detection was not active during this time frame. On top of this, WBC had a tendency to detect false collisions and not detect true collisions. This resulted in the user not being able to control the arms and having to restart WBC, which brought the arms back to the starting position, losing their progress. If they were unlucky the restarting resulted in a self collision and the emergency power off had to activated to avoid further damage.

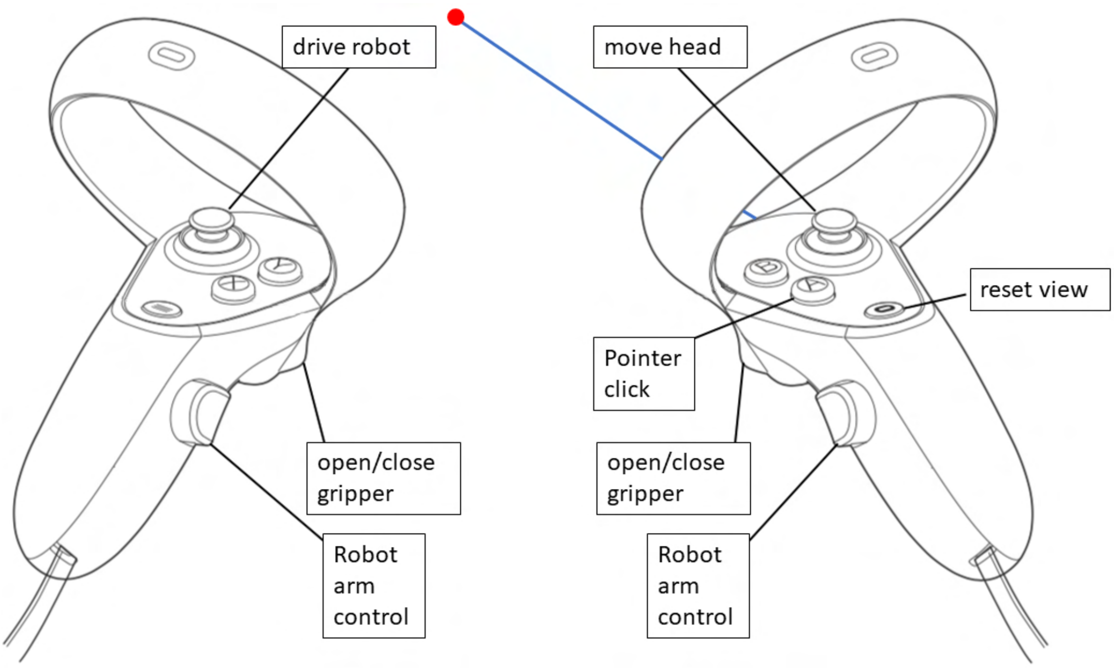

Figure 3.15: Controller mappings WBC.

The first iteration of the user interface to teleoperate the arms using WBC can be seen in figure 2.8, which was developed according to Lipton et al. [LFR18]. The two spheres could be grabbed by the user to be moved and rotated, while the end-effectors moved accordingly. Using spheres has the disadvantage of the operator not being able to identify the rotation. Ultimately this solution was disregarded for an interface more akin to the Mimicking Model.

Looking at figure 3.6, the unlabeled grip buttons where used to activate WBC as can be seen in figure 3.15. Once activated and while holding the buttons down the relative movement of the controllers were translated to the end-effectors. Whereas before, the virtual space was mapped to the robot space directly, now only the relative movement was transmitted. This had the advantage of the user still being able to choose when to engage with teleoperation, while being able to freely choose arm and hand positions, so the user could now also stay in the same position without having to adapt to the positions of the spheres. Also, a step wise teleoperation where the operator advances towards the task in steps is enabled in this way. E.g. one can press the grip button, move the controller forwards, release the button, return the controller to the original position, press the button, move the controller forwards again and repeat until the end-effector is in the desired position.

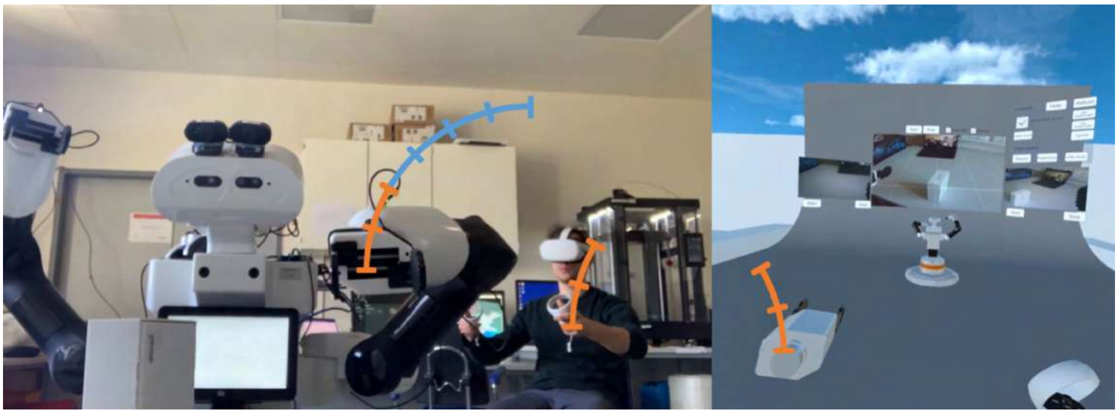

Figure 3.16: Relative teleoperation.

On top of this, it became obvious that relying on the visual feedback provided by the webcams and 3D model was not enough for the operators to be confident in their movements. To counteract this, the model of the controllers was changed to the model of the robot grippers only when engaging in WBC (see figure 3.16). This model also had the same orientation as the actual grippers. The orientation of the grippers became another point of confusion for users. First, the orientation of the controllers was mapped to the end-effectors absolutely, so when controller pointed forwards the grippers also pointed forwards. This was intuitive but it is often advantageous to grab something from above, which was barely possible with this solution and non-ergonomic in any case. As a result the rotation was made relative in accordance with the position. Now the operator could pre-rotate the grippers into the desired orientation and continue on holding the controller ergonomically. This took some users time of internalising, but was deemed an acceptable solution. For more details regarding this can be found in section 4.

3.3.7 Minor adjustments

As camera visualizations, movement control, body control and IK are functional at this stage, some minor adjustments are left to arrive at a stage where the prototype is able to be tested thoroughly. The Gripper controller is not deactivated during WBC, which means unlike the Head controller it can be addressed through one solution at all times. It was decided to use the trigger buttons to control the gripper in a latching fashion. Accordingly, the first press of the button closes the gripper up to a position where a certain resistance is detected (object is gripped) and the second button press releases it. The other solution would have been to grab as long as the trigger button is pressed and open on the release of the button. The first solution was deemed to be better suited as it was less likely to drop something that was already grabbed in this way.

The ability to rotate the 3D model of the robot was added as some users found it easier to operate while the model was facing them. Additionally markers were added in front of the robot that indicated the driving or turning direction while the robot was moving as to have some visual indication of movement apart from the webcams. On top of this, lines where rendered on to the central screen that changed according to the pose of the head that indicated the middle and the width of the robot (see figure 3.18). This is a visual aid for moving the robot through small spaces.

As there were constant connectivity issues regarding all cameras, separate *Start* and *Stop* buttons were added where users could manually restart the cameras. Also a *Stereo* button was added that can disable one of the webcams in order to disable stereo video as some users found it unpleasant to look at, especially when out of focus (see section 3.3.2). To decrease the learning curve of new users a small guide was integrated into the application that explains how to use the application, which will be further discussed int he next section. This guide was additionally printed out for the user tests.

3.4 System overview

The end result of the prototype is an advancement of the Homunculus Model, which uses the concept of a VRCR to counteract known adverse effect of VR. However during development certain aspect of the Homunculus Model were transcended. Characteristics of the Cyber-Physical Model as well as the Piloting Model and Mimicking Model were integrated. Following a strict definition of the Homunculus Model, the 3D representation of the robot as well Piloting Model inspired way of controlling the arms cannot be part of the Homunculus Model: The VRCR should only emulate a physical control room. The presence of a 3D model is able to break this perception and might lead the user to think they share the same virtual space as the robot (Cyber-Physical Model). The same applies for the WBC control which could not be realised in a physical control room. These objects and functions are augmented into the VRCR, hence the naming AHM. Similarly the users interaction with the virtual buttons using a virtual beam would not be possible in a physical control room.

Another aspect to point out is that this implementation of AHM is a complete control scheme for teleoperating a bipedal mobile robot, meaning that it also incorporates locomotion which is realised through a traditional control scheme comparable to the Piloting Model. This introduces another layer of control, but it does not break the perception of the user since the physical controller are also virtually represented in the VRCR.

To sum this up, the AHM aims to exploit all possibilities enabled by a VRCR and the VR hardware without breaking the perception of the user being in a virtual room teleoperating the robot. The AHM incorporates three layers of input and output. The user can interact with the VR interface buttons, use the physical controls of the controllers as well as interpreting the 3D visualization as can be seen in figure 3.17.

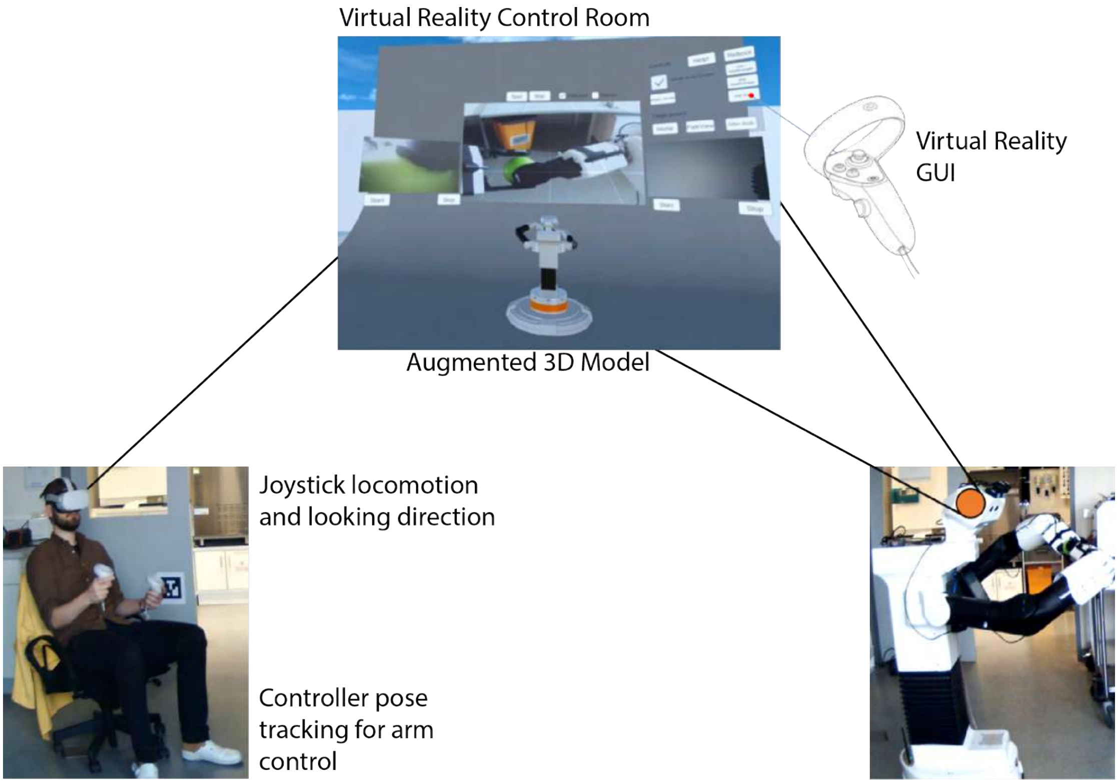

Figure 3.17: System overview.

A comprehensive overview over the system architecture can be found in the Appendix 5.2.

Incorporated into the software is the user guide seen in figures 3.18, 3.19 and 3.20. It can be accessed through one of the interface buttons and should enable any user to use the system with minimal external help. The user guide provides an overview of the graphical interface as well as the control mappings (page 1 and 2). The next pages (3 and 4) explain the WBC control mode as well as how to switch between modes and when to use them. The last pages (5 and 6) explain the concept of relative position and orientation and introduce the new control mappings of WBC.

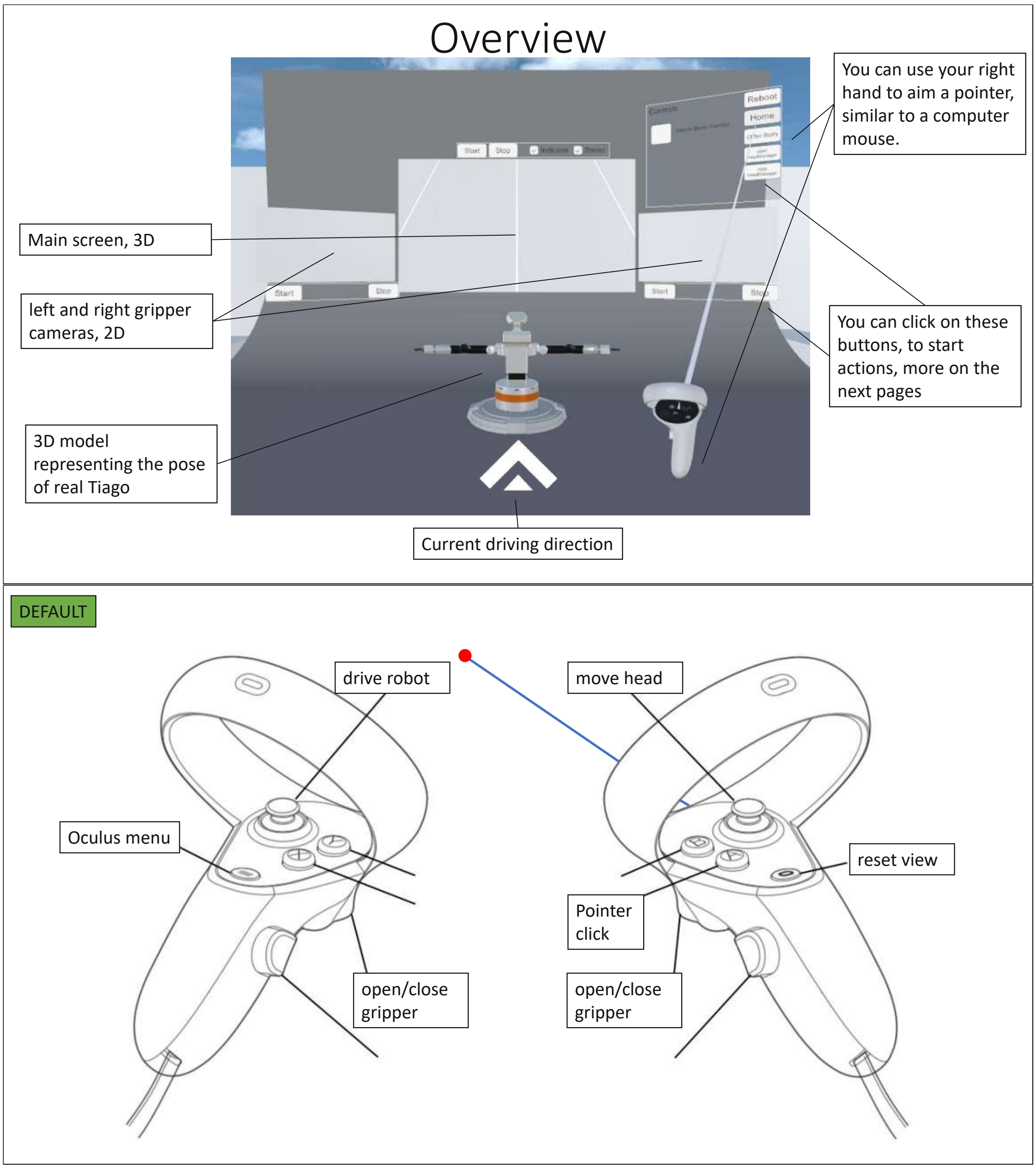

Figure 3.18: User guide, page 1 and 2.

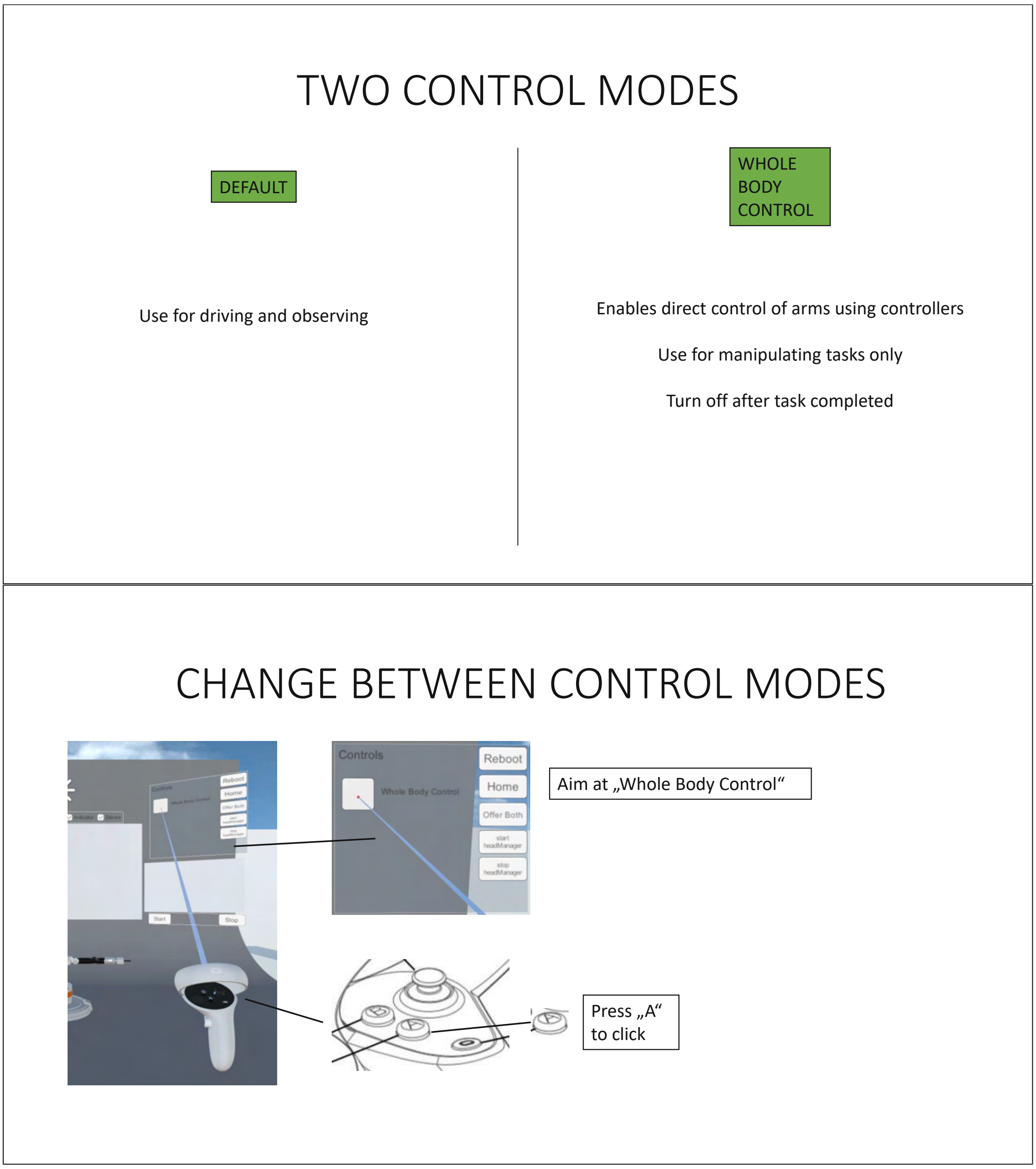

Figure 3.19: User guide, page 3 and 4.

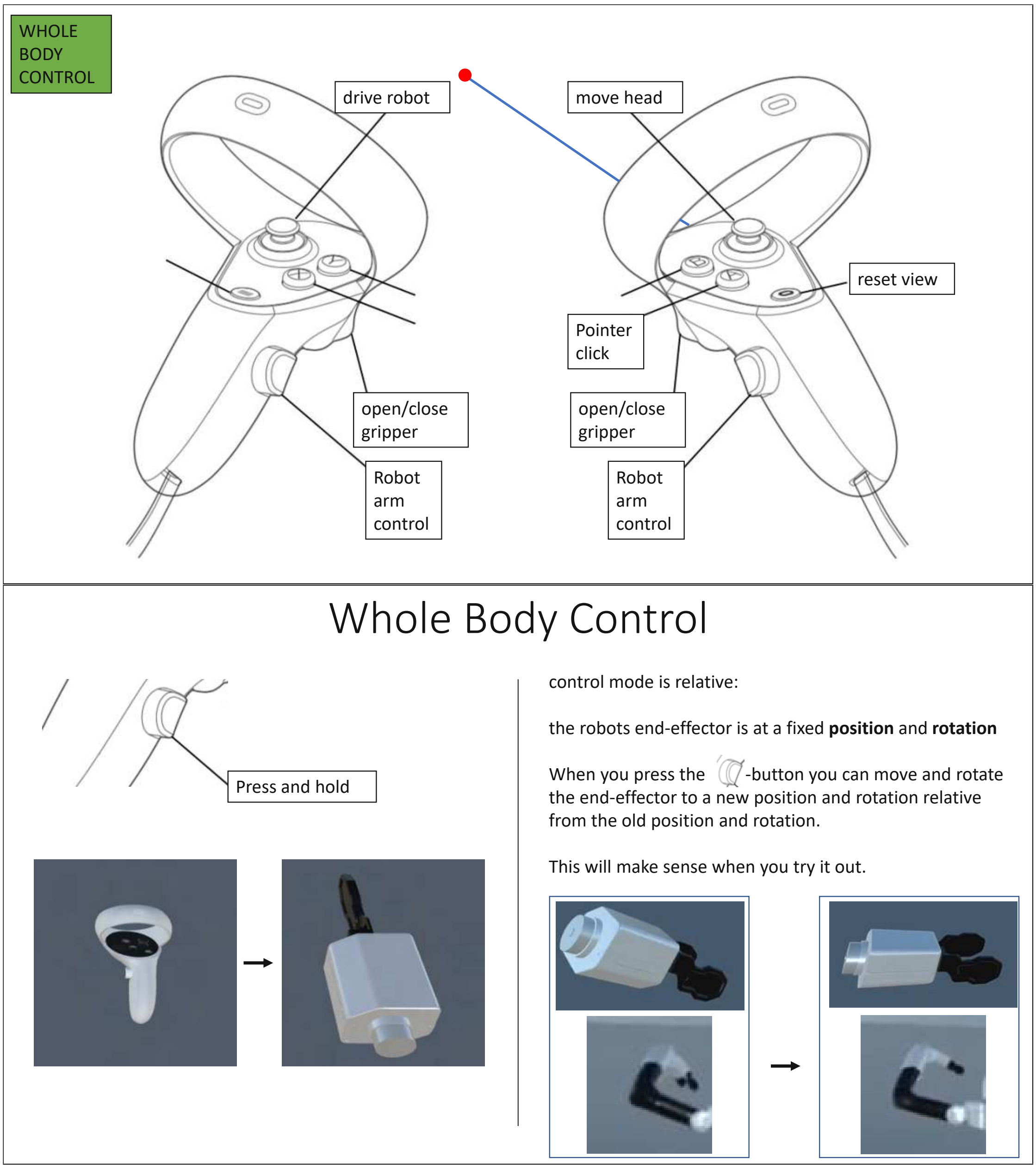

Figure 3.20: User guide, page 5 and 6.

CHAPTER 4

Experiments

Since the AHM up to this point is an unproven model, it will be evaluated through different methods explained below. Most importantly it will be subject to a user test with 13 participants. The tests are accompanied by observation (live and video), interviews and a questionnaire. The tests as well as all related data will be analyzed and assessed thereafter.

4.1 Methods

To counteract the ambiguity of the prototyping process, well defined quantifiable user tests were conceived to generate comparable results to other VR teleoperation systems. The tests as well as the supporting methods will be detailed in this section.

4.1.1 User tests

The participants are to fulfill five predefined tasks in a time span of 45min. These tasks are in ascending difficulty and introduce different capabilities of the system. The participants will also have to use the capabilities in tandem, e.g. drive and control arms simultaneously. The following tasks have to be performed:

- **Drive** Use the control sticks to move about in the room.
- **Door** Drive through an open door without hitting the frame, turn in a small space and reenter the room.
- **Observe** Drive towards a small thermostat display and move the gripper cam close enough to read the display.
- **Grab/Drop** Pickup a box and place it on a rectangular plate.

- **Handover** Pickup a ball and pass it from one hand to the other.

For benchmarking telemanipulation systems a vast amount of tasks have been conceived over time, such as cup stacking [Whi+20], unscrewing bottles and potato peeling [Whi+18]. However, the handover task is one present in most bi-manual teleoperated user tests. Also, some iteration of picking up an object and placing it in a predefined spot is also common. Since most teleoperation system do not use a mobile base, the *Drive* and *Door* tasks were conceived additionally to evaluate locomotion. One aspect not to underestimate is that much research is concerned with one expert operator trying to achieve the most difficult tasks, whereas these user tests are concerned with moderately difficult tasks for mostly inexperienced users in a small time frame. For this reason the time to complete a task is not measured, as most users never did any of the tasks before and would improve immensely when repeating the task. Instead it is considered how many of the tasks the users achieve in the given time frame.

4.1.2 Observation

In order to obtain additional auxiliary data the method of observation is used to capture the participants while conducting the research. Both outside view and the view from inside the VR headset are captured to allow further analysis. The reason for using observation is to uncover discrepancies between what the participants do and what they say [BB12]. Discrepancies might arise from the participants attempting to please the researcher or selective memory. The researcher takes an observer-participant role, as he both assists and observes during the tests instead of an outside observer [BB12].

4.1.3 Questionnaire

The questionnaire (see Appendix 5.2) is divided into different groups of questions.

The first few questions are designed to both evaluate if the participants had used the prototype before and if they have any experience using VR headsets specifically or other video game controllers in general. These question were designed to explore if previous experience in these domains is needed or if inexperienced users are also capable of using the system, which is what the researcher was trying to achieve.

Q1 Did you use any previous iteration of this prototype before?

Q2 Did you use this version of the prototype for the first time?

Q3 How much experience do you have playing video games with a game controller?

Q4 How much experience do you have using a virtual reality headset?

The following questions are designed to evaluate the likeability using several question on a Likert scale. Likeability was deemed very important by the researcher, as perceived ease of use might not reflect on the actual productivity of the system, but it is an important factor for the potential adoption of the developed system [Dav89]. Objective ease of use is relevant to the performance of the prototype whereas perceived ease of use is relevant for the users' decision to use the system at all [CT88].

Q5 How much did you enjoy using the system?

Q6 How much would you enjoy using this system routinely in your workday?

These following questions, as well the previous group of questions are partly adapted questions from the system usability scale [Bro96]. This group of questions is concerned with usability. It is specifically designed to see how pleased the participants were with the interface design and general control schemes chosen.

Q7 How much did you like the overall user interface? How intuitive did it feel to you?

Q8 How confident did you feel while driving the robot around?

Q9 How confident did you feel controlling the arms?

Q10 Do you feel like your were hindered more by your lack of experience (1) or by the system's ability to do what you command it to do (5) ?

Q11 Do you think using a virtual reality headset is necessary or would you rather use a traditional screen?

The following questions deal with the negative side-effects of prolonged use of the system. These questions are designed to evaluate the measures taken to counteract known side effect of VR headsets which was done in regard to the second research question.

Q12 While using the system, did you have any feeling of disorientation (1) or did you feel comfortable beeing in a virtual room (5) ?

Q13 Did you feel any adverse effect from using the system? Dizziness, nausea, headaches, ...

Q14 After the test, did you feel like you could have continued using the system without a break? Or were you glad you could stop wearing the headset?

The questionnaire concludes with open free text questions that capture possible future improvements to the system. These questions were designed with the third research question in mind.

Q15 Do you have any ideas on how to improve the system or would you change anything?

Q16 Anything else you want to mention?

4.1.4 Unstructured interviews

Throughout the prototyping stage as well as after the user tests unstructured interviews were conducted to capture feedback that wasn't included in the final questionnaire. The interviews gave directions for the iterative prototype development as well as influenced the holistic picture of the prototype in its environment. In addition to the ongoing tests during the prototyping stage, the following will cover the more in depth tests conducted at the end of this stage. The evaluation used 13 participants (9 male, 4 female) with ages ranging from 25 to 55. Of which 11 answered the questionnaire and all agreed to be videotaped.

4.2 Test results

This section presents the results of the user tests as well as analyzing the data generated. This will enable the construction of an overall picture of the performance capabilities of AHM in general and this system specifically.

4.2.1 Observation

Out of the five tasks described in section 4.1.1 almost all participants were able to complete tasks 1 to 3 without major problems. Figure 4.1 shows the different tasks and the success rates of the participants. Tasks *Drive* and *Door* were completed by everyone and only one participant was unable to complete task *Observe*. As expected, participants were less successful the more difficult the tasks became.

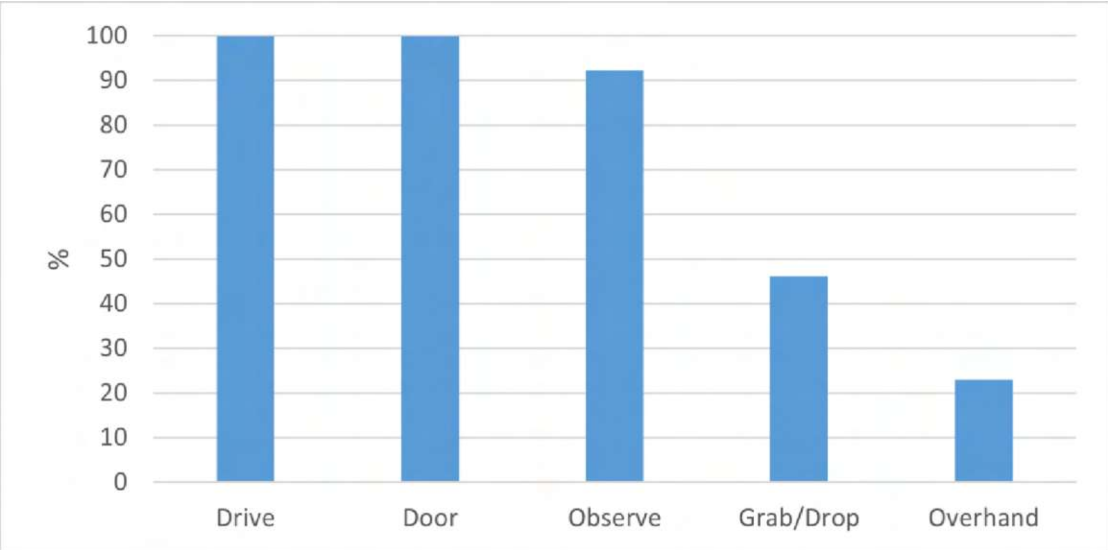

Figure 4.1: User tests tasks completion.

Interestingly, as can be seen in figure 4.2, no participant was able to achieve a more difficult task once they failed at an earlier one, e.g. if they failed at *Grab/Drop* they also failed at *Overhand*. Participants were given the chance to do so, but were also probably inclined to try to succeed at the task perceived as easier. User will be grouped into four categories *Expert*,*Proficient*,*Competent* and *Novice* according to their performance visualized in figure 4.2.

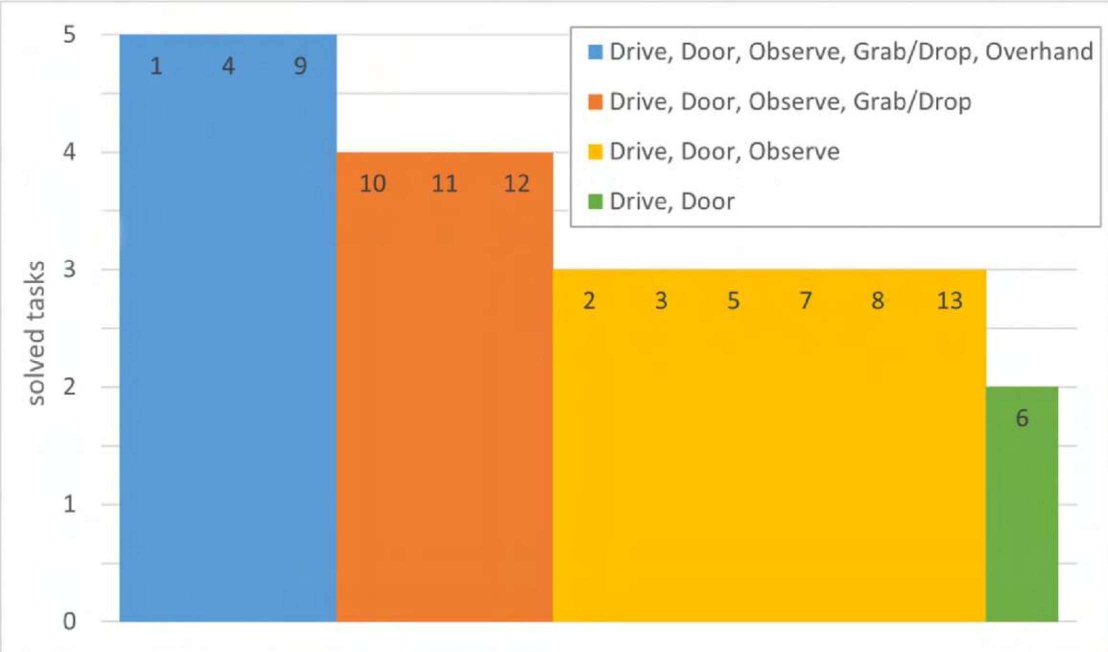

Figure 4.2: User tests tasks completion grouped.

There were no serious problems in navigating the robot, which was a concern during the prototyping stage, and even navigating through the door was less troublesome than expected by the researcher. Figure 4.3 shows the indicators on the central screen helping the participant perceiving the width of the robot. During interviews the indicator lines were perceived as very useful by the participants. Overall, the tasks concerning locomotion can be seen as very successful and the improvements made during the prototyping stage were effective. Even users without experience in video game control schemes were easily able to navigate the robot.

The *Observe* task was the first task were the participant needed to use WBC. It was conceptualized as an introduction were one arm needed to be moved and pointed to a relatively imprecise pose. Except one participant who had struggles to internalize the relative positional relation between VR controller and grippers, all participants were able to be successful. Figure 4.4 shows an example of a user completing task *observe*. This proves that the general way of controlling the arm was able to be understood by almost everyone, the following tasks are designed to test the fidelity of the WBC implementation.

The *Grab/Drop* task demands a higher precision from the user and is a test to see

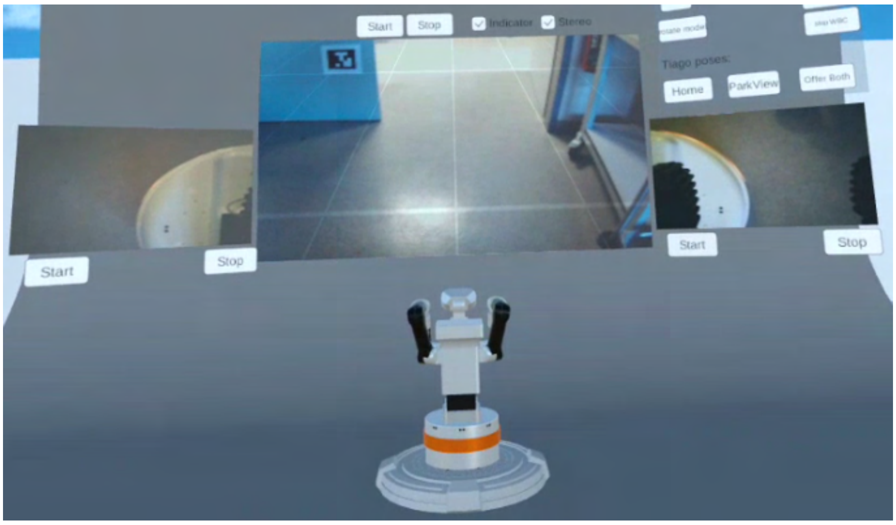

Figure 4.3: Task *Door*, participant using indicator lines.

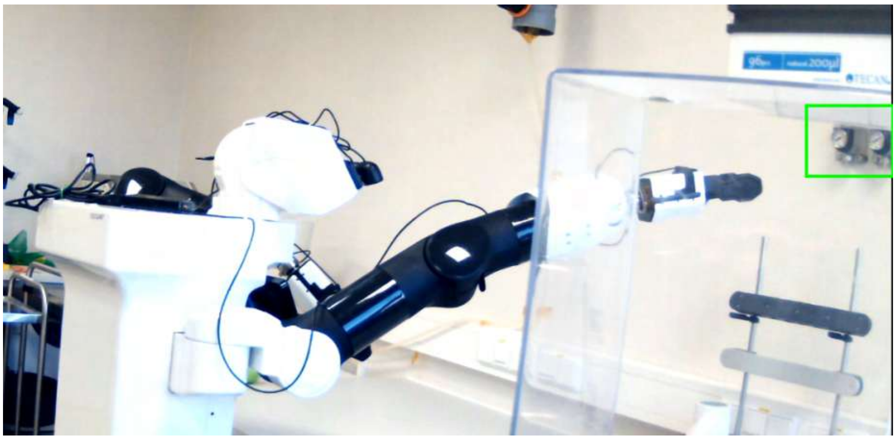

Figure 4.4: Task *Observe*, participant using WBC.

if WBC offers high enough fidelity for such tasks. Even though participants already had experience in driving the robot and getting to know the spacial dimensions, every participant struggled to drive close enough to the table for the arms to be in reach of the box. The reason being, that they perceived everything to be closer than it actually was. After reminding the participant that this is the case, most were able to adapt. When commanding the head to look as far down as possible, one cannot see the robots body, as can be seen in figure 4.3, which contributed to this warped perception. Overall the amount of environment that can be perceived simultaneously is limited, so most arm poses are outside the current field of view. Most particpants struggled to move the arms into a position where the main camera could see the grippers and if they could see them they often blocked a considerable amount of view. A strategy some participants adopted was positioning the gripper that would not grab the box so that they could observe the other gripper from another viewpoint (see figure 4.5).

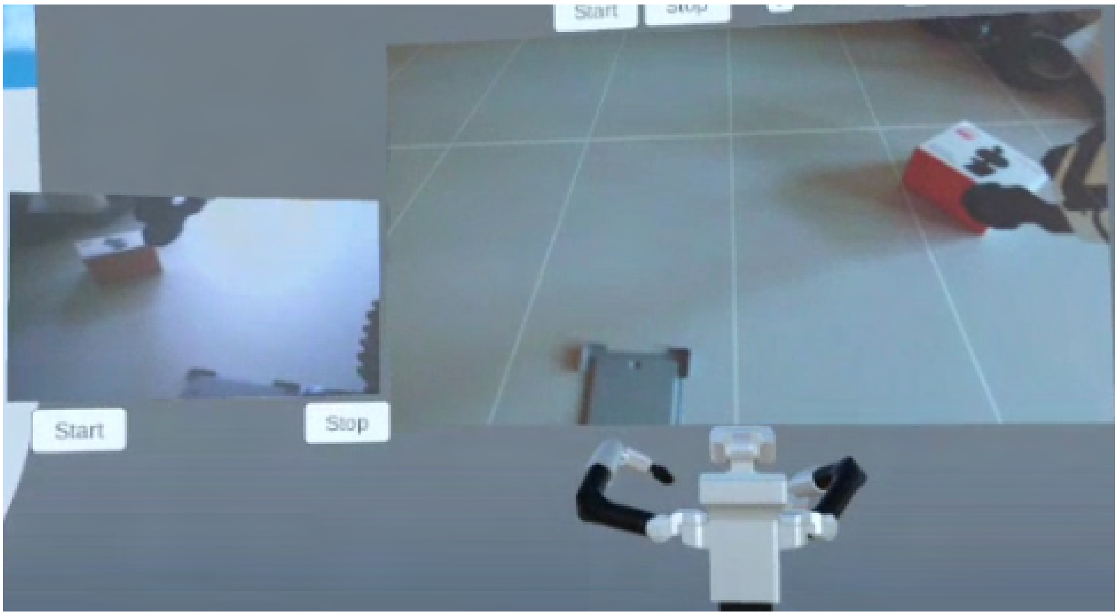

Figure 4.5: Participant using the left gripper cam as an additional view.

Participants were also troubled by limited availability of the gripper cam streams, most of the time they were able to get them back but some participants opted to just work around the limitation and work without one of the gripper cams, while still being successful (e.g. participants 4, 9 and 10).

Figure 4.6: Overhand test.

The *Overhand* was designed and proved to be the most difficult task. Failing gripper cams were less of a concern since this task is solved using the main cam as one can see in figure 4.6. What caused problems was one of the grippers constantly closing and not staying in a open position and therefore unable to grab the ball. This problem only occurred on the right gripper and was identified beforehand, but could not be fixed. After several iterations of opening and closing the problem sorted itself out for every participant. All participants that solved the task having previous experience in VR (Question 2 table 4.1), suggests that the overall precision of the system is high enough and the input lag low enough to enable tasks like this. Even though only 3 out of 13 participants solved the task, one has to be remember that very little to none of the participants had any previous experience.

4.2.2 Questionnaire

Based upon the expertise groups established above, the following will put the results of the tests in context with the answers given on the questionnaire which can be found in Appendix 5.2.

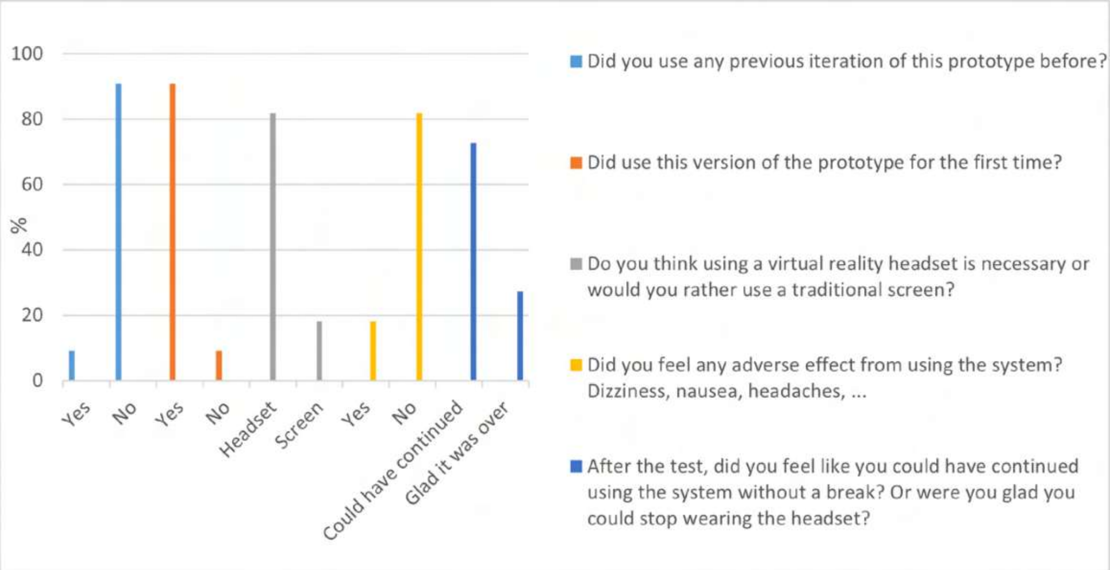

Figure 4.7: Binary questions, all participants.

11 out of 13 participants also took part in the questionnaire. Figure 4.7 shows the answers to all binary questions. Out of 11 participants only 1 had previous experience with the prototype. 7 participants answered identical and positive to all questions regarding usability (questions 3 to 5). One participant felt adverse effects, but not to a degree to answer *Glad it was over*. This participant did however not provide any feedback on why a *Screen* would be preferable.

Figure 4.8 shows the answers of those participants answering that they were glad the test was over. Out of the 3 participants that answered *Glad it was over*, 2 would prefer to use a *Screen*, but of those only 1 participant felt adverse effects. Therefore one participant would have liked to use a screen despite not feeling adverse effects. Those 3 participants

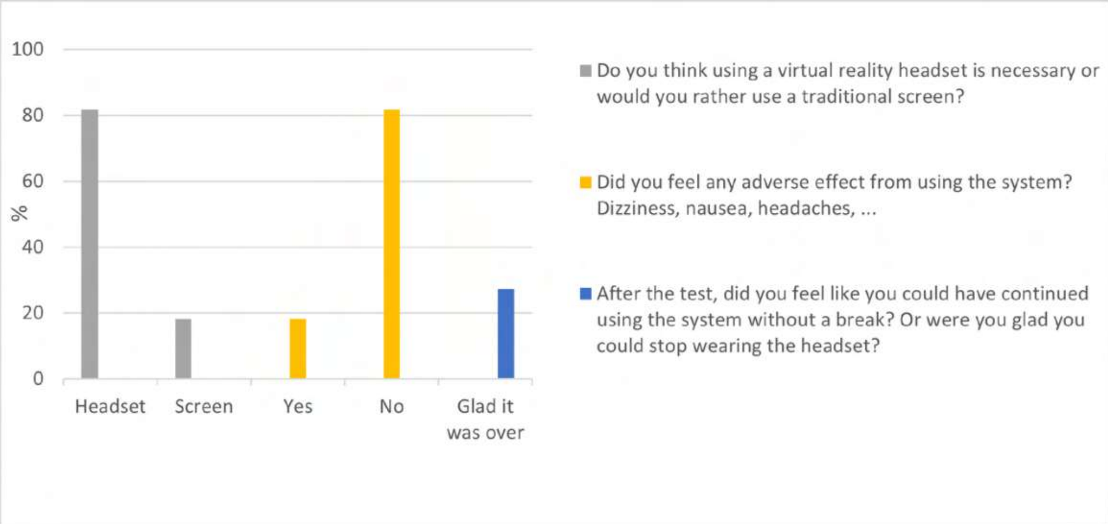

Figure 4.8: Binary questions, *Glad it was over* participants.

did belong to 3 different categories of expertise, but no one belonged to category *Expert*. Overall most did not feel any adverse effects and deemed the VR headset a necessity.

Figure 4.9, shows the median answers to the questions that were answered on a Likert scale across all 11 participants. All questions concerning likeability and usability have at least a median rating of 3. Confirming the implications from the previous question concerning adverse effects, participants felt confident being in the VRCR. Also, participants generally enjoyed using the system and liked the user interface. In contrast to how well the participants performed during the user-tests - all participants managed to solve the driving tasks while most struggled during WBC tasks -, they did not seem to differentiate greatly between driving the robot and controlling the arms. The answers to the question of enjoyment of using this system daily have the highest range. One participant would not enjoy using the system daily whatsoever while another one would enjoy it greatly. This is insight that some users could be intrinsically opposed to the technology.

Table 4.1 shows the median answers to the same questions grouped according the their expertise categories. What is most prominent at first glance, is that Expert participants have by far the most previous experience with VR, meaning that VR experience translates to capabilities in this VR teleoperation system. The prototype is therefore able to be used in the right hands and participants performing less well can be expected to improve. What is also noticeable that the median score of each expertise category correlates with their answers to questions concerning likeability and usability. Participants that performed better rated the the system higher than participants performing worse. This is in itself not surprising, but it has a confirming nature. If the previous hypothesis is true, that participants would improve with more experience, their rating of the prototype would therefore also improve.

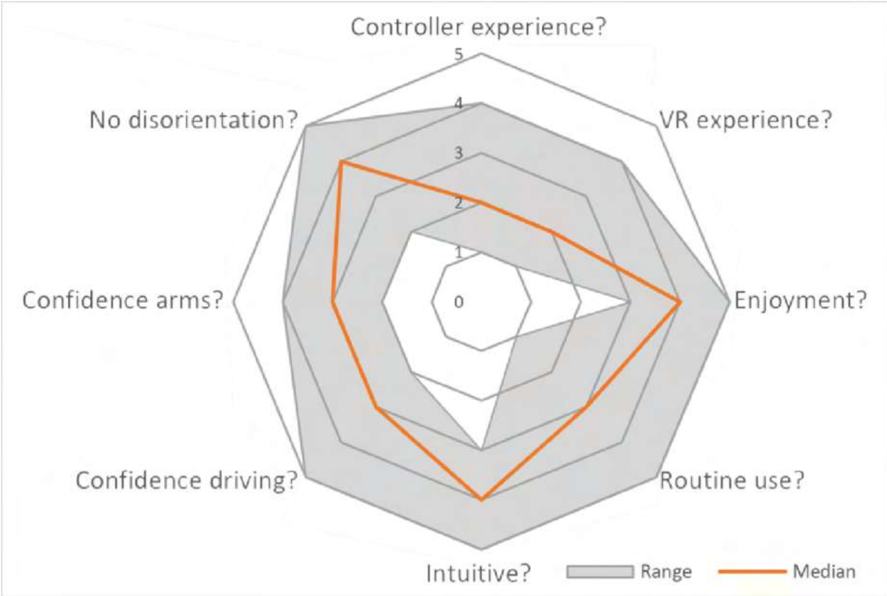

Figure 4.9: Likert scale questions.

Expertise	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q12
Expert	3	4	4	3	4	5	4	4
Proficient	2	2	3	3	3.5	3.5	2.5	2.5
Competent	2	2	4	3	3	3	3	3
Novice	4	2	3	2	3	2	2	2

Q3: How much experience do you have playing video games with a game controller?
Q4: How much experience do you have using a virtual reality headset?

Q5: How much did you enjoy using the system?
Q6 How much would you enjoy using this system routinely in your workday?
Q7: How much did you like the overall user interface? How intuitive did it feel to you?
Q8: How confident did you feel while driving the robot around?
Q9: How confident did you fell controlling the arms?
Q12: While using the system, did you have any feeling of disorientation (1) or did you feel comfortable beeing in a virtual room (5) ?

Table 4.1: Median answers of expertise categories.

Figure 4.10 shows the median answer to Questions 5 to 9 and 12 of each participant. This confirms the previous observation that participants performing better also rated the system higher. The relevance of the novice category is limited due to having only a singular participant.

The question seen in figure 4.11 was designed with anticipated hypothesis (that partici-

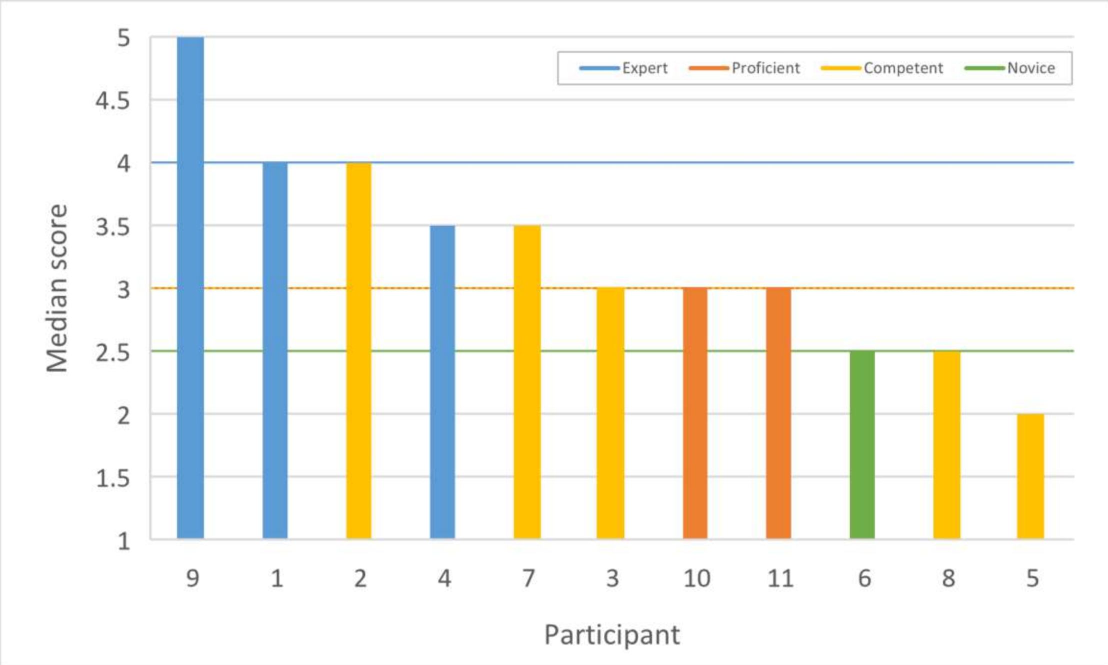

Figure 4.10: Likert scale questions per participant (Q5 - Q9, Q12).

pants performing better would also rate the system higher) in mind. In short it should answer if participants would attribute their success/failure to themselves or the system. The figure shows an interesting implication, participants performing well think they could have performed even better if the system would improve, while participants performing worse appear to blame their lack of experience. Those performing worse therefore assume that they would more easily improve by training, while those performing well think the way to improve would be to improve the system. Since expertise groups appear to correlate to the answers given, this further cements the grouping as well as the tasks as appropriate. Most importantly, those rating the system likeability and usability highly, still see improvements that can be made to the system.

4.2.3 Free text questions and interviews

Figure 4.12 shows the answers to the free text questions *Do you have any ideas on how to improve the system or would you change anything?* and *Anything else you want to mention?* categorized and visualized according to prevalence. Most commonly the participants complained about a lack of accuracy when controlling the arms. In the same vein, some participants provided ideas for additional control modes to counteract the inaccuracy: '*Controlling the whole arm and not just the end effector might be helpful in positioning the gripper*', '*Rotations and translations of the robot arm might feel more natural if they were relative to the gripper's local coordinate frame*'. Two participants asked for force feedback for collisions, while even more proposed that there should be a preview

Figure 4.11: Lack of experience vs. execution accuracy.

of where the arm is currently moving towards or complained about lag. One participant proposed to train inside the virtual environment before teleoperating the robot in order to get used to the system. So overall, more than half (54%) of the topics mentioned by the participants where directly connected to their problems of either controlling the arms or lacking feedback of what the arms will do.

The next biggest single category is concerned with the hardware. While no participant blamed the VR controllers for inaccurate tracking, multiple participants mentioned the headset being to heavy, one complained that it flickered: '*VR Glasses are very heavy and flicker*'. Two participants complained about the blur and proposed to use a different headset: '*Better VR headset: less blur, better depth perception*'. If they truly meant that the headset produced a blurred image or rather that the webcams used where focused on the wrong depth or generally were miss-aligned during their test is up for interpretation. However, one participant directly identified the cameras as the culprit and proposed to use bespoke depth camera hardware: '*Upgrade the Cameras for 3D View*'.

Furthermore, one participant proposed to move the interface closer to the user, which might stem from the participant not being aware that you can freely move closer in the virtual space. Unexpectedly, from the researchers perspective, only one participant complained about the reliability of the system, even though many tests had webcams that needed to be restarted and WBC breakdowns. During interviews the statements of the free text answers were mostly repeated. The participant that felt dizziness from the test reiterated their complains. The effects hindered them throughout the work day.

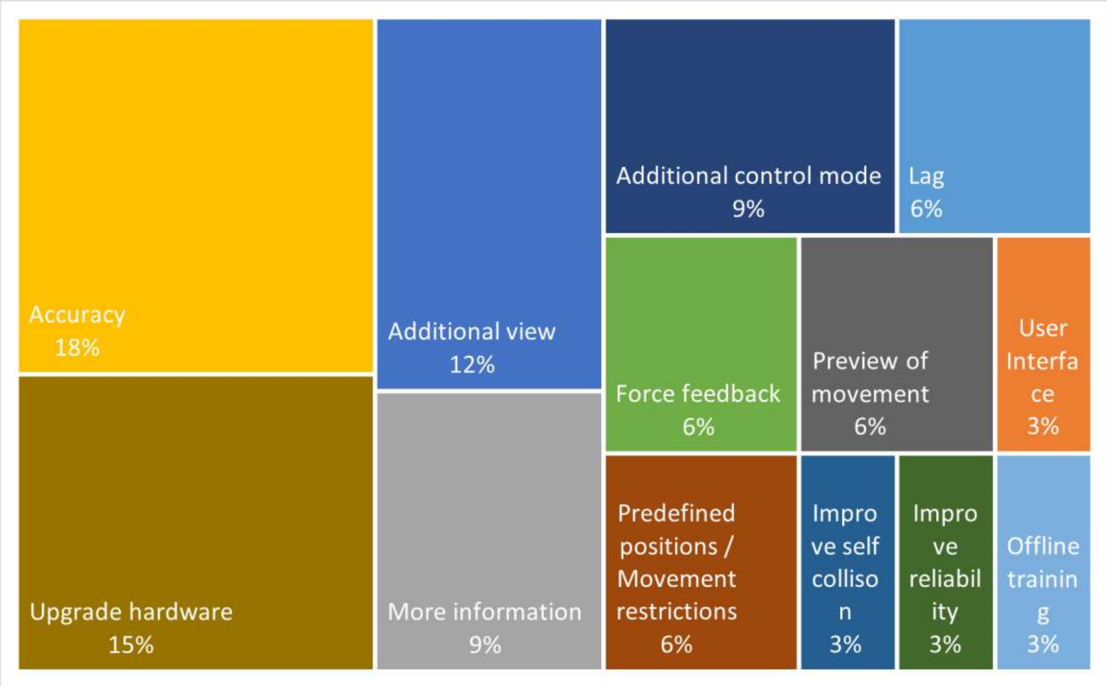

Figure 4.12: Free text answers categories.

What was apparent through the interviews was the participants perception of the system as a prototype. They did not judge the system too harshly for its reliability, thinking that these issues would resolve itself with further development. Multiple participants thought that the angle of the grippers in the VRCR differed from the robot's grippers or that the robot's grippers lagged behind. The researcher tried to verify that the angles differ but came to the conclusion that this was never the case, instead these complains stem from a misunderstanding of the relative control scheme combined with the delayed reaction time of the WBC package. Minimizing the delay of the system as much as possible and adding additional visual cues showing if WBC is still acting on its last input could improve the experience significantly.

4.3 Improvements

Interpreting the participants' feedback and the researchers own impressions, the prototype could be improved in multiple dimensions. The following describes potential software and hardware improvements that fundamentally change the operation of the system.

Software

First of all, without changing the principal way of teleoperating the robot, the general interface could be touched upon. The button which rotates the robots 3D model could be exchanged for a mirror allowing the user to see the model from the front. This has been adapted by the Reachy VR implementation seen in figure 4.13. Rotating the model was not used by most participants and when used created confusion on which arm they control.

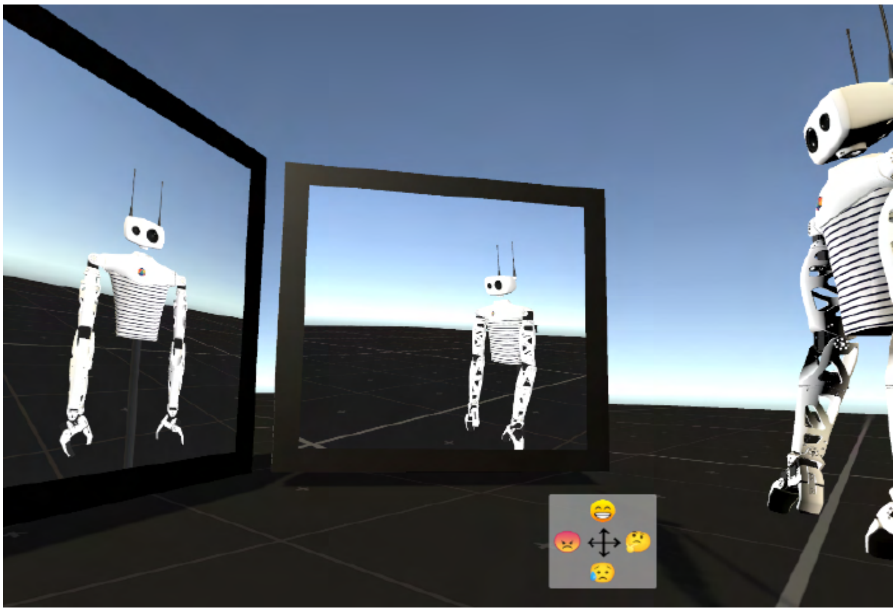

Figure 4.13: Reachy VR with virtual mirrors [Pol].

One point overlooked was the scaling of the positional movement between controllers and end-effector. The prototype worked with a fixed scaling, however giving the user control over the scaling would allow them more precise control. Another potential adaptation that could lessen the discrepancy between the user's expectation and the actual movement of the gripper would be changing from continuous IK to step wise. Instead of updating the pose every update loop the pose could only be updated when the WBC button is released. Only after that movement is completed, the next step of movement can be commanded. This would inevitably lower the potential speed of completing tasks but it might lower the learning curve enough to make this compromise acceptable. Additionally, what was mentioned by multiple particpants, was that they were missing a way to reset the grippers poses as well as no way to switch directly between some default orientations (e.g. a button press could cycle through five default orientations: forward, upward, down, right and left). This can be easily implemented and should be tested as usability could increase greatly through this addition. A general theme of comments provided by participants was the lack of both visual and haptic feedback. As the robot is equipped with sensors capable of detecting collisions these can be transmitted to the user with haptic feedback of the controllers. Ways to increase visual information provided will be discussed in section 5.1. Finally, the prototype did not implement any way of locomotion in the VRCR beyond real-walking in the VRCR. An established way of locomotion from other VR applications such as teleportation[Bol17] should be added to the system so users could see the model from different angles as well as freely chose the position of control to their liking.

Hardware

As this study was conceived as a feasibility study, the hardware used was generally chosen by their cost effectiveness. Excluding the VR headset as well as the robot which have been discussed previously, the hardware added to the robot can be improved drastically. Instead of of using two webcams, a bespoke stereo camera should be used to enable better depth perception. Both gripper cams could also be exchanged for stereo cameras. The webcams used were misaligned more often than not, while the grippers cams were very error-prone. Exchanging these would greatly increase usability as well as precision.

CHAPTER 5

Summary

The summary of this study includes an outlook in which ways VR teleoperation could potentially improve in the future, as well as explore how robot teaching can build upon VR teleoperation. This is followed by a conclusion summarizing the most important aspects of the study as well concluding in regards to the research questions.

5.1 Outlook

The following will discuss further developments and potential directions for VR teleoperation. Not only is the extension of the AHM itself explored, but potentially useful hardware developments in VR headsets.

5.1.1 Puppeteer Control Scheme

ID was added to the Articulation Body component in Unity with Version 2022.1 in May 2022. Most importantly, this allows the developer to have access to the virtual drive forces (how much force is needed to counteract outside forces), as well as gravity and coriolis forces applied to the joints. Using these values a *gravity compensation* mode can be simulated in Unity. Tiago++ has a *gravity compensation* mode using the same principle through the preinstalled torque sensors. Using virtual *gravity compensation* the operator can freely grab virtual parts of the robot model and move them to the desired pose and the real robot would follow accordingly. This can be used to construct a control mode similar to the one using IK: Instead of calculating how to reach a given end-effector pose, the operator could 'pull' and 'push' the end-effector to the desired position using Unity's physics simulation. Wrist rotation could be handled separately, either directly applying controller rotation or having a separate wrist rotation mode.

The links of the arms could also be moved separately. Since an end-effector pose can be reached with different arm configurations this would allow more control for the operator, e.g. the elbow of the robot could be lifted in a specific way to reach into a narrow space inside a lab device.

A similar concept has been researched for 3D skeletal articulation in animation for movies and video games. Jacobsen et al. developed a physical input device to control a 3D character. This is function reversed to what the Puppeteer Control Scheme (PCS) would achieve. Numaguchi et al. developed another device true to this idea.

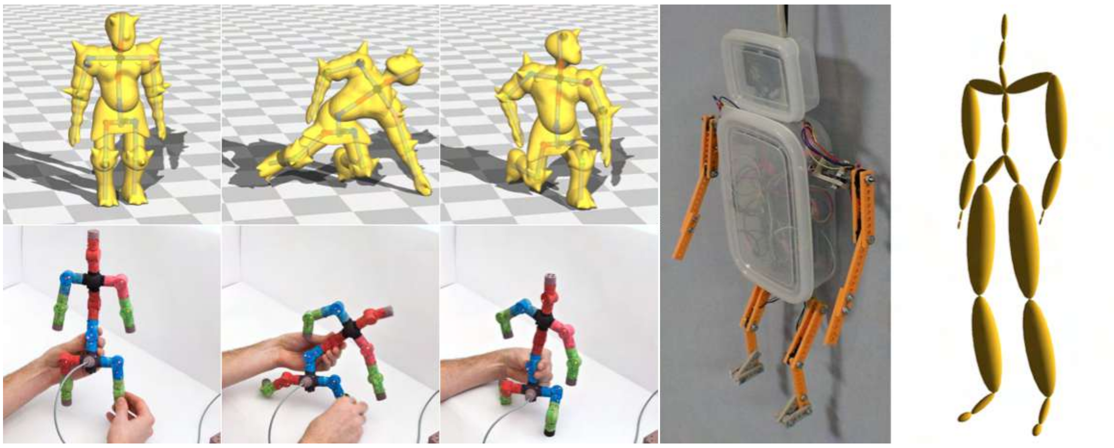

Figure 5.1: Input devices for character articulation [Jac+14], [Num+11].

This development happened during the end-stage of this work so it could not be tested, the author believes that using *gravity compensation* to allow the operator to manually move each part of the robot into the right pose could be used in high effect to increase usability as well as accuracy. Especially in combination with advancements to the AHM proposed below the operator could freely move in VR and then interact with the 3D model. Recently Bocquelet et al. presented a tool to quickly author character poses in Unity[Boc+22] which will make this development achievable very soon.

It will become essential at the stage of lab automation where the mobile robot already operates autonomously. When small malfunctions occur, such as the robot getting stuck, the operator could inspect the problem virtually and remotely move the robot's extremities carefully out of the way and restart whatever process it was trying to achieve.

5.1.2 Eye-tracking and hand-tracking

As established in section 3.2, VR headsets are the cheapest and most readily available way of enabling stereo vision as well as three dimensional pose tracking of the hands. In this work a standalone headset was purposefully chosen as these are the least cumbersome to use and install for users. Currently only the Oculus Quest 2, HTC Vice Focus 3 and Pico Neo 3 Pro/Eye fulfill the requirements to be used in teleoperation. One major advancement on the horizon is eye-tracking which the Pico Neo Pro Eye already offers. Stein et al. [Ste+21] offer a comparison of different eye tracking VR headsets. Essentially, eye tracking enables the system to control the stereo camera vergence angles. Out of focus video was a common problem during tests, that can be fixed with bespoke camera hardware and eye tracking. On top of controlling vergence angles, eye tracking adds an additional mode of input. Eyes can be closed, which can be interpreted as input and the tracking might be able to be used as a pointing device. These additional inputs could be used to free the operator from controllers and fully rely on the hand-tracking abilities of the headsets, which combined with the puppeteer control model discussed above could enable a very user friendly and intuitive way of teleoperation.

5.1.3 Mode Fusion/Top Down Mode

Section 2.2.3 (page 13) discussed the pre-established control modes for VR teleoperation and this thesis extended upon them with the AHM and the PCS. As mentioned in 2.2.3, control modes are not mutually exclusive but can be fused together, most easily by changing between them. Beyond that, parts of Cyber-Physical and Homunculus Model can be merged even further to what AHM does. All implementations using the Homunculus Model discussed in this work as well as the implementation developed for the visualization of sensors most mobile manipulators provide to perceive their environment, while the Cyber-Physical Model is based around these sensors.

The fundamental difference between Cyber-Physical and Homunculus Model is that in the former the operator and robot share the same cyber-physical space, while in the latter the operator acts in a completely separate virtual space, the VRCR. The introduction of the 3D model representation of the robot into the VRCR might create the impression of a shared space, the model is however just a visual aid. Once the aforementioned puppeteer control mode is introduced into the Homunculus Model, the perception of the VRCR as a separate virtual space is not broken in principle, the model is just a tool to control the real robot, however the operator might perceive it as if they share the same space. Even though most certainly useful, this false perception could be further enhanced by introducing other sensor data like textured point clouds of the robot environment around the 3D model in the VRCR. This can most likely be corrected by visually highlighting that the robot and the sensor data are projected into the VRCR and is subject to further research.

On top of these advancements of the Homunculus Model, a simplified top-down view locomotion control mode can be developed. This can be understood as a variation of the Piloting Model and should serve as the default way of moving the robot. This can also be developed to work with traditional computer input, so that the VR headset is only in use when needed. Alternatively, this top down view of the robot and its environment could be included in the VRCR as additional visual feedback. Surround view already exists in other applications such as cars. Boston Dynamics' Spot robot also includes a rudimentary surround view, see figure 5.2.

Figure 5.2: Surround view [Bos22], [Sha+19].

5.1.4 Robot teaching

Apart from further optimising the user-machine interaction, one use case beyond the intrinsic value of teleoperation is robot teaching. The vision is to enable autonomous operation after having learned actions from how the human operator performed them during telemanipulation. Compared to other domains, gathering high-quality input in teleoperation is more difficult since no established control interfaces exist. Robot teaching through imitation learning can be divided into behavioral cloning and inverse reinforcement learning. Zhang et al. [Zha+18] already demonstrated the feasibility of behavorial cloning in VR teleoperation.

5.2 Conclusion

After having established the environment in which the developed prototype should act, the author begun building the system with this in mind afterwards. After having it continuously improve through multiple prototyping iterations and then examined through the user-tests, a conclusive review can now be constructed. In order to do so each research question will be successively reflected upon.

Q1 How well is the implemented prototype suited to control a mobile robot in laboratory environment?

This question has to be answered in two parts. First of all, the prototype is undoubtedly suited to control a mobile robot, in particular the Pal Robotics Tiago++. The user tests proved that everyone can move the robot around and at least use it for observing tasks, while most were able to control the arms for semi complex tasks. Expert users are, while time consuming, able to reliably grab and transport objects. Since these tests were done in a laboratory environment the baseline answer would be that the prototype can be controlled in a laboratory environment. To which degree this is possible is however up for evaluation. As a prototype the system over performed in regards to the researcher's expectation, the system is, however, in no state for daily use. All core functionalities are there and could be made to work with enough perseverance from the user which is the most important conclusion regarding this question. No, the system is not capable to be used in a laboratory environment without extreme precautions, but enabling this capability just requires further development of already existing functionality. One can confidently expect a later iteration to be used in a laboratory environment without limitations.

Q2 How well can the negative effects of a VR headset in teleoperation use be negated through the developed solution?

This question was intrinsic to the development stage and the reason why the Homunculus control model was chosen over the Mimicking model. Answering this question most conclusively can be done through examining the data provided by the user tests. Out of eleven participants two mentioned adverse effects, while these were very mild for one of them. The other participant complained for dizziness for the remainder of the day. So, adverse effects were not completely eliminated through the measures taken to counteract them, but they were in no way a central complain of participants. There has been no long time study concerning adverse effect stemming from the time frame of this study, however the two user using the system most often (researcher + one participant) did not feel any adverse effects.

To answer the actual aim of the question: VR teleoperation can be further explored, the negative effects of prolonged use of the VR headset do not principally outweigh the benefits provided by it. This is further cemented by participants defending the necessity of the headset over a screen in the questionnaire.

Q3 What are the missing links to enable robust, usable teleoperation through a VR headset?

The scope of this question can be widened to not only answer the missing links regarding the developed prototype, but to answer it regarding the general state of VR teleoperation. This has to be done since the prototype was not developed to be an artefact on its own but to examine the current state of the topic in general. I.e. listing suboptimally performing hardware of the prototype does not answer the spirit of the question. What is holding back VR teleoperation the most is the software components needed to integrate between headset and robot not being mature enough. Several island solutions exist cited throughout this thesis, most of them being targeted towards one expert user teleoperating one specific robot. In this regard there are no limits to the capabilities of teleoperation, but at that point the cost benefits of using a VR for teleoperation are already outweighed by the advantages of exoskeletons, force-feedback arms and the like. The real benefit of VR headsets are their availabilty and usability. A standalone VR headset is the only hardware, that combines input, output and computing into one device outside of a traditional computer which it triumphs by its pose tracking of the controllers. Therefore it is the only device that enables easy scalability, enabling one robot to be controlled by multiple users. What is holding back robust, usable teleoperation is time and research.

As has been demonstrated in this study all functionality can be achieved, in principal, in relatively short time span, by a single researcher. The feasibility of this concept is conclusively shown in this work. What lies ahead is standardizing VR teleoperation, allowing researchers easier access and thus enabling further research into topics such as robot teaching trough teleoperation. It is feasible to develop a VR teleoperation system that functions in the realm of a prototype it is not feasible to expect every researcher to do so in order to attain new knowledge. Developments such as the Reachy robot can act as basis towards further research on VR teleoperation. Since the coexistence of commercially available VR headsets and adequate robots has just been seven years [WP20], one can expect future developments to pick up traction. Especially when considering the promising results from this study as well as other papers cited.

Conflict of interest statement

Janek Janßen and Ádám Wolf are employees of Baxalta Innovations GmbH, Vienna, ´ Austria.

Acknowledgements

This work was funded by Baxalta Innovations GmbH, a Takeda company.

List of Figures

List of Tables

Glossary

pose Describes position and orientation of an object. 15, 18, 43–45, 48, 66, 69, 70, 74, 79

robot teaching The act of adding capabilities through software, e.g. programming, demonstration, guidance. 2, 69, 72, 74

transform Transforms (transformations) in ROS are packages (*tf* and *tf2*), that are used to describe the robot's coordinate frames' relative poses over time. In Unity an object's transform (equiv. coordinate frame) is used store and manipulate its pose and scale. Cf. *Quaternion*, *Transformation matrix*. 43, 44

Acronyms

AHM Augmented Homunculus Model. 5–7, 25, 48, 49, 53, 56, 69–71

AR Augmented Reality. 11, 33

DoF Degrees of Freedom. 15, 28, 32

FOV field of view. 35

HCI Human-Computer Interaction. 25

ID Inverse Dynamics. 45, 69

IK Inverse Kinematics. 34, 43–45, 48, 66, 69

IPD interpupillary distance. 28

MR Mixed Reality. 7, 9, 11

PCS Puppeteer Control Scheme. 70, 71

ROS Robot Operating System. 2, 3, 19–21, 23, 31–33, 35, 36, 39, 43, 44, 79

SCARA Selective Compliance Assembly Robot Arm. 10

URDF Unified Robot Description Format. 33, 34, 41, 42, 44

URH Unity Robotics Hub. 3, 20, 33–35, 41, 43, 45

V4L Video4Linux. 35

VR Virtual Reality. 3–7, 9, 11, 13–16, 18–22, 25–33, 35, 36, 43, 44, 48, 49, 53–55, 57, 60, 61, 64, 66, 67, 69–74, 77

VRCR Virtual Reality Control Room. 17, 18, 28, 44, 48, 49, 61, 65, 66, 71

WBC whole body control. 44–50, 57, 58, 61, 64–66

XR Extended Reality. 1

Bibliography

[Han76] Hans P. Moravec. *Techniques Towards Automatic Visual Obstacle Avoidance*. Tech. rep. Stanford: Stanford University, Feb. 1976. URL: https://frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1977/aip.txt.

[CT88] John M. Carroll and John C. Thomas. „FUN“. In: *ACM SIGCHI Bulletin* 19.3 (Jan. 1988), pp. 21–24. ISSN: 0736-6906. DOI: 10.1145/49108.1045604.

[Dav89] Fred D. Davis. „Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology“. In: *MIS Quarterly* 13.3 (Sept. 1989), p. 319. ISSN: 02767783. DOI: 10.2307/249008.

[Mil+95] Paul Milgram et al. „Augmented reality: a class of displays on the reality-virtuality continuum“. In: *SPIE 2351*. Ed. by Hari Das. Witmer & Singer, Dec. 1995, pp. 282–292. DOI: 10.1117/12.197321.

[Bro96] John Brooke. „SUS: A 'Quick and Dirty' Usability Scale“. In: *Usability Evaluation In Industry*. Ed. by Patrick W. Jordan et al. 1st Edition. London: CRC Press, June 1996. Chap. 21, pp. 189–194. ISBN: 9780429157011. DOI: 10.1201/9781498710411.

[Bec99] Kent Beck. *Extreme Programming Explained: embrace change*. New York: Addison-Wesley, Oct. 1999. ISBN: 0201616416. DOI: 10.5555/318762.

[Kaw+99] K. Kawabata et al. „Mobile robot teleoperation using local storage“. In: *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*. Vol. 2. IEEE, Aug. 1999, pp. 1141–1145. ISBN: 0-7803-5446-X. DOI: 10.1109/CCA.1999.801133.

[SR99] K.J. Schilling and H. Roth. „Control interfaces for teleoperated mobile robots“. In: *1999 7th IEEE International Conference on Emerging Technologies and Factory Automation. Proceedings ETFA '99 (Cat. No.99TH8467)*. Vol. 2. University of Applied Sciences FH Ravensburg-Weingarten. Barcelona: IEEE, 1999, pp. 1399–1403. ISBN: 0-7803-5670-5. DOI: 10.1109/ETFA.1999.813153.

[SJS99] Sun-Gi Hong, Ju-Jang Lee, and Seungho Kim. „Generating artificial force for feedback control of teleoperated mobile robots“. In: *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*. Vol. 3. Kyongju: IEEE, Aug. 1999, pp. 1721–1726. ISBN: 0-7803-5184-3. DOI: 10.1109/IROS.1999.811726.

[HS06] Peter F. Hokayem and Mark W. Spong. „Bilateral teleoperation: An historical survey“. In: *Automatica* 42.12 (Dec. 2006), pp. 2035–2057. ISSN: 00051098. DOI: 10.1016/j.automatica.2006.06.027.

[CHB07] Jessie Y. C. Chen, Ellen C. Haas, and Michael J. Barnes. „Human Performance Issues and User Interface Design for Teleoperated Robots“. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37.6 (Nov. 2007), pp. 1231–1245. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.905819.

[AG10] Angelos Amanatiadis and Antonios Gasteratos. „Stereo Vision System for Remotely Operated Robots“. In: *Remote and Telerobotics*. Ed. by Nicolas Mollet. London: InTech, Mar. 2010. Chap. 4, pp. 59–72. DOI: 10.5772/9247.

[Che10] Ryad. Chellali. „Tele-operation and Human Robots Interactions“. In: *Remote and Telerobotics*. Ed. by Nicolas Mollet. London: InTech, Mar. 2010. Chap. 6, pp. 91–112. DOI: 10.5772/9248.

[Boh+11] Jonathan Bohren et al. „Toward practical semi-autonomous teleoperation: Do what i intend, not what i do“. In: *Advanced Robotics and its Social Impacts*. IEEE, Oct. 2011, pp. 20–23. ISBN: 978-1-4673-0796-3. DOI: 10.1109/ARSO.2011.6301974.

[Num+11] Naoki Numaguchi et al. „A puppet interface for retrieval of motion capture data“. In: *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA ’11*. Ed. by Stephen N. Spencer, Adam Bargteil, and Michiel van de Panne. New York, New York, USA: ACM Press, Aug. 2011, p. 157. ISBN: 9781450309233. DOI: 10.1145/2019406.2019427.

[Ser11] Daniel Serrano. *Introduction to ROS - Robot Operating System*. Tech. rep. NATO Science & Technology Organization - STO, 2011. URL: https://www.sto.nato.int/publications/STO%20Educational%20Notes/STO-EN-SCI-271/EN-SCI-271-06.pdf.

[BM12] Michel Beaudouin-Lafon and Wendy E. Mackay. „Prototyping Tools and Techniques“. In: *The Human–Computer Interaction Handbook*. Ed. by Julie A. Jacko. 3rd Edition. Boca Raton: CRC Press, May 2012. Chap. 47, pp. 1081–1104. ISBN: 9780429103971. DOI: 10.1201/b11963-ch-47.

[BB12] Jeanette Blomberg and Mark Burrell. „An Ethnographic Approach to Design". In: *The Human–Computer Interaction Handbook*. Ed. by Julie A. Jacko. 3rd Edition. Boca Raton: CRC Press, May 2012. Chap. 45, pp. 1025–1051. ISBN: 9780429103971. DOI: 10.1201/b11963-ch-45.

[Foo13] Tully Foote. „tf: The transform library". In: *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. Woburn: IEEE, Apr. 2013, pp. 1–6. ISBN: 978-1-4673-6225-2. DOI: 10.1109/TePRA.2013.6556373.

[BLB14] Peter Birkenkampf, Daniel Leidner, and Christoph Borst. „A knowledge-driven shared autonomy human-robot interface for tablet computers". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, Nov. 2014, pp. 152–159. ISBN: 978-1-4799-7174-9. DOI: 10.1109/HUMANOIDS.2014.7041352.

[Jac+14] Alec Jacobson et al. „Tangible and modular input device for character articulation". In: *ACM Transactions on Graphics* 33.4 (July 2014), pp. 1–12. ISSN: 0730-0301. DOI: 10.1145/2601097.2601112.

[Zie+14] Dina Zielinski et al. „iPipet: sample handling using a tablet". In: *Nature Methods* 11.8 (Aug. 2014), pp. 784–785. ISSN: 1548-7091. DOI: 10.1038/nmeth.3028.

[Aut15] David H. Autor. „Why Are There Still So Many Jobs? The History and Future of Workplace Automation". In: *Journal of Economic Perspectives* 29.3 (Aug. 2015), pp. 3–30. ISSN: 0895-3309. DOI: 10.1257/jep.29.3.3.

[CB15] Patrick Courtney and Francesco Becchi. *White paper on Laboratory Robotics in Europe Status and prospects within Horizon2020*. Tech. rep. Vienna: European Robotics Forum, Sept. 2015. URL: https://sila-standard.com/downloads-public/.

[Kam+15] Hyeong Ryeol Kam et al. „RViz: a toolkit for real domain data visualization". In: *Telecommunication Systems* 60.2 (Oct. 2015), pp. 337–345. ISSN: 1018-4864. DOI: 10.1007/s11235-015-0034-5.

[Bol17] Costas Boletsis. „The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology". In: *Multimodal Technologies and Interaction* 1.4 (Sept. 2017), p. 24. ISSN: 2414-4088. DOI: 10.3390/mti1040024.

[GC17] Paul Groth and Jessica Cox. „Indicators for the use of robotic labs in basic biomedical research: a literature analysis". In: *PeerJ* 5.11 (Nov. 2017), e3997. ISSN: 2167-8359. DOI: 10.7717/peerj.3997.

[Bau+18] Adrian S. Bauer et al. „Inferring Semantic State Transitions During Telerobotic Manipulation". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 1–9. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8594458.

[KN18] Tomáš Kot and Petr Novák. „Application of virtual reality in teleoperation of the military mobile robotic system TAROS“. In: *International Journal of Advanced Robotic Systems* 15.1 (Jan. 2018), p. 172988141775154. ISSN: 1729-8814. DOI: 10.1177/1729881417751545.

[LFR18] Jeffrey I. Lipton, Aidan J. Fay, and Daniela Rus. „Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing“. In: *IEEE Robotics and Automation Letters* 3.1 (Jan. 2018), pp. 179–186. ISSN: 2377-3766. DOI: 10.1109/LRA.2017.2737046.

[Whi+18] David Whitney et al. „ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots“. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018, pp. 1–9. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8593513.

[Zha+18] Tianhao Zhang et al. „Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation“. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Vol. abs/1710.04615. IEEE, May 2018, pp. 5628–5635. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8461249.

[Ann+19] Vivek Annem et al. „Towards Remote Teleoperation of a Semi-Autonomous Mobile Manipulator System in Machine Tending Tasks“. In: *Volume 1: Additive Manufacturing; Manufacturing Equipment and Systems; Bio and Sustainable Manufacturing*. Vol. 1. Pennsylvania: American Society of Mechanical Engineers, June 2019. ISBN: 978-0-7918-5874-5. DOI: 10.1115/MSEC2019-3027.

[KKK19] Yeon Kang, Donghan Kim, and Kwangjin Kim. „URDF Generator for Manipulator Robot“. In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, Feb. 2019, pp. 483–487. ISBN: 978-1-5386-9245-5. DOI: 10.1109/IRC.2019.00101.

[Kit+19] Richard Kitney et al. „Enabling the Advanced Bioeconomy through Public Policy Supporting Biofoundries and Engineering Biology“. In: *Trends in Biotechnology* 37.9 (Sept. 2019), pp. 917–920. ISSN: 01677799. DOI: 10.1016/j.tibtech.2019.03.017.

[MB19] Patrick Mania and Michael Beetz. „A Framework for Self-Training Perceptual Agents in Simulated Photorealistic Environments“. In: *2019 International Conference on Robotics and Automation (ICRA)*. Montreal: IEEE, May 2019, pp. 4396–4402. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793474.

[NM19] Michael Nebeling and Katy Madier. „360proto: Making Interactive Virtual Reality & Augmented Reality Prototypes from Paper“. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New

York, NY, USA: ACM, May 2019, pp. 1–13. ISBN: 9781450359702. DOI: 10.1145/3290605.3300826.

[PAL19] „TIAGo - PAL Robotics: Leading service robotics“. URL: https://pal-robotics.com/robots/tiago/. *Last accessed: Oct. 21, 2021.*

[Sha+19] Xuan Shao et al. „Revisit Surround-view Camera System Calibration“. In: *2019 IEEE International Conference on Multimedia and Expo (ICME)*. Vol. 2019-July. IEEE, July 2019, pp. 1486–1491. ISBN: 978-1-5386-9552-4. DOI: 10.1109/ICME.2019.00257.

[Thu+19] Kerstin Thurow et al. „Multi-floor laboratory transportation technologies based on intelligent mobile robots“. In: *Transportation Safety and Environment* 1.1 (July 2019), pp. 37–53. ISSN: 2631-4428. DOI: 10.1093/tse/tdy002.

[TB19] „Mobile Lab-Robot "Kevin"“. URL: https://www.labvolution.de/product/mobile-lab-robot-kevin-/176612/X256500. *Last accessed: Aug. 25, 2022.*

[Vil19] „TIAGo++, the robot you need for bi-manual tasks“. URL: https://blog.pal-robotics.com/tiago-bi-manual-robot-research/. *Last accessed: Aug. 21, 2022.*

[Bej+20] Bence Bejczy et al. „Mixed Reality Interface for Improving Mobile Manipulator Teleoperation in Contamination Critical Applications“. In: *Procedia Manufacturing* 51 (2020), pp. 620–626. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.10.087.

[Bur+20] Benjamin Burger et al. „A mobile robotic chemist“. In: *Nature* 583.7815 (July 2020), pp. 237–241. ISSN: 0028-0836. DOI: 10.1038/s41586-020-2442-2.

[GM20] Rinat R. Galin and Roman V. Meshcheryakov. „Human-Robot Interaction Efficiency and Human-Robot Collaboration“. In: *Studies in Systems, Decision and Control*. Vol. 272. Springer, 2020, pp. 55–63. DOI: 10.1007/978-3-030-37841-7_5.

[HD20] Ian Holland and Jamie A. Davies. „Automation in the Life Science Research Laboratory“. In: *Frontiers in Bioengineering and Biotechnology* 8 (Nov. 2020). ISSN: 2296-4185. DOI: 10.3389/fbioe.2020.571777.

[Kla+20] Tobias Klamt et al. „Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance“. In: *Journal of Field Robotics* 37.5 (Aug. 2020), pp. 889–919. ISSN: 1556-4959. DOI: 10.1002/rob.21895.

[Mac+20] Mateusz Maciaś et al. „Measuring Performance in Robotic Teleoperation Tasks with Virtual Reality Headgear“. In: *Advances in Intelligent Systems and Computing*. Vol. 920. Springer Verlag, 2020, pp. 408–417. DOI: 10.1007/978-3-030-13273-6_39.

[SR20] Kai von Szadkowski and Simon Reichel. „Phobos: A tool for creating complex robot models". In: *Journal of Open Source Software* 5.45 (Jan. 2020), p. 1326. ISSN: 2475-9066. DOI: 10.21105/joss.01326.

[Whi+20] David Whitney et al. „Comparing Robot Grasping Teleoperation Across Desktop and Virtual Reality with ROS Reality". In: *International Symposium on Robotics Research*. Ed. by Nancy M. Amato et al. Puerto Varas: Springer Cham, Dec. 2020, pp. 335–350. DOI: 10.1007/978-3-030-28619-4_28.

[WP20] Murphy Wonsick and Taskin Padir. „A Systematic Review of Virtual Reality Interfaces for Controlling and Interacting with Robots". In: *Applied Sciences* 10.24 (Dec. 2020), p. 9051. ISSN: 2076-3417. DOI: 10.3390/app10249051.

[Che+21] Jouh Yeong Chew et al. „Adaptive attention-based human machine interface system for teleoperation of industrial vehicle". In: *Scientific Reports* 11.1 (Dec. 2021), p. 17284. ISSN: 2045-2322. DOI: 10.1038/s41598-021-96682-0.

[Lim21] „Can robots speed up drug discovery?" URL: https://www.limaohio.com/news/business/464986/can-robots-speed-up-drug-discovery. *Last accessed: Nov. 4, 2021.*

[Ste+21] Niklas Stein et al. „A Comparison of Eye Tracking Latencies Among Several Commercial Head-Mounted Displays". In: *i-Perception* 12.1 (Jan. 2021), p. 204166952098333. ISSN: 2041-6695. DOI: 10.1177/2041669520983338.

[TCP21] Alexandru Tăbușcă, Cristina Coculescu, and Mironela Pirnau. „General consideration regarding the development of games using Unity technology". In: *Journal of Information Systems & Operations Management (JISOM)* 15.2 (Dec. 2021), pp. 267–283. ISSN: 1843-4711. URL: http://jisom.rau.ro/.

[Uni21] „Unity Documentation". URL: https://docs.unity3d.com/2022.2/Documentation/ScriptReference/.

[Boc+22] Florent Bocquelet et al. „AI and Physics Assisted Character Pose Authoring". In: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Real-Time Live!* New York, NY, USA: ACM, Aug. 2022, pp. 1–2. ISBN: 9781450393683. DOI: 10.1145/3532833.3538680.

[Bos22] „Boston Dynamics Support". URL: https://support.bostondynamics.com/. *Last accessed: Mai 25., 2022.*

[Bro] „VRcompare - The Internet's Largest VR & AR Headset Database". URL: https://vr-compare.com/standalone. *Last accessed: Mai 11., 2022.*

[Cen] „Intro - celiscas Webseite!" URL: www.celisca.de. *Last accessed: Aug. 25, 2022.*

[HG] „IVRE - An Immersive Virtual Robotics Environment". URL: https://cirl.lcsr.jhu.edu/research/human-machine-collaborative-systems/ivre/. *Last accessed: Apr. 4, 2022.*

[IEE] „ROBOTS: Your Guide to the World of Robotics - IEEE". URL: https://robots.ieee.org/. *Last accessed: Mai 13., 2022.*

[Ocu] „Oculus Quest 2: Our Most Advanced New All-in-One VR Headset“. URL: https://www.oculus.com/quest-2/. *Last accessed: Jan. 27, 2022.*

[Pol] „Reachy by Pollen Robotics, an open source programmable humanoid robot“. URL: https://www.pollen-robotics.com/. *Last accessed: Apr. 4, 2022.*

[ROSa] „ROS: Home“. URL: https://www.ros.org/. *Last accessed: Nov. 3, 2021.*

[ROSb] „pr2_surrogate - ROS Wiki“. URL: http://wiki.ros.org/pr2_surrogate. *Last accessed: Apr. 5, 2022.*

[Sil] „SILA Rapid Integration“. URL: https://sila-standard.com/. *Last accessed: Aug. 25, 2022.*

[Unia] „Unity Real-Time Development Platform | 3D, 2D VR & AR Engine“. URL: https://unity.com/. *Last accessed: Jan. 27, 2022.*

[Unib] „Unity Robotics Hub“. URL: https://github.com/Unity-Technologies/Unity-Robotics-Hub. *Last accessed: Jan. 27, 2022.*

Appendix

A Software architecture

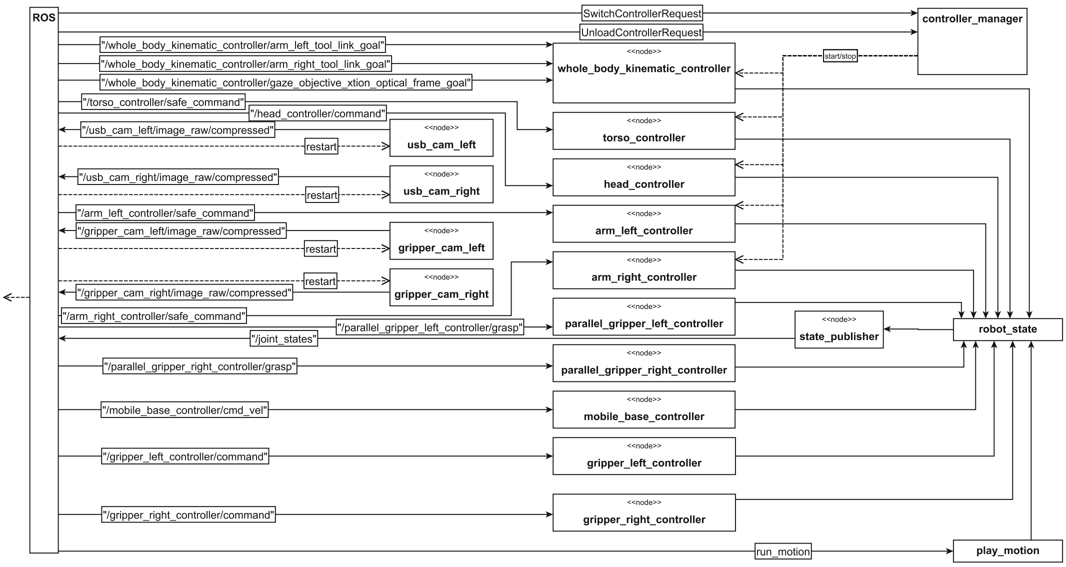

Figure 1: Software architecture, ROS.

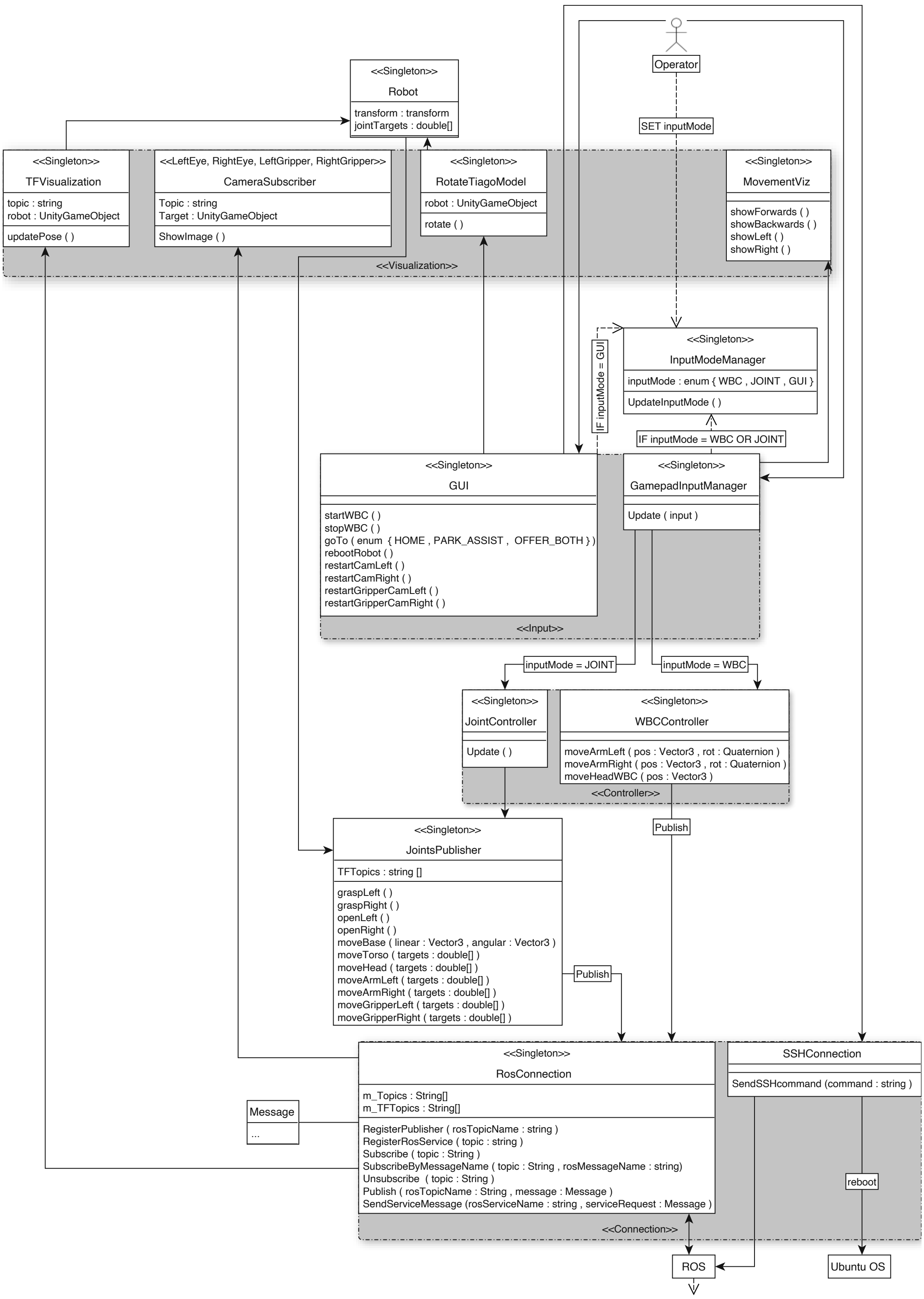

Figure 2: System overview, Unity.

B Questionnaire

Tiago VR User Tests

* Dieses Formular wird Ihren Namen aufzeichnen. Bitte tragen Sie Ihren Namen ein.

Introduction

This is a short survey in order to asses your experience using the prototype.

Please answer every question honestly. Your feedback is needed in order to asses the current state of this prototype and to determine future paths of development.

1

What is your name? (no answer if you want to stay anonymous)

3/31/2022

Questions about previous experience

scale:
1 means "no experience at all"
5 means "a lot"

2

Did you use any previous iteration of this prototype before?

- ◯ Yes
- ◯ No

3

Did use this version of the prototype for the first time?

- ◯ Yes
- ◯ No

3/31/2022

4

How much experience do you have playing video games with a game controller?

1	2	3	4	5
◯	◯	◯	◯	◯

5

How much experience do you have using a virtual reality headset?

1	2	3	4	5
◯	◯	◯	◯	◯

3/31/2022

Enjoyment

scale:
1 means "not at all"
5 means "a lot"

6

How much did you enjoy using the system?

1	2	3	4	5
◯	◯	◯	◯	◯

7

How much would you enjoy using this system routinely in your workday?

1	2	3	4	5
◯	◯	◯	◯	◯

3/31/2022

Ease of control

scale:
1 means "not at all"
5 means "a lot"

8

How much did you like the overall user interface? How intuitive did it feel to you?

1 2 3 4 5

9

How confident did you feel while driving the robot around?

1 2 3 4 5

10

How confident did you fell controlling the arms?

1 2 3 4 5

3/31/2022

11

Do you feel like you were hindered more by your lack of experience (1) or by the system's ability to do what you command it to do? (5)

1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯

12

Do you think using a virtual reality headset is necessary or would you rather use a traditional screen?

◯ Headset

◯ Screen

3/31/2022

Well-being

scale:
1 means "not at all"
5 means "a lot"

13

While using the system, did you have any feeling of disorientation (5) or did you feel comfortable beeing in a virtual room (1) ?

1	2	3	4	5
○	○	○	○	○

14

Did you feel any adverse effect from using the system? Dizziness, nausea, headaches, ...

○ Yes

○ No

15

After the test, did you feel like you could have continued using the system without a break? Or were you glad you could stop wearing the headset?

○ Could have continued

○ Glad it was over

3/31/2022

Feedback

In this section you can give feedback on how to improve the system.

16

Do you have any ideas on how to improve the system or would you change anything?

17

Anything else you want to mention?

3/31/2022

C References figure 2.9

Ref	Year	Category	Robot Type	Contribution
[CS16]	2016	Usability	(Virtual) Mobile	Identifies user preferences between using a traditional computer interface over an immersive VR interface for teleoperation
[DSL16]	2016	Interaction	Mobile	Develops a collaborative human-robot system to accomplish real-time mapping in VR
[Lee+16]	2016	Interaction	Mobile	Develops a visual programming system to define navigation tasks
[The+16]	2016	Visualization	Humanoid	Develops a method to use stereo panoramic reconstruction to reduce perceived visual latency during teleoperation
[XPH16]	2016	Visualization	Manipulator	Evaluates the affects of different viewpoints on success when teleoperating a construction robot
[Rol+17]	2017	Interaction	(Virtual) Mobile & Aerial	Investigates the utility of predictive capabilities in VR interfaces for multi-robot teams using a traditional interface as a baseline
[The+17]	2017	Usability	Manipulator	Compares a developed VR programming interface with a direct manipulation interface and a keyboard, mouse, and monitor interface
[MI17]	2017	Infrastructure	N/A	Develops an open-source cloud-based software architecture to interface ROS with Unity
[Bri+18]	2018	Visualization	Dual-Arm Manipulator	Evaluates using virtual features to display task-related information to improve operator performance in completing teleoperation pick-and-place tasks
[JOK18]	2018	Robot Control and Planning	Manipulator	Compares different VR interaction techniques for teleoperation
[Koh+18]	2018	Visualization	Manipulator	Develops a method to efficiently process and visualize point-clouds in VR

Ref	Year	Category	Robot Type	Contribution
[KN18]	2018	Visualization	Mobile with Manipulator	Evaluates the best way to visualize stereo cameras inside a VR headset to minimize motion sickness
[LFR18]	2018	Robot Control and Planning	Dual-Arm Manipulator	Develops a teleoperation framework that can quickly map user input to robot movement and vice-versa
[Zha+18]	2018	Visualization	(Virtual) Aerial	Evaluates the effects of visual and control latency in drones when using VR
[Bab+18]	2018	Infrastructure	N/A	Develops a framework to interface ROS with Unity
[Whi+18]	2018	Infrastructure	N/A	Develops an open-source framework to interface ROS with Unity
[CP19]	2019	Visualization	(Virtual) Mobile	Develops an image projection method that remove discrepancies between robot and user head pose
[FJ19]	2019	Interaction	Dual-Arm Manipulator	Evaluates using different controllers in teleoperation
[Gor+19]	2019	Interaction	Dual-Arm Manipulator	Develops a telemanipulation framework that incorporates a set of grasp affordances to simplify operation
[AHY19]	2019	Interaction	Humanoid (Bipedal)	Summarizes data visualization and interaction techniques of VR video games for adoption to VR robot interfaces
[Hir+19]	2019	Robot Control and Planning	Humanoid (MobileBase)	Develops teleoperation system that imitates user's upper body pose data in real-time
[Rol+19a]	2019	Usability	Mobile with Manipulator & Aerial	Compares a traditional interface to a VR interface for multi-robot missions
[Sto+19]	2019	Visualization	Mobile with Manipulator	Compares an immersive VR visualization to a monitor video-based visualization for robot navigation
[Su+19]	2019	Visualization	Manipulator	Compares a representative model visualization of the full environment to a real-time point cloud visualization of the real environment for teleoperation

Ref	Year	Category	Robot Type	Contribution
[Tso+19]	2019	Robot Control and Planning	Manipulator	Develops a framework that allows robot teleoperation through uses of a digital twin
[Van+19]	2019	Visualization	Manipulator	Investigates the influence of displaying different levels of environmental information has on task performance and operator situation awareness in VR robot interfaces
[Vem+19]	2019	Robot Control and Planning	Aerial	Develops an optimization based planner to control a painting drone in VR
[Yas+19]	2019	Robot Control and Planning	Aerial with Manipulator	Develops a teleoperation system for aerial manipulation that includes tactile feedback
[Gau+19]	2019	Robot Control and Planning	Dual-Arm Manipulator	Develops a deep correspondence model that maps user input to robot motion for teleoperation
[Xi+19]	2019	Robot Control and Planning	Dual-Arm Manipulator	Develops a predict-then-blend framework to increase efficiency and reduce user workload
[Ast+19]	2019	Infrastructure	N/A	Develops an open-source solution that help[sic!] calibrate VR equipment (HTC Vive) inside a robot cell (hardware-agnostic, only requires ROS-Industrial and MoveIt plugin)
[Rol+19b]	2019	Infrastructure	N/A	Defines a system architecture to work with multi-robot systems using ROS and Unity
[Bec+20]	2020	Visualization	Mobile	Develops and evaluates a human perception-optimized planner to reduce motion sickness
[Elo+20]	2020	Robot Control and Planning	Humanoid (Bipedal)	Develops a control architecture that utilizes a VR setup with an omni-directional treadmill to create a fully immersive teleoperation interface
[Het+20]	2020	Interaction	Dual-Arm Manipulator	Compares two different VR control interactions, position control and trajectory control, for robot operation

Ref	Year	Category	Robot Type	Contribution
[Mac+20]	2020	Usability	Mobile with Manipulator	Compares displaying camera streams on a monitor and displaying stereo cameras streams inside a VR headset for teleoperation
[Sun+20]	2020	Robot Control and Planning	Manipulator	Develops two robot controllers to decouple an operator from the robot's control loop for teleoperation
[Wan+20]	2020	Robot Control and Planning	Manipulator	Develops a method that estimates human intent in VR to control a welding robot
[WT20]	2020	Visualization	Aerial	Develops a controller that synchronizes a drone's movement with the user's head movement to reduce motion sickness
[Whi+20]	2020	Usability	Dual-Arm Manipulator	Compares a VR interface to traditional interfaces for teleoperation
[KBD20]	2020	Robot Control and Planning	Manipulator	Develops a motion planner using deep reinforcement learning to map the human workspace to the robot workspace for teleoperation

Table 1: References figure 2.9.

[CS16] Marvin A. Conn and Sharad Sharma. „Immersive Telerobotics Using the Oculus Rift and the 5DT Ultra Data Glove“. In: *2016 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, Oct. 2016, pp. 387–391. ISBN: 978-1-5090-2300-4. DOI: 10.1109/CTS.2016.0075.

[DSL16] Jianhao Du, Weihua Sheng, and Meiqin Liu. „Human-guided robot 3D mapping using virtual reality technology“. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 4624–4629. ISBN: 978-1-5090-3762-9. DOI: 10.1109/IROS.2016.7759680.

[Lee+16] Joseph Lee et al. „Visual programming for mobile robot navigation using high-level landmarks“. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 2901–2906. ISBN: 978-1-5090-3762-9. DOI: 10.1109/IROS.2016.7759449.

[The+16] Konstantinos Theofilis et al. „Panoramic view reconstruction for stereoscopic teleoperation of a humanoid robot“. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, Nov. 2016, pp. 242–248. ISBN: 978-1-5090-4718-5. DOI: 10.1109/HUMANOIDS.2016.7803284.

[XPH16] Tang Xinxing, Zhou Pengfei, and Yamada Hironao. „VR-based construction tele-robot system displayed by HMD with active viewpoint movement mode“. In: *2016 Chinese Control and Decision Conference (CCDC)*. IEEE, May 2016, pp. 6844–6850. ISBN: 978-1-4673-9714-8. DOI: 10.1109/CCDC.2016.7532231.

[MI17] Yoshiaki Mizuchi and Tetsunari Inamura. „Cloud-based multimodal human-robot interaction simulator utilizing ROS and unity frameworks“. In: *2017 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, Dec. 2017, pp. 948–955. ISBN: 978-1-5386-2263-6. DOI: 10.1109/SII.2017.8279345.

[Rol+17] Juan Roldán et al. „Multi-Robot Interfaces and Operator Situational Awareness: Study of the Impact of Immersion and Prediction“. In: *Sensors* 17.8 (July 2017), p. 1720. ISSN: 1424-8220. DOI: 10.3390/s17081720.

[The+17] Michail Theofanidis et al. „VARM“. In: *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. New York, NY, USA: ACM, June 2017, pp. 215–221. ISBN: 9781450352277. DOI: 10.1145/3056540.3056541.

[Bab+18] Edwin Babaians et al. „ROS2Unity3D; High-Performance Plugin to Interface ROS with Unity3d engine“. In: *2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*. IEEE, Dec. 2018, pp. 59–64. ISBN: 978-1-7281-2842-9. DOI: 10.1109/AIAR.2018.8769798.

[Bri+18] Filippo Brizzi et al. „Effects of Augmented Reality on the Performance of Teleoperated Industrial Assembly Tasks in a Robotic Embodiment“. In: *IEEE Transactions on Human-Machine Systems* 48.2 (Apr. 2018), pp. 197–206. ISSN: 2168-2291. DOI: 10.1109/THMS.2017.2782490.

[JOK18] Christian Just, Tobias Ortmaier, and Lueder A. Kahrs. „A user study on robot path planning inside a Virtual Reality environment“. In: *International Symposium on Robotics (ISR)*. Munich, 2018, pp. 1–6. ISBN: 978-3-8007-4699-6. URL: https://ieeexplore.ieee.org/abstract/document/8470635.

[Koh+18] Sebastian Kohn et al. „Towards a Real-Time Environment Reconstruction for VR-Based Teleoperation Through Model Segmentation“. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018, pp. 1–9. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8594053.

[KN18] Tomáš Kot and Petr Novák. „Application of virtual reality in teleoperation of the military mobile robotic system TAROS“. In: *International Journal of Advanced Robotic Systems* 15.1 (Jan. 2018), p. 172988141775154. ISSN: 1729-8814. DOI: 10.1177/1729881417751545.

[LFR18] Jeffrey I. Lipton, Aidan J. Fay, and Daniela Rus. „Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing". In: *IEEE Robotics and Automation Letters* 3.1 (Jan. 2018), pp. 179–186. ISSN: 2377-3766. DOI: 10.1109/LRA.2017.2737046.

[Whi+18] David Whitney et al. „ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018, pp. 1–9. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8593513.

[Zha+18] Jingbo Zhao et al. „The Effects of Visual and Control Latency on Piloting a Quadcopter Using a Head-Mounted Display". In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, Oct. 2018, pp. 2972–2979. ISBN: 978-1-5386-6650-0. DOI: 10.1109/SMC.2018.00505.

[AHY19] Jordan Allspaw, Lilia Heinold, and Holly A. Yanco. „Design of Virtual Reality for Humanoid Robots with Inspiration from Video Games". In: 2019, pp. 3–18. DOI: 10.1007/978-3-030-21565-1_1.

[Ast+19] Morten Andre Astad et al. „Vive for Robotics: Rapid Robot Cell Calibration". In: *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*. IEEE, Nov. 2019, pp. 151–156. ISBN: 978-1-7281-3787-2. DOI: 10.1109/ICCMA46720.2019.8988631.

[CP19] Harvey Cash and Tony J. Prescott. „Improving the Visual Comfort of Virtual Reality Telepresence for Robotics". In: 2019, pp. 697–706. DOI: 10.1007/978-3-030-35888-4_65.

[FJ19] Anton Franzluebbers and Kyle Johnson. „Remote Robotic Arm Teleoperation through Virtual Reality". In: *Symposium on Spatial User Interaction*. New York, NY, USA: ACM, Oct. 2019, pp. 1–2. ISBN: 9781450369756. DOI: 10.1145/3357251.3359444.

[Gau+19] Sanket Gaurav et al. „Deep Correspondence Learning for Effective Robotic Teleoperation using Virtual Reality". In: *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, Oct. 2019, pp. 477–483. ISBN: 978-1-5386-7630-1. DOI: 10.1109/Humanoids43949.2019.9035031.

[Gor+19] Gal Gorjup et al. „An Intuitive, Affordances Oriented Telemanipulation Framework for a Dual Robot Arm Hand System: On the Execution of Bimanual Tasks". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Nov. 2019, pp. 3611–3616. ISBN: 978-1-7281-4004-9. DOI: 10.1109/IROS40897.2019.8967782.

[Hir+19] Matthias Hirschmanner et al. „Virtual Reality Teleoperation of a Humanoid Robot Using Markerless Human Upper Body Pose Imitation“. In: *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, Oct. 2019, pp. 259–265. ISBN: 978-1-5386-7630-1. DOI: 10.1109/Humanoids43949.2019.9035064.

[Rol+19a] Juan Jesus Roldan et al. „Bringing Adaptive and Immersive Interfaces to Real-World Multi-Robot Scenarios: Application to Surveillance and Intervention in Infrastructures“. In: *IEEE Access* 7 (2019), pp. 86319–86335. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2924938.

[Rol+19b] Juan Jesús Roldán et al. „Multi-robot Systems, Virtual Reality and ROS: Developing a New Generation of Operator Interfaces“. In: 2019, pp. 29–64. DOI: 10.1007/978-3-319-91590-6_2.

[Sto+19] Patrick Stotko et al. „A VR System for Immersive Teleoperation and Live Exploration with a Mobile Robot“. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Nov. 2019, pp. 3630–3637. ISBN: 978-1-7281-4004-9. DOI: 10.1109/IROS40897.2019.8968598.

[Su+19] Y. H. Su et al. „Development of an Effective 3D VR-Based Manipulation System for Industrial Robot Manipulators“. In: *Asian Control Conference (ASCC)*. Kitakyushu: IEEE, July 2019, pp. 1–6.

[Tso+19] Ievgenii A. Tsokalo et al. „Remote Robot Control with Human-in-the-Loop over Long Distances Using Digital Twins“. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2019, pp. 1–6. ISBN: 978-1-7281-0962-6. DOI: 10.1109/GLOBECOM38437.2019.9013428.

[Van+19] David B. Van de Merwe et al. „Human-Robot Interaction During Virtual Reality Mediated Teleoperation: How Environment Information Affects Spatial Task Performance and Operator Situation Awareness“. In: 2019, pp. 163–177. DOI: 10.1007/978-3-030-21565-1_11.

[Vem+19] Anurag Sai Vempati et al. „A Virtual Reality Interface for an Autonomous Spray Painting UAV“. In: *IEEE Robotics and Automation Letters* 4.3 (July 2019), pp. 2870–2877. ISSN: 2377-3766. DOI: 10.1109/LRA.2019.2922588.

[Xi+19] Bao Xi et al. „A robotic shared control teleoperation method based on learning from demonstrations“. In: *International Journal of Advanced Robotic Systems* 16.4 (July 2019), p. 172988141985742. ISSN: 1729-8814. DOI: 10.1177/1729881419857428.

[Yas+19] Grigoriy A. Yashin et al. „AeroVr: Virtual Reality-based Teleoperation with Tactile Feedback for Aerial Manipulation“. In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, Dec. 2019, pp. 767–772. ISBN: 978-1-7281-2467-4. DOI: 10.1109/ICAR46387.2019.8981574.

[Bec+20] Israel Becerra et al. „Human Perception-Optimized Planning for Comfortable VR-Based Telepresence“. In: *IEEE Robotics and Automation Letters* 5.4 (Oct. 2020), pp. 6489–6496. ISSN: 2377-3766. DOI: 10.1109/LRA.2020.3015191.

[Elo+20] Mohamed Elobaid et al. „Telexistence and Teleoperation for Walking Humanoid Robots“. In: 2020, pp. 1106–1121. DOI: 10.1007/978-3-030-29513-4_80.

[Het+20] Rebecca Hetrick et al. „Comparing Virtual Reality Interfaces for the Teleoperation of Robots“. In: *2020 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, Apr. 2020, pp. 1–7. ISBN: 978-1-7281-7145-6. DOI: 10.1109/SIEDS49339.2020.9106630.

[KBD20] Kaveh Kamali, Ilian A. Bonev, and Christian Desrosiers. „Real-time Motion Planning for Robotic Teleoperation Using Dynamic-goal Deep Reinforcement Learning“. In: *2020 17th Conference on Computer and Robot Vision (CRV)*. IEEE, May 2020, pp. 182–189. ISBN: 978-1-7281-9891-0. DOI: 10.1109/CRV50864.2020.00032.

[Mac+20] Mateusz Maciaś et al. „Measuring Performance in Robotic Teleoperation Tasks with Virtual Reality Headgear“. In: *Advances in Intelligent Systems and Computing*. Vol. 920. Springer Verlag, 2020, pp. 408–417. DOI: 10.1007/978-3-030-13273-6_39.

[Sun+20] Da Sun et al. „A New Mixed-Reality-Based Teleoperation System for Telepresence and Maneuverability Enhancement“. In: *IEEE Transactions on Human-Machine Systems* 50.1 (Feb. 2020), pp. 55–67. ISSN: 2168-2291. DOI: 10.1109/THMS.2019.2960676.

[Wan+20] Qiyue Wang et al. „Virtual Reality Robot-Assisted Welding Based on Human Intention Recognition“. In: *IEEE Transactions on Automation Science and Engineering* 17.2 (Apr. 2020), pp. 799–808. ISSN: 1545-5955. DOI: 10.1109/TASE.2019.2945607.

[WT20] Kandai Watanabe and Masaki Takahashi. „Head-synced Drone Control for Reducing Virtual Reality Sickness“. In: *Journal of Intelligent & Robotic Systems* 97.3-4 (Mar. 2020), pp. 733–744. ISSN: 0921-0296. DOI: 10.1007/s10846-019-01054-6.

[Whi+20] David Whitney et al. „Comparing Robot Grasping Teleoperation Across Desktop and Virtual Reality with ROS Reality“. In: *International Symposium on Robotics Research*. Ed. by Nancy M. Amato et al. Puerto Varas: Springer Cham, Dec. 2020, pp. 335–350. DOI: 10.1007/978-3-030-28619-4_28.