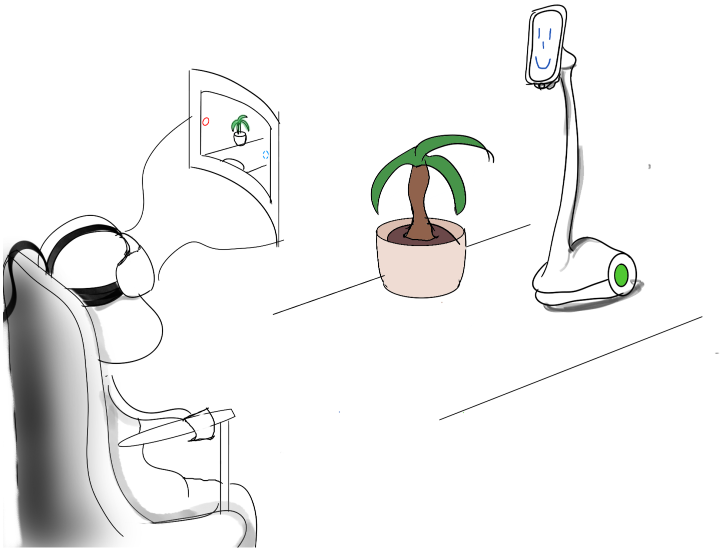
Data-driven design of a gaze controlled telerobot



Written by

Oliver Repholtz Behrens, s163866
Sebastian Hedegaard Hansen, s163870

Bachelor thesis
DTU - Technical University of Denmark
25.06.2018

Content

Acknowledgements

We would like to thank John Paulin Hansen for supervising this project. Thank you to the *GazeIt* research program [16] for providing us with the resources needed to conduct the experiment. We would also like to thank Guangtao Zhang for co-supervising this project and allowing us to be a part of his experiment.

Thanks to Diako Mardanbegi for letting us use his software and answering our questions. Finally we would like to thank DTU Skylab for letting us use their facilities for conducting the experiment and the 32 people who participated in the experiment.

Title:
Data-driven desing of a gaze controlled telepresence robot

Project:
Bachelor thesis

Project period:
Spring 2019

Written by:
Oliver Repholtz Behrens, 163866
Sebastian Hedegaard Hansen, s163870

Supervisors:
John Paulin Hansen
Guangtao Zhang

Copies:
1

Number of pages:
67

Date of submission:
25.06.2019

Sebastian Hedegaard Hansen -S163870

Oliver Repholtz Behrens - S163866

Abstract

The use of gaze to control a telepresence robot could provide motor disabled people with a new level of mobility. With a well developed robotic telepresence system that uses gaze as a control input, individuals with motor disabilities could be able to occupy jobs operating the robots in various situations. This raises the question if virtual training environments could be used to educate this whole new kind of workforce. This report investigates if simulated environments potentially could be used to replace training in reality when educating robot operates. In order to evaluate operators performance, alertness and awareness new measures needs to be developed. A primary focus in this report will therefore be to investigate what information that can be extracted from implementing a pro-saccade test in the same HMD that the operator uses to control the robot.

To investigate these aspects a between-subjects experiment was conducted with 32 participants operating a gaze controlled telepresence robot. The experiment showed that the operators task completion time was independent of the environment they had been trained in, only the number of collisions was higher for the test subjects trained in VR.

The results of the pro-saccade test remained constant between trials for each test subject, and thus no learning effect was seen. But the results of the pro-saccade test indicated that the best robot performers had shorter latencies on their saccades indicating higher alertness and was experiencing lower workload than the worst robot performers. The experiment showed interesting results but further research into the correlation between eye movements and performance, alertness and awareness needs to be done. Based on the finding from the experiment 5 possible concepts was developed, 3 concepts that includes the use of a dynamic saccade test, 1 concept of a VR training environments and 1 concepts of a interface that uses visual stimuli to prim users to make saccades. With further testing, these could be implemented in the robots operating system to enhance the user experience of driving a gaze controlled telepresence robot.

1 Introduction

It is becoming increasingly normal to see a robot driving down the hallway at workplaces or conferences. But the use of these telepresence robots has applications beyond this use. Telepresence robots can be used by people with motor disabilities to navigate in environments that was not accessible before. For people with motor disabilities from the neck down, such as people suffering from ALS, the use of telepresence robots could create a new way of self-embodiment in a machine. Before this is possible new ways of controlling a telepresence robot needs to be developed to accommodate very limited physical movement. Here the use of gaze to control a telepresence robot could provide this possibility.

This project is conducted as a part of a ongoing research program at the Technical University of Denmark's management department, called GazeIt [16]. The research group works on how gaze technology can be used as an interaction method to improve the everyday lives of people with motor disabilities. Most recently the research of gaze controlled telepresence robots was initiated.

There are a significant number of people in Denmark who are unable to work due to a motor disability. These individuals cost the society a lot of money in both medical expenses and in lost work hours. If these people could be included in the general workforce it would not only benefit the society, but also benefit them self and possibly increase their quality of life. The jobs they can occupy is limited at the moment. Therefore a new type of jobs needs to be created. These include the use of telepresence robots and gaze interaction. The overall motivation of the research program is to create a remote workforce consisting of motor disabled persons in wheelchairs and ALS-patients, with jobs as telepresence robot operators using gaze as a control input. OryLab [3] which is a project partner recently opened a pop-up café where all the waiters/waitresses was motor disabled controlling telepresence robots and some controlled the robots using gaze. This leads to the question about training and general adaption of the new technology. Is there a need to develop specialized training systems, and how should they be designed? Is it enough to train in a simulated Virtual Reality environment or do you need to experience real world scenarios before entering a workplace?

In this project a experiment was conducted as a part of Guangtao Zhangs Ph.d. project [16]. The experiment was conducted in DTU Skylabs facilities in Lyngby where 32 participants was recruited. The general purpose with the experiment was to investigate the possibility to train individuals in controlling gaze controlled telepresence robots in a simulated environment. However, the general focus in this report will also be on the investigation of how a pro-saccade test might be used to assess something about an individual's mental state or ability to drive a robot using their eyes, by estimating peoples perceived workload, level of situational Awareness (SA) and alertness. This is done because situational awareness, alertness and workload is measured using very invasive tests where user is asked a series of question both during and after a task (SPAM and NASA-TLX). If it is possible to determine a user's mental state with a simple saccade test the need for these invasive tests could potentially be eliminated. Furthermore, ALS patients does not have the possibility to clearly communicate how they are feeling the use of a pro-saccade test to determine their mental state would be valuable information for the health care personal assisting the patients.

The first part of this report will present an empirical study into previous research conducted within the fields relevant for gaze controlled telepresence robots. These research fields consists of telepresence robots, gaze control, general eye-movements, situational awareness and presence. It is then presented how a pro-saccade test was created in a VR-environment and implemented in the experiment. The experiment method, design and setup is explained, and the results are analyzed and discussed. Based on the prior work, results and discussion of the experiment, 5 concepts is presented that based on the data-analysis could be relevant to implement and test in a gaze controlled telepresence robot system.

This leads to the research question for this project:

1.1 Research question

How does training environments affect operator performance, awareness and alertness in operating a gaze controlled telepresence robot? Could operator performance, awareness and alertness be evaluated without intervening in the operators primary task - using a saccade test?

How could a dynamical test be incorporated in the design of a new interface for a gaze controlled telepresence robot?

2 Prior research

This section presents the relevant research within the fields of this project. The sections is meant as an inspiration to our research question and as a foundation for the discussion of the results from the experiment. Gaze controlled telepresence robots has a great potential to change the lives of people with motor disabilities. The research (empirical material) on this topic is at the time of this project limited. However, gaze controlled telepresence robots is a technology combined by two research fields; telepresence robots and gaze control. Thus, the section presents and highlights the results and challenges of previous studies, within the fields of gaze controlled telepresence robots and the research fields closely related to it. How to evaluate the performance in teleoperation is also examined with presence, situational awareness(SA) and their correlation as the main topics of interest. Literature about different eye-movements and how they have been used as a measure of different cognitive aspects of a person. The results of the presented studies is all relevant for the study and development a of gaze controlled telepresence robot systems. The literature/empirical-material can be seen as a review of what research that already has been conducted regarding gaze controlled telepresence robots.

2.1 Telepresence robots

This subsection presents literature about the different use cases of telepresence robots, the challenges associated with the use of the robots and the proposed solutions to these challenges. The literature presented in this section primarily focuses on the use cases where telepresence robots is used in office environments or by people with motor disabilities.

2.1.1 What is it?

New ways of being *present* emerges with technological improvements, where interactions is not bound by your physical location. The newest element in this development is the use of robotic telepresence, that combines regular telepresence (video conferences etc.) with being physical embodied in a robot at a remote location. This allows the operator to navigate in the physical space that they connect to [23]. This has a number of practical and social applications, that could impact how people work and interact with each other. These include the use of telepresence robots in healthcare, elderly care, long distance relationships, office environments or in environments that can be hostile for humans (nuclear power plants etc.) [48].

Telepresence robots can have many different designs but the basic principles are the same [23]. A person from a remote location connects to the robot through the internet. The operator sees a live video stream from the robots perspective with both picture and sound. The robot is controlled by the operator using one of various control inputs (keyboard, joystick etc.), see figure 1.

2.1.2 Telepresence robots in a office environment

The use of telepresence robots in a office environment are benefiting both the employers and the employees. A company could save a lot of money by not having to transport your workers great distances for a meeting, both in travel expenses and lost work. A relatively expensive robot as

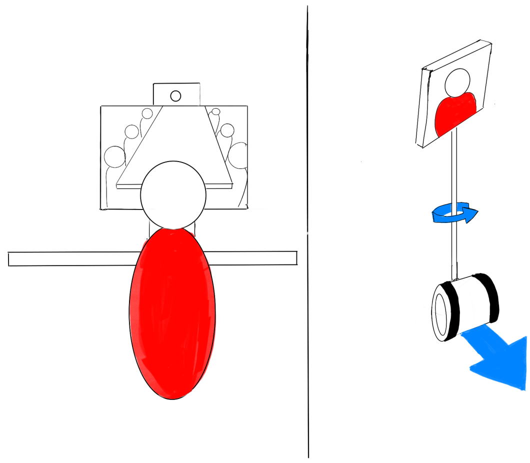


Figure 1: Left: The robot operator attends a meeting from home. Right: The operator is embodied in the robot at a remote environment

the *double 2* [39] can therefore quickly earn in its value in gained work hours and reduced travel expenses. The distant worker operating the telepresence robot has the freedom to work from home, but still has the ability to interact with colleagues in a physical space [23].

The effect of using telepresence robots in office meetings have been discussed with no real definitive conclusion. Tsui et al. [48] describes the difference in opinions, regarding the use of telepresence robots in office meetings, between 6 groups of professionals. The groups that were negative towards the use of the robot, experienced technical issues and felt the telepresence robot was unnecessary in a static office meeting that requires little movement. In one group, the operator felt more included in the meeting compared to participating in a regular conference call.

Another use case for telepresence robots is in the more informal situations that requires a physical aspect, like the casual hallway talk or the walk from one meeting to another [23]. After a meeting you often have some final details you need to discuss with individual colleagues. The use of telepresence robots could allow colleagues to exit a meeting together and continue the conversation. Kristoffersson et al. [23] describes a company in the San Francisco area, that used telepresence robots in a longer period of time. The robot was located at the operators usual desk even though no meeting was going on. This gave the employees on site the ability of dropping by to communicate in a much more casual fashion.

2.1.3 Telepresence for people with motor disabilities

For people with physical disabilities the telepresence robots serves as a way to move around in physical spaces they would not usually be able to access. This could provide a feeling of freedom that is valuable, if the operator usually is bound to a wheelchair or bed.

Neustaedter et al. [36] described the use telepresence robots at a conference. A number of participant suffered from some sort of accessibility challenge. It was expressed that the experience of moving around with a robot did not differ that much from moving around in a wheelchair. This had the added benefit that the operator could relax in their own private home. Which can be important for individuals with motor disabilities that relies on the use of various equipment etc. for normal daily tasks (going to the bathroom, eating etc.).

2.1.4 Social norms and ethical issues

With new technology there is a learning curve for the people interacting with the technology. This leads to a number of situation where the norms of social interaction needs to be adapted or redefined. After the initial fascination phase, a number of issues occurs [36]. One issue raised was based on the telepresence robots mobility, as the robots have limited degrees of freedom. A normal minor action (like asking someone sitting behind you, if you are blocking their view) could be perceived as a bigger action. Another example is the robots lack of body language, when approaching a person to engage in conversation, the approached had a difficult time figuring out if the robot actually wanted to talk or just go past [36].

When communicating it is important to be able to get a sense of the other persons presence and attention. This is not a problem in stationary situations, as you are face to face, but when a person is walking with a robot some social awkward situations occurs. Tsui et al. [48] observed and registered walking positions of people walking in a hallway and communicating with a telepresence robot [48]. The majority of the participants was walking in unnatural ways, with the extreme cases of walking backwards. Additionally Steinfeld et al. [43] showed that the cognitive load experience by the operator increased by the extra task of controlling the robot, while talking to an colleague. The operators attention was more on the driving task, than the actual conversation. This made it difficult to have a natural conversation and even more difficult to have a technical conversation [48].

Neustaedter et al. [36] discussed the social importance of eating and drinking together. Distance workers was not able to join the coffee or lunch break in a natural way, because of the physical restrictions but also because of the possible time difference.

In general the acceptance of new technology greatly vary depending under what circumstances it is implemented. This is also the case with the use of telepresence robots. Kristoffersen et al. [23] proposed some concerns regarding the use of telepresence, in the healthcare sector. The concerns was about loss of human interaction, the replacement of professionals and staff which could result in reduced healthcare. These concerns needs to be addressed when designing and implementing new technology such as telepresence robots.

2.1.5 Problems and current solutions

Besides the challenges regarding social behavior and ethical concerns, there are a number of practical and technical problems that reduces the user experience in operating a telepresence robot. A recurrent subject in previous literature is the technical problems. Robot operators experienced the video cutting out, lack of audio and other network connection issues, due to component quality and unstable WIFI-connections [48]. In some cases it was experienced that if the battery ran out or the robot lost the internet connection, the robot would return to the docking station, which is very disturbing in a meeting situation.

Desai et al. [7] suggested two different types of video profiles depending on the situation, static and dynamic. Due to the shortcomings in the wireless network connection, it is impossible to get perfect quality in every aspect of live video streaming (contrast, latency, resolution, color depth, pauses in video etc.). Therefore Desai et al. [7] suggest that this trade-off should be considered when choosing the video quality depending on the use case of the robot. When moving (dynamic mode) aspects such as latency, fewer pauses in video and scale perception is valued more than resolution, color depth etc. On the other hand when standing still (static mode) aspects such as color depth/contrast and resolution was more valued.

Another recurrent issue by Neustadter et al. in [36] is the lack of spatial awareness and difficulties navigating in unfamiliar settings. The operator is restricted to the cameras line of sight, most commonly displayed on a two-dimensional screen, where depth perception is hard to interpret. This contributes to operators often having a hard time determining where they are located in a building. The use of maps was suggested, but with the current lack of live positioning technologies it was argued as useless. By providing more information to the operator through sensors, adds to the cognitive load as well as putting more pressure on the network. These aspects refers to a persons situational awareness based on their *perception*, *comprehension* and *projection* of their surroundings. The 3 aspects is essential for how a persons situational awareness is measured.

A technical solution that have been suggested in multiple articles [48, 23] is the implementation of autonomous functions. By letting the robot decide a desired path, the operator can keep their attention on having a conversation or even working. This could also increase the safety of robot operation, as obstacles could be detected locally and avoided.

This subsection showed that telepresence robots have many practical applications and can greatly enhance human interaction compared to regular telepresence systems. Companies have benefited from using telepresence robots to reduce the cost of having distance workers and people with motor disabilities have used telepresence robots to move freely around while being at home.

A number of challenges still needs to be considered in future designs of telepresence robots. The task of driving a telepresence robot can be cognitively demanding and might lead to possible awkward social interactions for both the robot operator and the people interacting with the robot. This issue originates from the difficulty in operating the robot while talking and the lack of spatial awareness while being embodied in a telepresence robot.

Additionally there are a number of technical issues that reduce the user experience of operating a telepresence robot. These problems are regarding quality and lag in video and audio. Until the technology catches up, a future design of a robotic telepresence system should focus on what use situations the robot will be used in.

2.2 Eye tracking - Gaze

This subsection presents how ones point of gaze has been used in previous studies to track a persons eye-movements, as an interaction input and to control different vehicles. The section also presents the problems that are associated with using gaze an input method.

2.2.1 What is gaze technology

Eye tracking using gaze technology is the measures of eye movements to obtain information about where one is looking(point of gaze) [50]. The technology has a huge potential in a wide range of applications, especially as an interaction tool for people with motor disabilities, who cannot interact with a normal interface when operating everyday items, and some might need assistance for speaking. Gaze technology could potentially be a life changing interaction method for these people, but it has many potential and already used applications beyond this.

2.2.2 Applications of gaze technology

Eye tracking technology can be used to monitor where the eye is looking, and have a wide range of applications in different fields of work. Eye tracking is used in psychology and marketing to map where and what people are looking at, when they are exposed to something new or perform tasks wearing an eye-tracker. Knowing where the user is looking in given situations can provide a lot of different information. This can be useful when researchers and developers wants to see and assess how users react when presented with a new interface. A study was conducted by Bækgaard et al. [1] on how childrens pupils reacted, when exposed to two different LEGO assembly manuals, one booklet and one digital. The study suggest that it is possible to collect and use pupillary measurements to assess aspects such as task effort and difficulties when exposed to a new interface. Eye tracking can also be used in assessing how professionals use their eyes compared to normal users. For instance eye-tracking was used in the E-LEAGUE to track where professional *Counter Strike: Global Offensive(CS:GO)* players were looking during a game [47]. The results showed that professional players spend much more time looking at the game's mini-map (which shows an overview of the environment and activities nearby) than normal CS:GO players do.

Gaze can also be used as a hands-free input, leaving the hands free to do other stuff. This opens up for a different variety of use cases. In a previous study by Hansen et al. [18] gaze has been used as an input for children to interact with a LEGO building manual on a tablet. The children could turn, zoom and change pages using their eyes and head-movements, instead of using a traditional booklet. The study showed that most of the tested children could use the gaze-interactive building manual without big problems.

A study conducted by Minakata et al. [31] examined how gaze as an interaction method performed compared to a traditional input method as a mouse, and other hands-free inputs such as head movements and a foot mouse. They measured and compared the throughput and speed of a pointing task. The study showed that mouse and head movement almost had equal performance on activation speed. Gaze was significantly slower and had a lower throughput than both mouse and head movement. Why would gaze then be used if the throughput and activation speed is less than other input methods? Not everyone has the same control over their body, some is in a wheelchair and need some help doing everyday things, some other have no physical control over their body, such as ALS-patients. Gaze serves as one of their only possible physical control inputs. Thus, gaze as a control-input makes it possible for people with motor disabilities to interact with their environment given the right equipment.

Gaze is already being used as an interaction tool for people to communicate and interact with their surroundings. People who cannot use their hands, can interact with other people by gazing at a keyboard on a laptop or another device as a text entry port [28]. Hands-free text entry enables disabled people to interact with their surroundings, and even to speak out loud by reading the typed text aloud. But communicating using gaze through text entry is really time consuming. Thus, there are a lot of different methods to speed up the text entry. The most common methods are different keyboards and interfaces which includes a predicted word list, character prediction etc. such as GazeTalk [28].

Gaze for communication can potentially serve as an alternative to expensive communication systems, such as the one Stephen Hawking used. Recently two bachelor students from the Technical University of Denmark developed an eye-tracking application for smartphones, that could interact with the users smart home or other devices connected to the internet [44]. This served as a nice and much cheaper alternative than the current eye-tracking technologies that previously was created.

2.2.3 Remote control using gaze

In previous studies gaze has been used as a hands-free input to remote control vehicles such as drones and robots. **Drones** are already remote controlled and often uses first person view glasses as vision, and a normal joystick controller as a control input. A study was conducted by Hansen et al. [17] on how experienced computer gamers was able to fly and maneuver a drone using gaze control as an input method. The experiment tested different control mappings, to see which of these that would be easiest for the operators to control the drone, without much training in the environment. The control of the drones had four degrees of freedom (different control inputs) which was *Speed, Rotation, Translation and Altitude control*, two which would be controlled by gaze and two controlled by arrow-keys. The experiment showed that the mapping of the controls had a impact on the operators performance and experience in controlling the drone. The main findings of the study was that gamers could control the drone using any of the suggested interfaces. The fourth control-mode (*Rotation* and *speed* by gaze; *translation* and *altitude* by arrow-keys) was rated as the most reliable and a bit easier than the other control-modes. They were all rated as being equally fun to control. The fourth control-mode resembled normal gaming environments the most, where the mouse is used to turn the viewpoint - for which gaze served as a natural replacement [17]. Thus, creating a control mode that resembled a familiar environment enhanced the usability and created a less steep learning curve.

Tall et al. [45] conducted a study on how gaze could be used to remote control a **LEGO Robot**. The aim of the study was to examine if a hands-free input could be used to drive a robot, and use the results to examine if this could be a safe control-mode for a motor disabled person to control their own wheelchair. The experiment was conducted using five different control inputs. The problems faced in the study was regarding image quality and lag, resulting in bad results and problems when remote controlling the vehicle. The overall results of remote controlling a car with gaze as a control input, showed that high-price gaze devices had similar performance to buttons and mouse as control inputs with lap-time as a performance parameter, but when using a cheaper eye-tracking setup such as a webcam the results decreased significantly. The study shows and suggest that high-performance eye-trackers and the test setup could be used to test gaze control on other types of vehicles. Thus, gaze has the potential, as a control mode, to allow hands-free control of a wheelchair for the operator, which is a crucial need for people who have limited or no physical control over their hands, such as most ALS-patients.

Wheelchair: ALS-patients needs an interaction input that allows them to control their own movements and get a feeling of independence. Patients with motor disabilities could here benefit from gaze control to drive their own wheelchairs and to communicate with others. This has been explored by Eid et al. [10], who created a gaze controlled wheelchair for a man having ALS. The patient no longer had the physical strength to move his hands, and thus he could no longer operate his previous wheelchair. He had tried operating a wheelchair using gaze control before, but the interface was mentally overloading and really difficult to control. The study presented a gaze control interface, which allowed the user to regain better control over his wheelchair by applying some level of autonomy to the operator-interface and creating an algorithm for much faster calibration. This allowed the user to drive his wheelchair in unknown and ever changing environments, by letting the user pick specific positions he wants to reach on a temporary map. The study showed, that it was possible for the ALS-patient to operate his wheelchair using their interface. A limitation of the study was that only one user was involved in the testing and development of the gaze system.

This subsection showed that eye-tracking can be used to measure and evaluate were people are focusing and how much attention they are paying. It further shows that gaze can be used as an interaction method. For people with motor disabilities gaze could serve as a good alternative to traditional input methods in text entry and as a control input to their wheelchair, if some level of autonomy was included. The section presented results that showed how gaze could serve as a control input in remote vehicle control. The control mapping was more successful when it resembled mappings that the users was familiar with from other control systems. When remote controlling vehicles using gaze the problems was often hardware related.

2.3 Gaze controlled telepresence robots

This subsection presents the combination of telepresence robots and the use of gaze technology as a control input, a gaze controlled telepresence robots. The subsection presents the research that has been conducted in this field, what the possibilities and challenges there is in gaze operating a robot from a remote location. It will be described how motor disabled can benefit from a combination of these two technologies.

Traditional wheelchairs for people with physical disabilities are usually controlled using a joystick as input method. Even though this does not require high levels of physical effort, it still excludes a number of people who does not have the ability to move a muscle from the neck down. This issue was raised by Eid et al. in [10], where a system consisting of an electric wheelchair and a pair of eye tracking glasses, was designed for and tested with an ALS-patient. The study proved that it was possible to control a wheelchair using gaze for an ALS patient. It was reported by a participant in the study conducted by Neustaedter et al. [36] that the use of a telepresence robot carries many similarities with the use of a wheelchair. The findings by Eid et al. [10] is therefore interesting to consider in the future design of gaze-controlled telepresence robots.

A gaze controlled telepresence robot is simply put a telepresence robot controlled by gaze as an input method. The eye position is measured using an eye-tracker, and the gaze input is transferred to the robot trough the internet. Zhang et al. [54] argues how a gaze controlled telepresence robot could generate value for people with motor disabilities. They describe a robotic telepresence system that uses gaze as a control input (this is the same robotic telepresence system used in this project). In this setup the robot operator will have to wear a head mounted display (HMD) with build in eye-trackers to control the robot. In another study Zhang et al. [53] tested the performance of the gaze controlled telepresence system on 16 test subjects. The study compared two input methods (eye gaze and joystick). The study showed that it is possible to control a telepresence robot using gaze as a control input. But the mean performance of the participants was lower for gaze control compared to joystick control - measured on task completion time and deviation from optimal path. In addition to this, each participant responded to a NASA-TLX to assess their perceived workload when performing the task. This showed a significant mean difference between input methods, with gaze control reported as more demanding. Driving the robot using a joystick also resulted in a higher level of situational awareness compared to driving the robot using gaze. The study used 6 measures on a 7-point scale to assess presence, and only showed a difference in one aspect of presence namely the *possibility to exam* between control methods.

Gaze controlled telepresence robots recently gained a lot of attention, since the Japanese firm **OryLab** created a temporary pop-up cafe, where the customers was only served by telepresence robots operated by persons with disabilities [3]. The disabilities of the cafe's servants vary in severity. Thus, the control of the robots vary, but for the ALS-servenats they used gaze to operate the robot, and to speak with the customer visiting the cafe. An example of the OriHime-D robot serving a customer and the gaze control interface, as OryLab presents it, can be seen in figure 2.

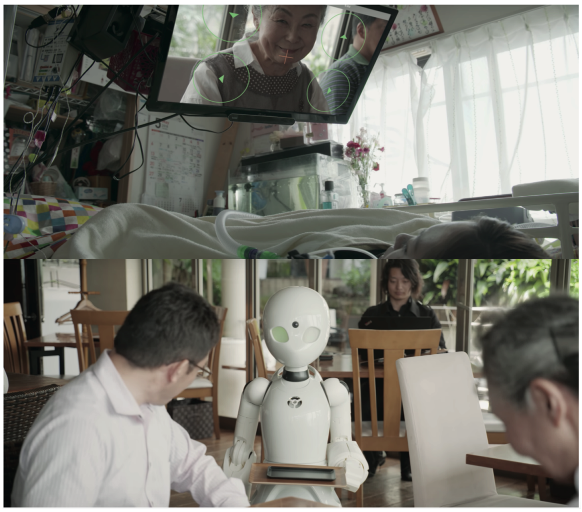


Figure 2: ALS-patient serving customer at OryLab's pop-up cafe. Top: ALS-patient at home sees a live video feed from the cafe. Bottom: The robot waitress at the cafe

OryLab provides a great example of how gaze controlled telepresence robots could improve the life quality of motor disabled persons, by providing jobs and making them feel useful. A great variety of jobs could in theory be undertaken by this group of people operating the telepresence robots.

This subsection showed that the use of gaze to control a telepresence robot is possible. The experience of presence while operating the telepresence robot using gaze was almost the same as when operating using a joystick. There still is a challenge in the level of situatonal awareness the user experience while driving the robot using gaze.

2.4 Saccades and other eye movements

This subsection is written as an empirical foundation for the projects experiment and the results is meant to be compared with the finding in this section. Different types of eye-movements will be presented, with a focus on saccades and how they are measured using a saccade test. The section will investigate how saccades and other eye movements have been used and analyzed in literature as a psychophysiological measures of alertness, awareness and mental state. Furthermore the use of saccades in the design of interfaces in gaze interaction is described.

2.4.1 General description of saccades and other eye movements

Humans and other animals use their eyes to scan their surroundings, recognize a face or catch a ball. For this to be possible a series of eye movements are used. The different eye movement are described by mulvey in [34] and are listed here: *fixation, saccade, microsaccade, smooth pursuit, vergence, Vestibular reflex, accommodation* and *pupil dilation*, and can be seen in table 1 for a short description.

Eye movement	**Description**
Fixation	$>$100 ms within 1 degree of vision.
Saccade	Movement to new areas of the visual field greater than 2 degrees.
Microsaccade	Very small movements which occur irregularly during a saccade.
Smooth pursuit	Movement of the eye to follow a moving target with the same velocity and trajectory as the target.
Vergence	Movement of the eyes inward, i.e. in opposite directions, to offset retinal disparity for close objects.
Vestibular ocular reflex (VOR)	Movements to maintain the point of regard during head movements.
Optokinetic reflex	Saccade-smooth pursuit movements to focus on moving scenes.
Accommodation	Changes in the shape of the lens to focus light from objects at varying distances.
Pupil dilation	Changes in the size of the aperture in the iris in order to maintain optimal light levels inside the eye.

Table 1: Descriptions of the different eye movements, from Mulvey [34]

Saccades are fast movements of the fovea from one fixation point to another. The speed of a saccade can vary, but is characterized as one of the fastest movements a human can perform along with blinking. While the saccade is happening the sensitivity of the eyes drops to nearly nothing, meaning that you are blinded while the saccade occurs. The brain is able to stitch together an image of what you are seeing between two fixation points [34]. The saccadic eye movements is therefore used to quickly scan the most important features of the scene in front of you. A saccade is a ballistic movement, meaning that the direction of the movement is predetermined before the saccade begins. Influence on the saccade therefore needs to be done 70ms before the saccade, else it will not have an effect on the actual saccade [14].

2.4.2 Saccade tests

The study of a person's saccadic eye movements have been widely used in psychology, to give assessments of cognitive and inhibitory processes [29]. In addition the use of saccade test can be used to diagnose a series of diseases [46].

Types of saccade tests: There are two main types of saccades when doing a saccade test: Pro-saccades and anti-saccades. The pro-saccade test is designed to determine an individuals ability to respond to an appearing target, see figure 3. The test subject is asked to focus on a fixation point and then look towards the appearing *target* as quickly as possible, measuring if they made a correct saccade towards the target, their reaction time, amplitude of their saccade etc.

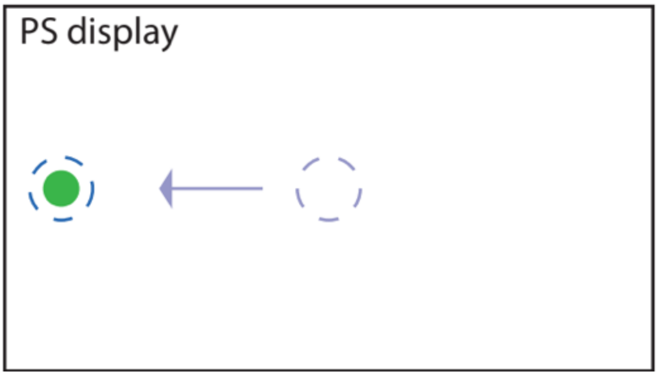


Figure 3: Pro-saccade test: The user fixates in the center, and the *target* appears on the left

In an anti-saccade test the test subject is asked to focus on a fixation point and when a *distractor* appears, look in the opposite direction, see figure 4. The anti-saccade test measure the test subjects ability to restrain the urge to look at the *distractor* and instead look away [29]. The two test differ in what is demanded from the test subject. In the anti-saccade test the subject needs to suppress the immediately urge to perform a pro-saccade towards the target and instead look away in the mirror direction. A pro-saccade on the other hand is influenced by a series of other cognitive processes [20]. A pro-saccade is highly dependent on what part of the visual scenes that are most relevant for the goals and intentions of the observer. Therefore different conclusion about the operators alertness and awareness can be drawn from both tests.

Use of saccade test previously: An anti-saccade test has been used to assess if elderly could drive a car by Schmitt et al. [41]. The study showed that there is a strong link between the

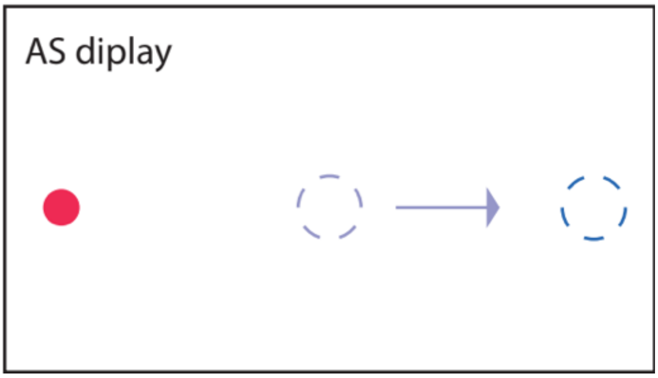


Figure 4: Anti-saccade test: The user fixates in the center, and the *distractor* appears on the left.

anti-saccade test results (correct eye-movements) and driving behaviour. Thus, the anti-saccade test might very well be an indicator of the inability to drive. Another study conducted by Munoz, et al. [35] found that children and elderly had slower reaction time on both pro - and anti saccade tests, than young adults (20-30 years old), and that children was most prone to make direction errors.

2.4.3 Saccades in design - Gesture based design

Saccade based gestures has been discussed as an alternative to other gaze inputs. Instead of using dwell time as an interaction input, where inputs to an interface is registered when the user have gazed at an area in a certain amount of time, the use of gestures could be used.

An example of this is seen in the article *Single gaze gestures* by Mollenbach et al. [33], were it was proposed to use anti-saccades to interact with a computer. An example of this could be the user had to look in the opposite direction of some kind of stimuli to give an input. This makes it easier to distinguish between an actual command from the user and a natural inspection of the screen or their surroundings.

Saccade based gestures has also been seen in text entry by gaze (EyeWrite), where the user inputs the individual letters by a series of saccades [28]. The interface is here a grid consisting of 5 fields (4 in the corners 1 in the middle). See figure 5 for an example of the letter *T*.

2.4.4 Eye movements use to determine mental states

Different eye movements have been examined in previous literature, to address the correlation with peoples perceived workload, sense of SA and presence. Saccades and other eye movements have been used and analyzed in literature as a psychophysiological measures of alertness, awareness and mental state.

Amplitude of a saccade: Depending on the situation and task, the amplitude of saccadic eye movement varies. In high complexity driving situations Cardona [2] found the number of large amplitude saccades increased. In a study by Majaranta [27] test participant had to count the number of tones in a sequence. With the increase of complexity(cognitive workload) the amplitude of saccades decreased. This task was based on the hearing sense.

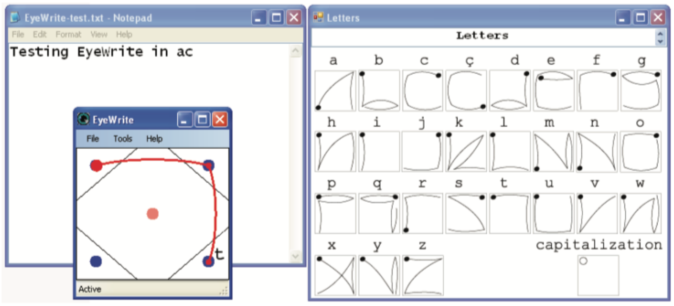


Figure 5: Picture of a saccade gesture based text-entry system EyeWrite and its alphabet, as shown in [28].

Latency of a saccade: The latency of a saccade can intuitively be a measure of how fast a individual are at reacting to a suddenly appearing stimuli. Thus, latency of a saccade could be a measure for how alert a person is.

Summer discuss the possible reasons for high and low latency when performing a saccade in [25] chapter 22. He argues that the system that initiates the saccade is very prioritized in how the saccade is performed, both in the movement and speed of the saccade, meaning that it is not only about the target but more about the *relevance* of the target in connection to the general visual field. Individuals therefore wont perform the fastest saccade possible, as the processing of what going on in their line of sight slows the saccade down. When a stimuli is presented the first thing that is processed is therefore not what sparks most attention(bright color, size, etc.), but what is most important for the observer. This is backed up by Hutton [20] that formulates it as an evaluation happening: *"saccadic eye movements have such long latencies because we need to work out not just where to look, but whether it is worth our looking there at all (given all the possible places we could be looking)"*.

Another study has shown that the latency of a saccade is independent of the target of interests nature. It does not matter if the target is stationary or moving nor does the angular displacement of the stimuli influence the latency of a saccade [6].

Blinks: Wilson [13] describes the psychophysiological assessment of a persons level of SA, where blink duration and frequency was investigated. Wilson describes the study conducted by Vidulich et al. [49] where different information displayed for a pilot in a combat mission is designed to conduct higher or lower levels of SA. The study concluded that the conditions with the lowest level of SA causes short blink duration and a higher frequency of blinks. The connection between workload and SA (which is also explained in the SA section) was used to explain this phenomenon, but is was also stated that the two principles is different but highly dependent. These findings are supported by Damos [5] that explains the decrease in blink duration as a way to prolong the fixation period in high workload tasks. Another example is Cardona's [2] investigation of the influence of spontaneous blinks and saccade-blink-pairs in driving situations with varying complexity levels. The study did not find any correlation between the complexity level and the number of spontaneous blinks, but they found a significant increase in pairs of saccades and blinks with higher complexity levels.

Pupil size: The size of the pupil is highly dependent on the neurological activity of an individual [19]. It has been concluded that the difficulty of a problem is directly correlated with the expansion of the pupil. In an experiment by Hess and Polt [19] test subjects was asked a mathematical problem ranging in difficulty (7x8 up to 16x23) where their pupils were measured. The study found that the percentage change of the diameter increases significantly with the difficulty of the questions. This was the same results shown by Bækgaard et al. [1] where the pupil dilation of children was investigated during a LEGO assembly task.

The subsection explained what a saccade is, how a saccade test works and what parameters that are measured during the test. The previous use of saccade tests showed that results of a saccade test is age related and that a saccade-test has been used to assess the inability if elderly to drive. The use of saccades in design have been presented where saccade gestures was used to interact with an interface.

Findings in the literature regarding the meaning of eye movements in different situations was also presented. The main findings was that an increase in workload caused higher blink frequency, shorter blink duration, pupil dilation, increase in blink saccade pairs and a higher number of large amplitude saccades. The latency of a saccade in a saccade test can be seen as a measure of the alertness (reaction time) of the test subject. In previous studies it was shown that saccade latency's was independent of the nature of the target of interest. Thus, it seems reasonable to assume saccade latency's can be used as a objective measure of alertness.

2.5 Presence

This subsection presents what presence in virtual reality and teleoperation is and why it is an important parameter to think off when designing telepresence systems. The subsection also shows how presence can serve as a measure of user-experience and how sickness correlates with presence in previous studies.

Presence is an important factor when entering another reality than your own. This could be a factor of feeling present in a virtual reality fictive game environment, or it could be something as the feeling of being present in a remote location (Remote-Reality) which is the case when operating a telepresence robot. Presence has many definitions one of them is the International Society for Presence Research(ISPR) that defines presence as:

"Presence (a shortened version of the term "telepresence") is a psychological state or subjective perception in which even though part or all of an individual's current experience is generated by and/or filtered through human-made technology, part or all of the individual's perception fails to accurately acknowledge the role of the technology in the experience" - ISPR [15].

Said in other words, presence is the feeling of being present in some reality that you are not physically situated in. Thus, presence is a very important factor for the designers of both telepresences robots and Virtual-Reality environments. When Minsky [32] first talked about telepresence in 1980 and how we needed a safe and reliable mechanical system for obtaining real telepresence in a remote environment, he probably did not expect this would be the same challenges faced when creating a sense of presence in a virtual environment. Jerald who also uses the ISPR's definition of presence, describes the illusion of presence in Virtual Reality as consisting of four components in his book [21] chapter 4, namely:

- **The illusion of being in a spatial place**: Occurs when the users sensory modalities is equal to the stimuli from the environment. This state of illusion can be achieved with low latency, high frame rate and good calibration of the device.
- **The illusion of self-embodiment**: The perception of having a body when looking around in the virtual environment. The body does not have to resemble our own, could be a robot body.
- **The illusion of physical interaction**: If the user tries to touch something a physical, feedback should be provided, such as colliding with some part of the scene.
- **The illusion of social communication**: Social presence is the possibility to communicate verbally and use gestures when interacting with other people in the environment.

A good example of how presence have been used as a psychological variable is a study by Regenbrecht et al. [37], where they examined the correlation between presence and the fear of heights in virtual environments. The study showed that the more present people feel the more danger they feel.

Breaks in presence (BIP) can occur when the user finds him/her self being present in their own reality wearing a HMD. This can be caused by external factors such as noise around the person or the HMD being uncomfortable to wear. But it also happens when the user has to remove the headset to fill out a survey, the need for reorientation in the real world causes breaks in presence [42]. Breaks in presence reduces the feeling of being somewhere else and should at all causes be avoided as much as possible.

Measuring presence: Several ways to measure presence have been used and proposed mainly as post-trial surveys. This is often a problem when you want to measure presence while the test subjects wears a HMD. The problems of the test subject having to leave the virtual environment is the longer duration of the experiment, possible disorientation and the break in presence(BIP). Thus, Schwind et al. [42] recently conducted a study to see if it was possible to create a questionnaire within the virtual environment. The study compared two groups one that did the presence surveys in the virtual environment and the other did it as a post-trial questionnaire. The study showed that completing the questionnaire in the virtual environment did not change the measured presence, but it did increase the consistency of the variance. Riley et al. [38] also addresses the problem in assessing presence stating that there is a lack of valid and reliable objective measures of presence and that there is no agreement of the best way to measure presence. They end the discussion by stating the only consensus is that a reliable, repeated and robust measure of presence is needed.

Is presence related to cybersickness?

Witmer and Singer [52] showed that there seems to be a correlation between presence and simulator sickness. The study showed that test subjects that felt less presence reported more symptoms of simulator sickness. The study suggested that the symptoms of simulator sickness (headache and nausea) removes focus from the presence in the environment, which seems very reasonable. More recently Jerald [21] also addressed the correlation in his book about virtual reality (chapter 17.4). He addresses it as a part of the design guidelines for VR-environments: Unrealistic physics in an environment can help reduce motion-sickness but tends to reduce the feeling of presence as well. Things that enhance the felling of presence such as a wide view, can provoke cybersickness. Thus, it's suggest that there is a tradeoff between possible motion sickness and the sense of presence that needs to be incorporated in the design of virtual environments.

The subsection presented the concept of presence and what aspects that can enhance the feeling of presence in another environment for the user. The image quality of the simulated/remote environment greatly influence the level of experienced presence and breaks in presence should be avoided if possible. There seems to be a correlation between presence and motion-sickness, but the results vary between the presented studies.

This shows and suggests it could be feasible to create a dynamic test for assessing parameters such as presence within the virtual environment, to asses the state of the operator. This could be done both before and/or during the task.

2.6 Situational awareness(SA)

This subsection introduces the different aspects of situational awareness and presents how SA can be measured as found in the literature. The subsection also addresses why it is important to have a high level of situational awareness when interacting with complex systems and how a higher level of SA is obtained.

2.6.1 Introduction to situational awareness

When operating complex computer systems the user is exposed to a lot of information. All this information needs to be processed to make decisions. An example of this is an air traffic controller, that needs to monitor a series of data streams to maintain the secure landing and takeoff for planes. The importance of situational awareness (SA) in operating complex dynamic systems have been described extensively in the literature. Endsley [12] describes the elements of SA, how to obtain it, and the influence it has on decision making. Endsley describes the process of obtaining SA in 3 levels: *perception*, *comprehension* and *projection*. *Perception* is the ability to perceive the environment you are in, what is around you, who is around you and if they/it are moving or static? Based on these observations the observer will *comprehend* the significance of each of the elements in the situation. In the end the observer will be able to *project* the future of that situation. Obtaining a high levels of SA is not necessary a spontaneous moment but will be developed over time. The individuals SA in a certain moment in time is therefore not only influenced by the situations present but also its past.

A series of human factors affect the individual's ability to obtain high level of SA. Every individual is different in their ability to remember (working memory and long-term memory), their experience and their individual goals. This affect where the individual places their attention, when *collecting* information from a situation and their ability to actually perceive what is happening. New information stored in the working memory will be processed by comparing the observation with previous observations stored in the long-term memory. This way, a learning effect is introduced where so called mental models are developed. With time and repetition of similar situations a certain level of autonomy is developed along with a more robust confidence in interpreting what kind of information is received. The ability to divide attention also varies a lot from person to person. It has been shown that cognitive errors are highly related with the individuals ability to split their attention towards multiple tasks at the same time [12]. The goals of the individual are linked with their process of obtaining high level of SA. To reach a goal, a plan is derived based on the inputs from the environment. The individual is selective in what parts of the scene that is labeled as important for them to reach their goal.

In the above paragraph the human factors have been described, but outside factors also influence how individuals obtains high level of SA. Outside factors such as the system the individual is operating and the nature of the task itself. The system has an information stream that is processed and displayed to the operator of the system through some kind of interface. A system might have shortcomings in the amount and accuracy of the data that is collected. The sensory of different system have technological restrictions that makes it impossible to gather all kinds of data. The data might not be presented to the operator as the interface is not set up for that kind of data. Design of an interface is therefore of crucial importance for the individual's probability of obtaining complete SA [11]. Systems and tasks are increasingly getting more complex with more data streams entering the systems and thereby increasing the number of components an operator needs to interact with. This tends to increases the workload needed to achieve a given level of SA. Individuals with insufficient cognitive abilities might therefore fail to reach the desired level of SA which will affect their performance. With the implementation of an increased level of automation in a given system, an individuals level SA might be reduced. It has been argued that the reduction of manual involvement in a task and the loss of feedback from a system, will reduce a individuals level of SA [12]. With this said some manual tasks could be replaced by automation and would enhance SA and reduce the workload experienced. It is up to the system designers in collaboration with the operator to determine what task should be automated and to what degree. To achieve a high level of SA it is important that all of the above aspects are taken into consideration when developing a system. In figure 6 the different aspects presented above is shown in Endsley's model of situational awareness.

The importance of thinking SA into the design of a system has briefly been covered in the above section. Endsley [11] developed a number of principles for SA-oriented design.

2.6.2 Measuring situational awareness

Validating and measuring if a concept actually works as intended, is an important part of a design process. In this section the different methods for measuring SA will be presented.

Salmon et al. [40] describes 5 technique's to measure SA: freeze probe recall techniques, real-time probe techniques, post trial subjective rating techniques, observer rating techniques and process indices. In table 2 the different methods is presented.

The technique used to assess SA in the experiment conducted in this project, was a combination of the *freeze probe recall techniques* - and *the real-time probe techniques* - SPAM developed by

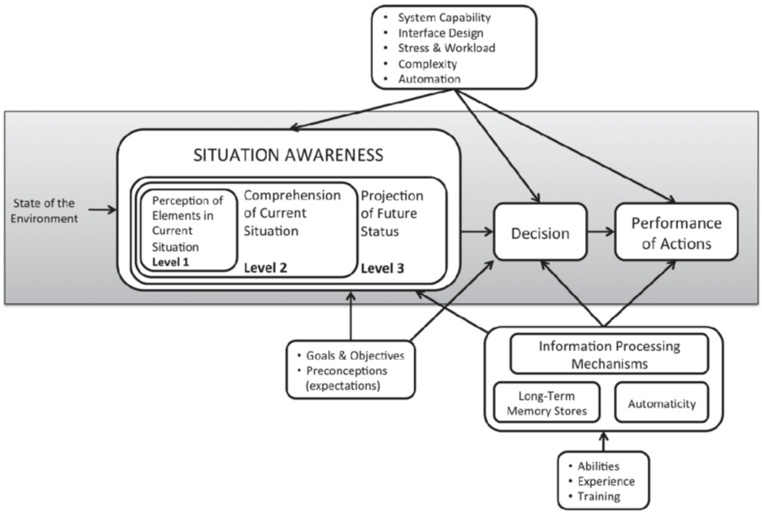


Figure 6: Endsley model of situational awareness: Shows that human factors and external factors influences a operators ability to reach high levels of SA and how this affects decision making.

Method	**Description**
Freeze probe recall techniques	The main task is paused and a question regarding the current situation is asked. The answer is compared to the state of the system.
Real-time probe techniques	During the task a question will be asked about the current task, without freeze. The most popular example of such method is the SPAM where SA is measured on the reaction time on the answer and the perceived workload is measured on how fast you react to a question being asked.
Post trial subjective rating techniques	The test subject is asked to rate their perceived SA after the task is complete.
Observer rating techniques	The task is being observed without intervention at any point. The test subject is assessed based on predefined SA behaviour. The technique is criticized on its validity as the observations are subjective for the observer.
Process indices	Recording the process of developing SA using eye- tracking or asking the test subject to *think out load.*

Table 2: The 5 different measurement techniques of SA presented by [40]

Durso [9]. Which will be further described in the experiment section.

The section presented how both human factors and outside factors influences an individuals ability to obtain all 3 levels of situational awareness and how to act based on these. SA is highly influenced by the individual's cognitive abilities and experience with a system. The complexity of a system have consequences for the workload experienced while operating the system and therefore also how hard it is to obtain SA.

When being present in another environment than the one you are physically situated in, it is important to understand and be aware of the environment which you are telepresent in. SA is therefore an important factor to consider when designing a gaze controlled telepresence robot system. This should be considered to enhance the user experience of operating the robot system and make navigation in unfamiliar environments easier. Making the navigation task easier should carefully considered, since it was shown that implementing automation into a system is followed by a risk of reducing the level of SA.

2.7 Connection between presence and situational awareness

This subsection presents how presence and situational awareness is related. The aspects that determine if a person reaches a certain level of SA have many overlapping elements with the feeling of presence. This section will be based on the SA and presence literature, and our own understanding of the principles.

Correlation between situational awareness and presence in the literature: Jung et al. [22] investigated the correlation between SA, presence and performance when playing a simple game on a hand held console. The study showed that with a higher level of SA the more presence a person felt. This is backed up by Riley et al. [38] that investigated the possibility of quantifying telepresence experiences. The study concluded that "*SA and attention could serve as a objective measure of telepresence*".

Own understanding: SA is defined by by Endsley [11] in 3 levels - *perception, comprehension* and *projection*. The illusion of presence consists of the following 4 factors: The illusion of being somewhere you are not, self-embodiment, physical interaction and social communication. Each of the 4 factors of presence have been compared to the 3 levels of SA to see how they might influence each other, this can be seen in figure 7.

The illusion of being somewhere you are not: As described in section 2.5 the feeling of presence is highly connected with the quality of the simulation. This could be aspects such as number of visible details, scene realism and technology issues (latency, resolution, framerate, etc.). Therefore we argue that the first level of SA (perception) is closely connected to the quality of the environment. If it is not possible to clearly see which objects that are present in the environment, it will complicate the process of perceiving the environment. On the other hand if the individual is in an illusion that they are present somewhere else than where they are physically situated, their perception of the environment will be *sharpened*. This is a result of the better correlation between the sensory of the individual and the stimuli from the environment.

Self-embodiment: If an individual reaches the illusion that the body they are controlling in a simulated environment is their own, the feeling of presence is often higher. When this happens the ability to comprehend and project dimensional aspects of an environment gets more intuitive. If you are moving around in a room, manifested in a robot, then you tend to have a better sense of how long/wide the room is. This is a result of you having a reference point in the dimensions of the virtual body.

Physical interaction: Physical interaction with an environment provides feedback from the surroundings. This gives a person a more comprehensive sense of what is around them. If you walk/drive into a wall or over some obstacle, the system will provide some sort of feedback signal that indicates something has happened. The extra sensory information can be used to perceive the environment in new ways and be used to comprehend how this new information might influence future actions in the environment.

Social communication: Communication between two individuals is not only limited to a verbal aspect, but also non-verbal aspects. To have the illusion of social interaction, not only demands the ability to hear the other person, but also to see him/her, so that small gestures, eye movements etc. can be registered. Here it is important not only to be able to perceive the other persons behavior, but also to comprehend the meaning it might have for the conversation. Therefore if a high enough level of SA is not achieved, a lot of information from the other person will therefore be lost.

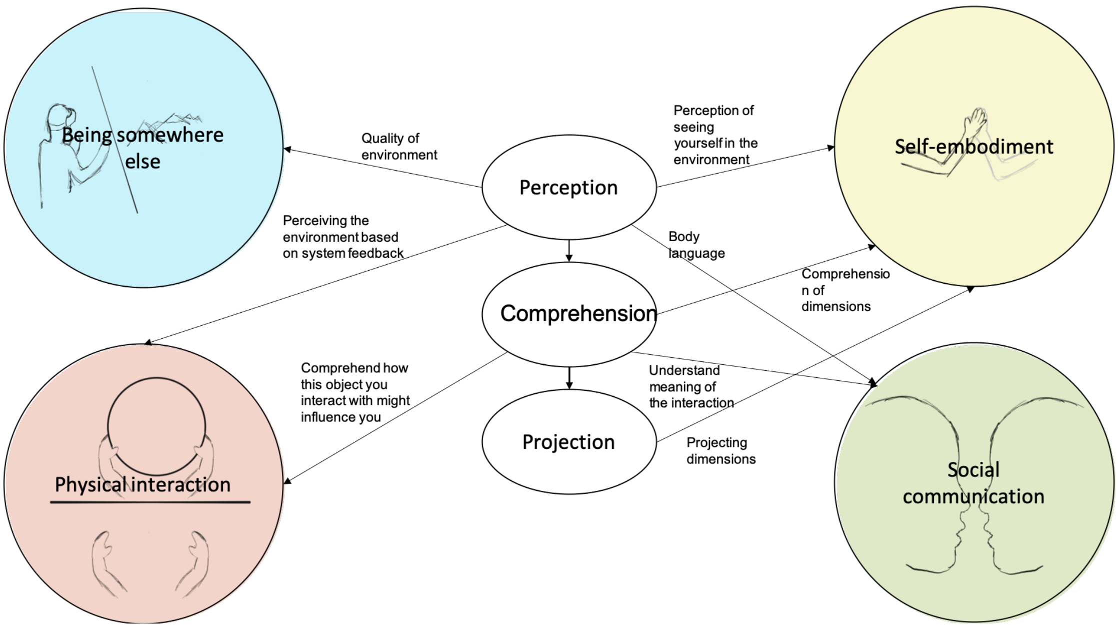


Figure 7: Connection between SA and presence: The figure explains how the 4 aspects of presence relates to the 3 levels of situational awareness

As Riley et al. [38] showed in their study it seems that there is a close relation between the SA and presence experienced in teleoperation. This leads to the investigation of how this could be measured? Is it necessary to measure both presence and SA if they are a representation of each other, or could we create an objective measure that provides information on both parameters?

2.8 Summary prior research

This section gave a review of the the empirical material that establishes the background knowledge for this project. The literature presented shows that it is possible to control a teleprensece robot using gaze as a control input. Gaze controlled telepresence robots could create big value to the lives of motor disabled persons. Allowing motor disabled to be physically embodied in a robot could help them experience events they would not usually attend due to accessibility problems (concerts, conferences etc.) and most importantly it could provide motor disabled with jobs as OryLab showed [3].

The use of gaze to control a telepresence robot is followed by a series of problems. One of the main recurrent problems in telepresence is the technical problems as lag, latency etc. resulting in bad image quality that leads to a reduced feeling of presence and a decrease in spatial awareness with the missing depth perception in the video image. Another problem presented is that navigating a regular teleprensence robot in unfamiliar environments often is very cognitively demanding, and that gaze control is rated as more cognitively demanding than the use of a traditional control inputs, but at the same time it serves as the only input method for some motor disabled. It was also shown that an ALS-patient could operate his wheelchair using gaze as a control input with the help of autonomous assistance - autonomous functions should still be implemented with care since it might reduce the operators spatial awareness and in the end SA. Another problem in the robot operation is that operators has to wear a HMD. Thus, measuring presence using classic measurements the user will have to take off the HMD to fill in a questionnaire, and risks the break in presence.

Based on the knowledge gathered about what different eye-movements can tell about a persons experienced workload, awareness and alertness, and the use of saccade tests, we suggest that a saccade test might be able to provide us with information regarding the presence and SA of the user, which was described as closely related. Thus, for the experiment a pop-up saccade test have been implemented in the interface of the robot, that is very similar to the structure of the SPAM-queries.

3 Implementing a saccade test in Unity

3.1 Our approach

In this section the implementation of a saccade test for the conducted experiment is presented. A pro-saccade test was chosen as the most simple measure of how *alert* the test subject would be to some target appearing in their field of view. Furthermore the test had many similarities with the design of the SPAM measure of situational awareness. Here the test subjects response time on some stimuli estimates the perceived workload of the test subject. The purpose of implementing a pro-saccade test, was to investigate if there was a correlation between a persons saccadic eye movements, their driving related performance and cognitive workload. If a person is quick at responding to a suddenly appearing target, then it can be argued that they have a surplus in cognitive capacity. This could indicate that the driving task itself is not so challenging that other task gets neglected. The persons ability to perform this relative simple task will be compared with the other aspects of the experimental parameters, such as their ability to operate the robot without any collisions.

The aim of implementing the test is to examine what the results of a pro-saccade test can provide of valuable information, when the test subject wears a HMD with built in eye-trackers to control the robot. Could the test potentially be used as an alternative to more intrusive measures such as the SPAM-queries used to assess situational awareness?

3.2 SaccadeMachine

The newly developed software called SaccadeMachine by Mardanbegi et al. [29] was used to analyze the data from the pro-saccade test in the experiment. This software takes a large number of data inputs and exports the analyzed data in csv-files. The software takes a folder of csv-files of individual trials as an input. Table 3 shows the required measurements for the software to run.

Most of the data such as the coordinates of the subject's gaze, target position and the timestamp is extracted from the FOVE headset and Unity. The rest of the data is static and are entered by the examiner, see saccade panel in Unity in figure 8. The only thing that had to be created from scratch was a way to detect if the test subject was performing a saccade. We therefore created a simple saccade detection algorithm. For every frame it examined the previous and current gaze coordinates and evaluated the absolute difference. If the difference was bigger than a predefined threshold a saccade was detected. When the algorithm detected a saccade the value of *IN_SACCADE* would change from null to 1, see figure 9 for an overview of the algorithm. The algorithm worked in *real time* and dynamically detected the saccades. Thus, first coordinate of a saccade was not registered as *IN_SACCADE* as the algorithm needed the previous coordinate to register a saccade. This could have been avoided if the gaze data was analyzed after the test had been conducted. This fault was seen as acceptable, since the missing reading represented such a small percentage of the collected readings in one saccade. If the analysis had been conducted after the experiment other and more complex algorithms could have been used. Algorithms such as PyGaze could have been used to detect saccades [4].

Using SaccadeMachine When the data from the saccade test had been collected it was uploaded

Input	Description
GAZE_X	X-coordinate of gaze position
GAZE_Y	Y-coordinate of gaze position
IN_SACCADE	Is the test subject performing a saccade
FIXATION_INDEX	Index of the fixation
SAMPLE_MESSAGE	Target on/off set and fixation on/off set
TIMESTAMP	Timestamp
TRIAL_INDEX	Number of trial
RESOLUTION_X	Screen resolution x
RESOLUTION_Y	Screen resolution y
IN_BLINK	Is the test subject blinking
TARGET_X	X-coordinate of the target
TARGET_Y	Y-coordinate of the target
SUBJECT_GENDER	Gender
SUBJECT_AGE	Age
SUBJECT_NAME	Name - Used the ID of the test participant
SUBJECT_GROUP	Can be used to group test subjects
IN_FIXATION	Is the test subject in a fixation

Table 3: SaccadeMachine data input: Measurements required for the software to analyze the data

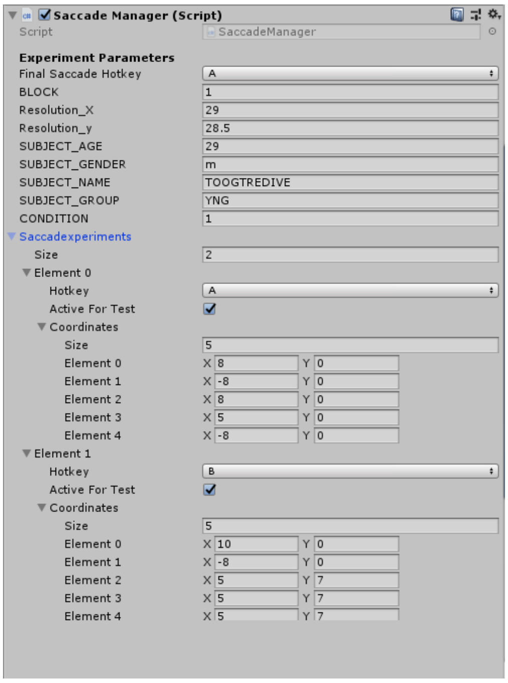


Figure 8: The control panel in Unity, where the different information about the test subject and saccade test are entered

to the software. The type of test was chosen, and the screen dimensions and distance from the screen was entered. How these dimensions was chosen, is discussed in section 3.3. See figure 10a

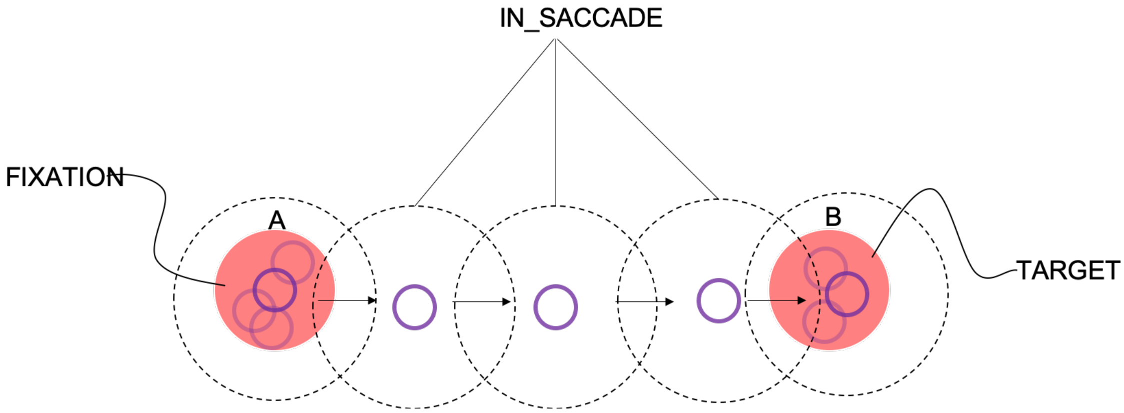
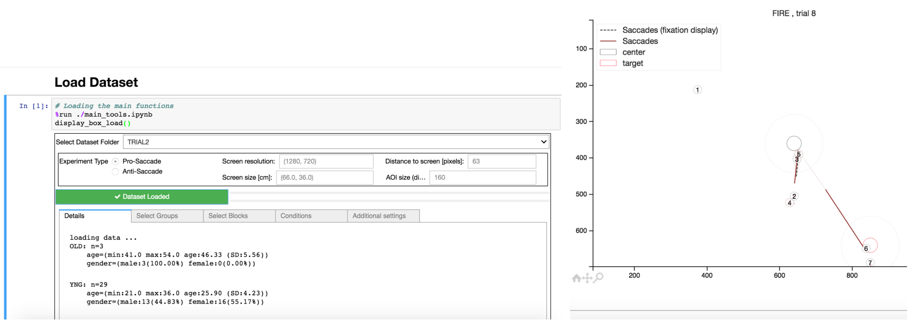


Figure 9: Saccade algorithm: Determines if the user is performing a saccade by comparing if the current point of gaze is within a threshold of the previous point of gaze



(a) The data is being uploaded to the software (b) Plot of a pro-saccade test

Figure 10: Pictures from the interface of Saccade Machine

for an overview of SaccadeMachine's upload screen [29].

The software detects if there has been a calibration offset by looking at the fixation point and the median coordinates of each fixation trial. The software offers a correction of these calibration offsets when analyzing the data [29]. The software provides the following results after analyzing the data set, see table 4 .

In addition to the results in table 4, the software create custom graphs that presents the results. For each test subject every saccade trial can be visual inspected, see figure 10b for an example.

3.3 Implementation of the pro-saccade test in Unity

The saccade test was implemented in the game engine Unity where the rest of the user interface was implemented. We created a virtual *screen* with a transparent glass background and a solid red dot, see figure 11.

The panel was designed in Unity so that the experiment conductor easily could enter the test subject's information, the coordinates for the target position and the number of positions during the test, see figure 8. During the experiment the experiment conductor could press a *hot key* on the keyboard of the computer and the test was initialized. While the test was running all the data

Title	Description
TRIAL_SUCCESSFUL	Total number of successful trials (with correct or corrected saccades)
TRIAL_CORRECTED_ONLY	Total number of trials with corrected saccade
TRIAL_CORRECT_ONLY	Total number of trials with the first saccade detected as correct
TRIAL_FAILED	Total number of failed trials (wrong saccade, bad data quality, etc)
LATENCY_FIRST_SACCADE	Latency of the first saccade after target onset regardless of its direction
LATENCY_CORRECT_SACCADE	Latency of the first correct saccade after target onset
LATENCY_CORRECTED_SACCADE	Latency of the corrected saccade after wrong saccades
LATENCY_ANY_CORRECT_SACCADE	Latency of the correct or corrected saccades
LATENCY_WRONG_SACCADE	Latency of the first wrong saccade after target onset
AMPLITUDE_CORRECT_SACCADE	Amplitude of the correct saccade
AMPLITUDE_CORRECTED_SACCADE	Amplitude of the corrected saccade
AMPLITUDE_WRONG_SACCADE	Amplitude of the wrong saccade

Table 4: The results returned from the analysis using SaccadeMachine [29].

Figure 11: Picture of the pop-up saccade test during the experiment

was collected in the background and stored locally on the computer with a dynamic file naming system. The full C# code for the pro-saccade test can be found in appendix B.1.

3.4 Challenges and sources of errors:

During the implementation process a series of problems and concerns was encountered and needed to be taken into account when developing the test.

Frame rate: There was a concern regarding the hardware's capabilities. Mardanbegi et al. [29] used a 500 Hz eye tracker (500 readings per second) to demonstrate the use of the SaccadeMachine, where as the FOVE HMD used in this project only had a 70 Hz eye tracker. This raised the question if it was possible to extract meaningful data from the participants as saccades are such rapid eye movements. After some research into the speed of a saccade and a look at the test samples provided with the software, it was concluded that it should be possible to get meaningful results for a test.

New software: The problems encountered with the SaccadeMachine software was difficult to solve. As the first third party who used the software, there was no online forum where we could ask our questions. In general, the feedback from the software was very sparse and the input data needed a very specific format, meaning that there was no flexibility in how the data could be formatted when uploading the data.

VR screen: A issue emerged regarding distance assessments in implementing the saccade test in a virtual environments. SaccadeMachine requires the size of the screen and distance from screen in cm as inputs. The screen resolution in pixels is also required as an input. Thus, we would have to interpret the distance in our 3D-environment. The format of the saccade test pop-up was created in the same scale as a 1280x720 format, and was used as the screen resolution, even though the FOVE HMD had a higher resolution. The size and distance was determined by inspection of the Unity environment. The distance from the users point of view to the *screen* was know in arbitrary units in Unity. So was the dimensions of the screen, see figure 12. Thus, these was converted to centimeters.

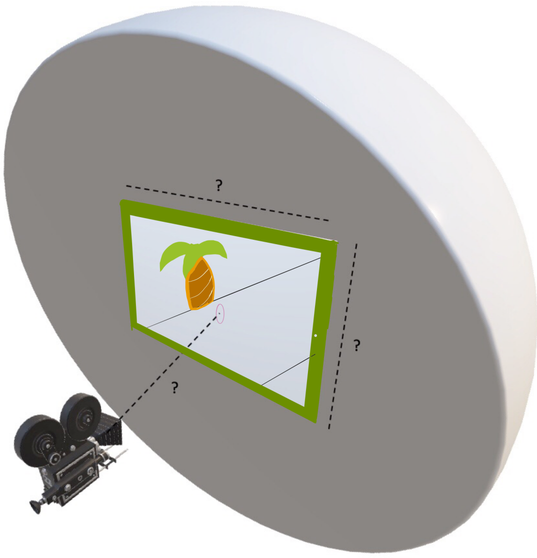


Figure 12: Determining the screen size of saccade panel

3.5 Summary of experiment preparation

This section explains how a pro-saccade test was implemented in Unity. The problems and solutions associated with implementing a saccade on a 2D screen in a 3D environments is also explained. The main problems/errors of implementing a pro-saccade in the test setup used in this study was determining spatial measures of the saccade pop-up test in Unity and the low frame rate of 70 Hz of the FOVE HMD. Most of the measurements that needed to be collected during the pro-saccades could be collected through the FOVE-headset and Unity environment, such as the coordinates of the gaze position relative to the target position. The only measurement we needed to create outside the standard measures was a saccade detection algorithm. A simple dynamic saccade detection algorithm that detects the saccade based on previous gaze position was implemented and is presented in the section. A brief presentation of the SaccadeMachine software which was used to analyze the data gathered in the pro-saccade tests can be found in the section [29].

4 Experiment/method

This section explains in detail how the experiment was planned, setup and conducted. The experiment was a part of a bigger ongoing research programme **GazeIT** [16], and was planned and conducted together with Guangtao Zhang as a part of his Ph.d. project. The research groups primary research question for this experiment was how simulation-based training environments impacts the learning curve in gaze operation of telepresence robots [54]. This project uses many the of same elements in our research question. Thus, Our contribution consist of the execution of the experiment and the implementation of the pro-saccade test.

Aim: Them aim of the experiment is to examine how virtual training environments and changes in the environment influences the learning curve when operating a gaze controlled telepresence robot. Secondly we examine what type of information the results of a pop-up pro-saccade test during robot operation can provide about the test participants.

4.1 Experiment description

The experimental method is a between-groups design. The experiment is divided into three parts, one pretrial, 1 training sessions consisting of 5 trials and a final trial, see the experimental design in appendix A.1. In the pre- and final trial the test subject had to complete some given tasks. In the training session no tasks would be given, and thus no interruptions would occur while operating the robot. The experiment was counterbalanced according to appendix A.11.

In the experiment the test subjects had to operate the robot through a maze while wearing a HMD with build in eye tackers. In the HMD's display the user had a live video stream from the remote environment. A control panel with a control border was visible to the user and if they gazed inside the control grid the border would light green and they had control over the robot (13). If they looked outside the control grid, the border would light red and the robot would stop moving and the operator could look around. The control panel/grid was same used by Zhang et al. in [53], and can be seen in figure 13. The figure illustrates the mapping of the robot controls in the interface. The grid was not visible for the user. In the bottom of the screen, outside the control panel, the buttons of the interface can be seen. The only button used in this experiment was the *Park button* which is the green **D** indicating that the robot is in drive mode, if the user gazed at the button the robot would stop driving and go into park mode (figure 13). The purple dot visible on the video stream is a cursor and indicates the users point of gaze, this could be used by the user to see where their viewpoint is registered if the calibration should be a little bit off.

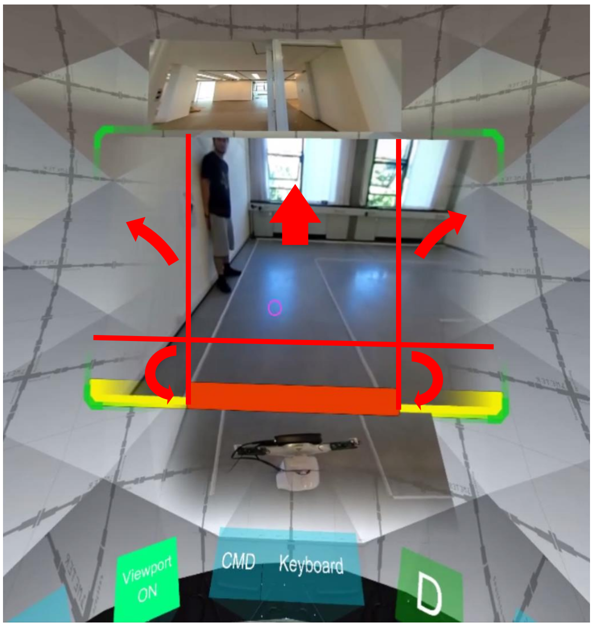


Figure 13: The control grid used in the experiment illustrating the control mapping. The grid was also used by Zhang et al. in [53]

4.2 Participants

For the experiment 32 test subjects was recruited, they were all able-bodied. The test participants all received a gift card with a value of 100 kr. for participating in the experiment. The mean age of the test participants was 28.19, SD=7.31, max age=54, min age=21. 15 of the recruited test participants was female (0.47%) and 17 males (0.53%).

4.3 Setup

The experiment was conducted in Skylabs DesignLab. The duration of the experiment for each test subject, varied from 30 minutes to 3 hours at most, the average duration was just below 60 minutes. The test setup for the experiment can be seen in Figure 14.

Equipment: For the experiment two computers was used. One desktop PC to run Unity and one laptop on which the test subjects could fill in the presented surveys in LimeSurvey. Both computers was set up outside DeisgnLab. The last element outside the test room was the FOVE HMD connected to the desktop PC. The robot used for the experiment was a **Padbot P1**, which was placed inside DesignLab and mounted with a **Ricoh Theta V** - 360° camera. The robot communicated with the HMD and desktop PC through a Raspberry Pi using a closed internet connection. The last element in the test room was the maze walls which was placed on the floor according to the maze layout presented in the counterbalance scheme (A.11). Two persons

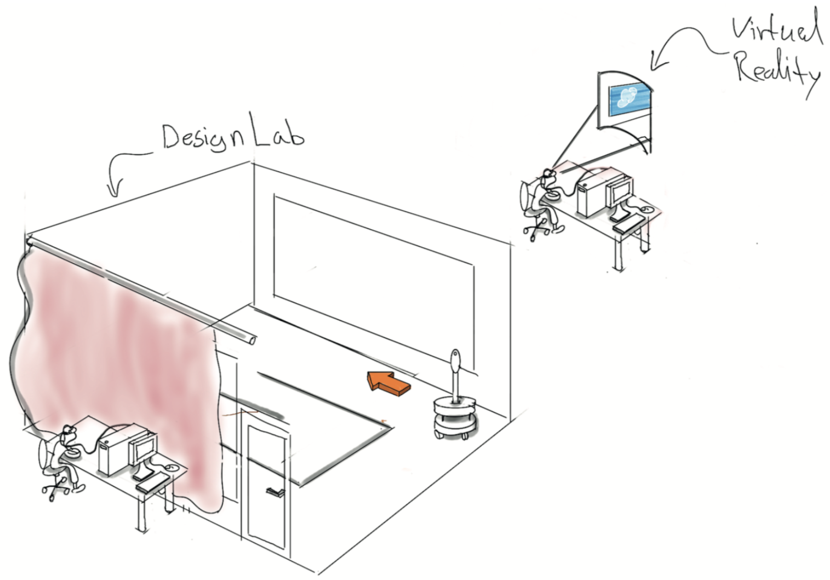


Figure 14: The experimental setup in DesignLab. Left: A user operating the robot inside DesignLab. Right: A user is trained in VR

was needed to run the experiment. One experiment conductor who would be outside DesignLab together with the test subject. The other person had to be inside the room to note the number of collisions, interact with the test subject and provide them their task. Two phones was used to provide two-way communication between the test subject and the person in DesignLab. The cellphone outside the room was placed in front of the operator and put on loudspeaker.

To run the experiment the existing platform and interface developed by *GazeIT* [8] was used. This is the same platform as the Saccade test was implemented in before conducting the experiments. Two different versions of the software was used in the experiment, one when operating the real robot in reality environment and another when operating a virtual robot in a virtual environment.

4.4 Procedure

The structure of the experiment was planned and written down in a protocol/checklist. A manuscript containing instructions (A.6) was read aloud before each trial and a picture of the control grid was used to explain the controls. This was done to assure all test participants received the exact same amount of information before the test. The checklist (A.10) was based on the experimental protocol (A.9), and was followed carefully to assure all trials would be executed as similar as possible. This was done to minimize the errors due to different presentations of the control grid and the tasks.

Before the test participant arrived the maze layout inside the test room was prepared and the information regarding test-layout was put into Unity to prepare the data collection. An overview of the experiment procedure from start to finish can be seen in figure 15.

Upon arrival the test subject was presented with a consent form, that they would have to read and sign to participate in the experiment A.4. Before starting the experiment the test subjects

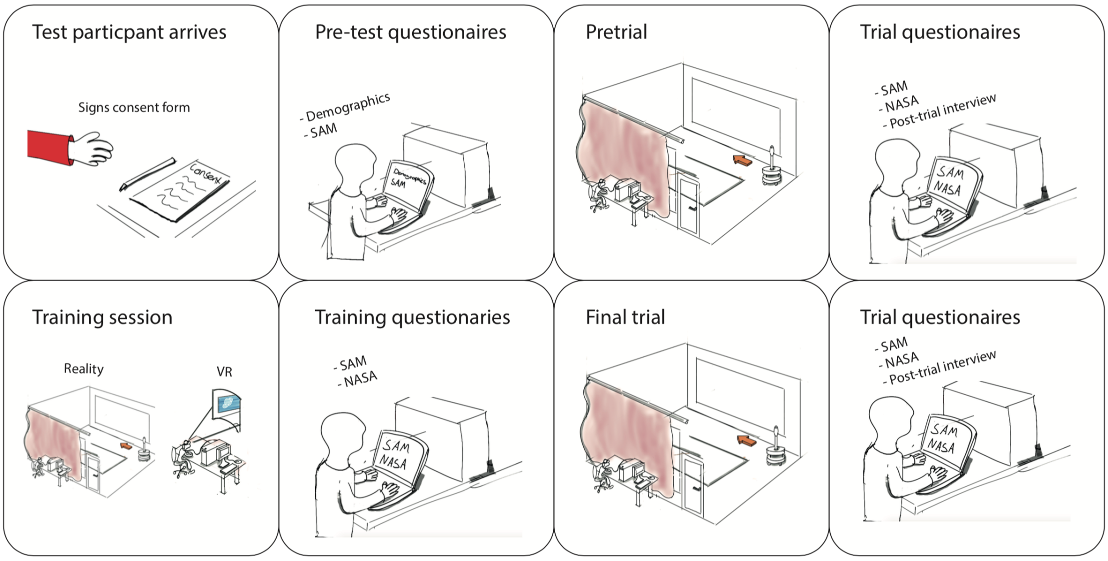


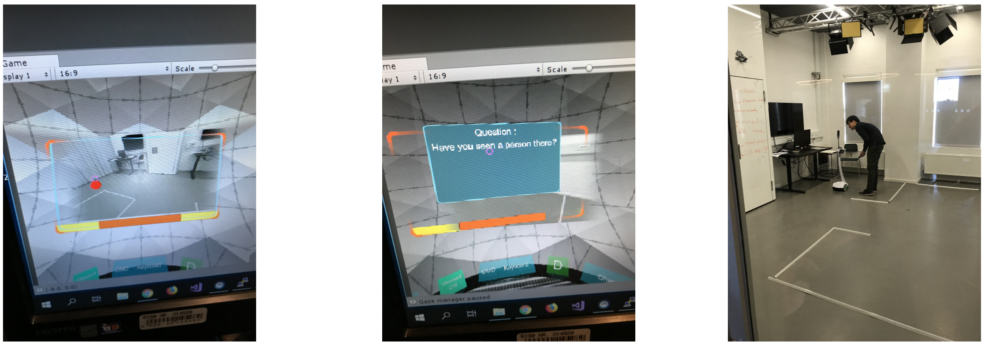
Figure 15: Experiment procedure form the test subject arrives to the end of the experiment

was asked to fill out a demographic survey and a self assessment scheme (SAM). When they had filled in the first part of the survey the experiment conductor would help the test subject put on the HMD and make sure that their eyes was calibrated correctly.

As mentioned the experiment was divided into 3 parts one pretrial, one training session and the final trial. The pretrial was done to see how the test subject performed before receiving training, then a training session of 5 trials was conducted and at the end the user would do a final trial to see the effect of the training. In the pre- and final trial the operator of the robot had to complete three different types of tasks.

1. **Saccade test**: The first type of task the user would have to perform was a pro-saccade test. The control of the robot was stopped and the pro-saccade test would begin. The users was instructed to follow the red dot, before the trial began, see figure 16a.
2. **SPAM-querries**: The second type of task appeared in the users view point as a blue screen with a question on it. The operator was instructed to answer the questions orally and respond as fast as possible, when the pop-up appeared the robot control stopped, see figure 16b.
3. **Person interaction**: Before operating the robot the test subjects was given their first task, which was to drive the robot towards the person in the room and to avoid the maze on the floor. When they reached the person they had to put the robot into park mode. When the robot was in park mode the operator would be instructed in their next task. Which was to operate the robot to the other window and then drive back to glass door in the room, see figure 16c.

Before starting the pre- and final trial the test subjects was informed that they were facing south, which is an information that they needed to answer one of the situation awareness queries. During the trials the experiment conductor activated the saccade test and SPAM-queries manually, the



(a) Pro-saccade test (b) SPAM-queries (c) Person interaction task

Figure 16: Pictures of the three type of tasks in the experiment

answers to the queries and errors during the experiment was noted on a separate scheme for each test subject, see appendix A.7. For an example of either the pretrial or the final trial including the task, see figure 17.

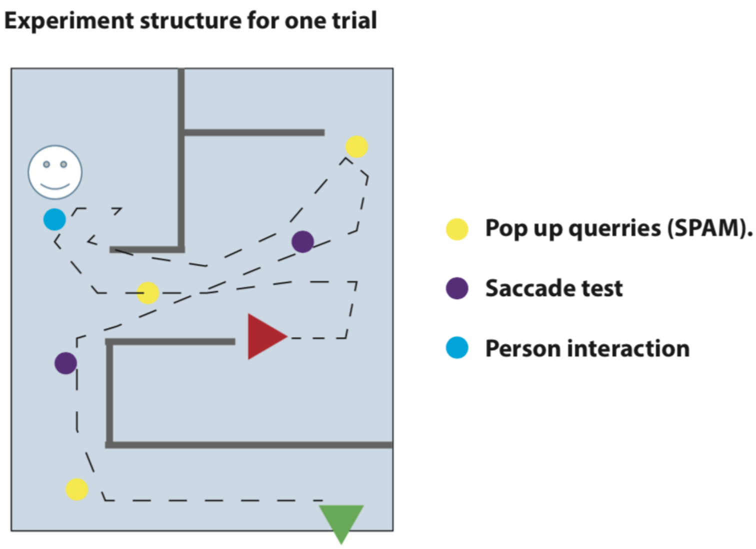


Figure 17: Illustration of the progress in either a pretrial or a final trial, that shows how the tasks appeared

After the pretrial the test subject was asked to take off the HMD and was again asked to fill out an self assessment manikin(SAM), a NASA-TLX survey to asses their own perceived workload in operating the robot and a post trial interview where they could share their experience in operating the robot. The test subject would then put on the HMD and get calibrated again. When calibrated they would have to go through 5 training sessions which would vary depending on their test conditions. After the training session they were asked to take of the HMD and fill out the SAM and NASA-TLX surveys again. The final trial followed the same structure as the first.

4.5 DATA-collection/measurements

Throughout the experiment several types of different data was collected both qualitative and quantitative.

LimeSurvey During the experiment the test subjects would have to answer different types of surveys, the following surveys was presented to the test subject using LimeSurvey on a laptop placed outside DesignLab.

- **Demographic information:** The first questionnaire presented was regarding demographics of the test subject and their prior knowledge and experience with gaze interaction, teleoperation and Virtual Reality.
- **SAM - Self-Assessment Manikin (SAM):** The SAM is a pictorial self-assessment rating scale of the operators momentary feelings of pleasure, arousal and dominance (A.3). This questionnaire was presented to the test subject four times. This is done to see how people assess their own emotions during the robot operation and the experiment.
- **NASA - Task load index(TLX):** The NASA - TLX is a questionnaire where the user asses his/her perceived workloads on 6 different sub-scales ranging from 1-21 in performing a given task, this survey is often used in human-factors experiments [51](A.2). Thus, the questionnaire was used to measure the operators perceived workload in operating the telerobot. The data is analyzed using the RAW-TLX, where the individual workloads are analyzed separately.

Experiment conductor notes

- **Post - Trial interview:** After the pre- and final trial the user evaluated the trial orally together with the experiment conductor. The test subject was asked some simple questions to assess how much time was spent, number of collisions, and how confident they were. Finally some open questions was asked such as if they experienced any discomfort, technical errors, how the overall experience operating the robot was and if they have any suggestions for improvements. The answers for the post trial interview was noted in a Google drive document (A.8).
- **Performance notes:** The answers to the situational awareness SPAM-queries was noted as correct or not in a performance scheme (A.7), where the number of collisions and estimated coverage for each trial was noted too.

Unity

- In Unity the data from the saccade test was collected from the FOVE headset which has a frame rate of 70 Hz, this was formatted and saved locally as a csv file.
- The task completion time was collected in Unity by collecting a timestamp at the beginning of the trial and again in the end.

- During the experiment the test subjects level of SA was measured using a Situation Present Assessment Method (SPAM). The version of the SPAM - queries used in this experiment has elements from the *freeze probe recall technique* where the primary task is paused compared to the "regular" SPAM where the questions are asked while the primary task is still going on. The response time to the SPAM-queries is collected in Unity and exported as a JSON-file. The SPAM-queries is activated by the experiment conductor. When the test subject answers the question the experiment conductor presses a button. This is how the response time is collected(A.5)

4.6 Experimental design

To investigate the effect of training three **independent variables** was examined: *Training environment*, *Maze layout* and *Trial order*. When the test subjects were trained with the robot two different *training environments* was used. One was *reality-training* with the real telepresence robot, and the other was *VR-training* with a virtual robot. *Maze layout* varied in two ways, either the test subjects trained with the robot in the *same layout* as their pre- and final trail or else they trained in a *different layout* than they did in their pre- and final trial, see figure 18 for the two different maze layouts. *Trial order* was the *pretrial* before training and the *final trial* after training. In figure 19 the experimental design is presented, based on the counterbalancing scheme A.11.

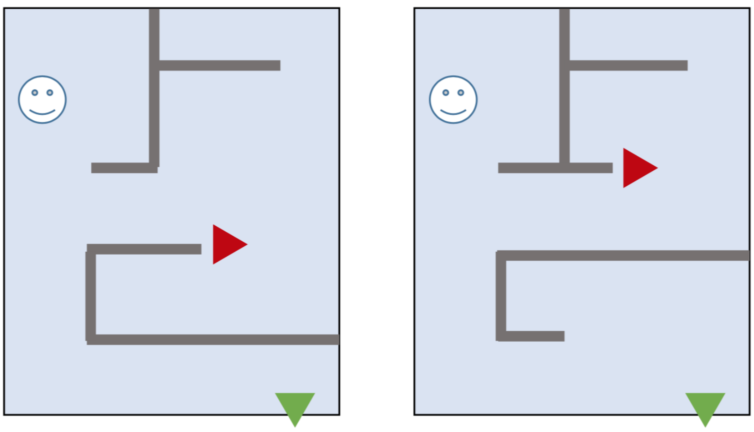


Figure 18: The two different maze layout used in the experiment

Dependent variables of investigation was *Task performance, workload, situational awareness, saccade test results* and *self-assessment* (A.1).

Null hypothesis: When operating a telepresence robot wearing a HMD the users show no difference in their task performance, workload, situational awareness, saccade test results and self-assessment independent of the training environment they trained in and maze layout in the pre-and final trial.

Saccade test hypothesis: The saccade test results shows no correlation with any of the other dependent measures.

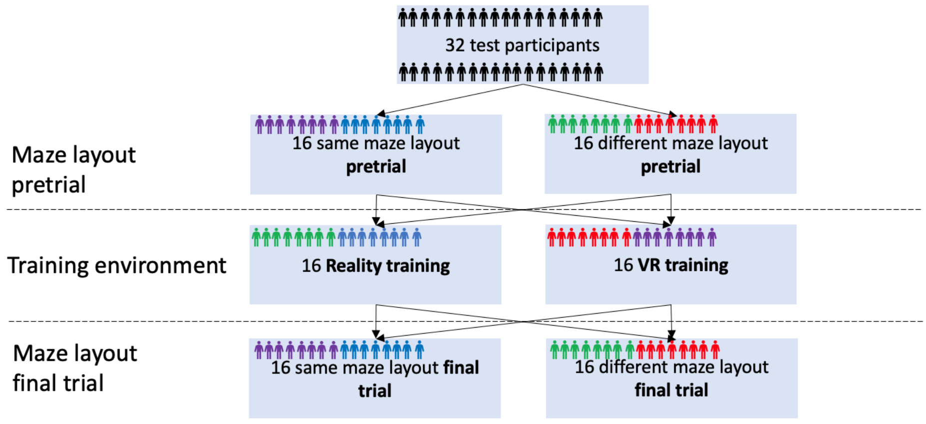


Figure 19: Visual illustration of the experimental design, showing how the participant was divided into groups based on the independent variables

4.7 Source of errors

Virtual environment: The VR-training environment had some errors that has to be mentioned.

1. The maze layout in VR, that was meant to resemble the same layout as the reality, was quite different. There was two key errors in the maze layout: If the user had to drive the robot towards the window as instructed, they would have at least two collisions going back and forth, and after the user had reached the window it was theoretically possible for the user to drive outside the maze and straight to the door (finish). This was realized after test participant seven. Thus, the test subjects after test subject 7 was informed to drive as close to the window as possible without making a collision and asked to drive back to the door through the maze. A two way-ANOVA was conducted between the first 4 participants trained in VR and the last 4 that was trained in VR. The test showed no significance on the duration and number of collisions in any of the 5 training trials.

2. The control input from the gaze was different, instead of having a clearly separated control grid when turning, as with the real robot (figure 13), the control when turning was more gradient, and the virtual robot was driving with a lower speed.

3. The user could not use head movements to pan and look around the virtual environments. This could create bad habits for the user when training in VR.

Experiment conductor errors: When running the experiment it happened that the conductor made an error or two. Here are some of the errors that was made:

1. Forgetting to press **C** as the last SPAM-query, if the last pressed key was not **C** the SPAM data for the trial would not be saved in the JSON-file.

2. Forgetting one of the saccade test. During each trial two saccade test should be presented it happened that only one was shown.

3. Pressing the wrong key for the SPAM-query. When this happened the person would either receive four questions in a trial to makes sure all aspects was covered (perception, comprehension and projection) or either they would receive two questions in the same category.

SPAM-queries: One of the key measures in the SPAM-queries is the response time of the participant, and since the response time depends on the experiment conductor pushing a button this is an uncertainty. This could easily make the results vary a couple of seconds.

Park mode: It was often hard for the test participants to activate the park button. This was assumed to be because the button is placed in the bottom of the screen, which often is where the FOVE HMD has a hard time tracking the eyes, and also reflects that people cannot keep the cursor steady on the park button for the long dwell-time currently needed.

4.8 Summary experiment

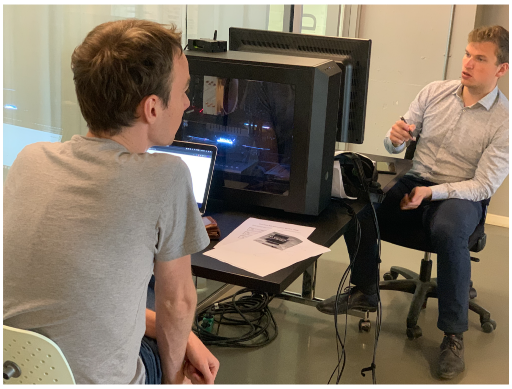
The aim of the experiment was to examine how virtual environments, familiarity of the test environment and trial order affected the robot performance and what a pop-up pro-saccade test during the robot operation task could say about the test subjects.

The experiment was a between-groups design. The experiment was conducted in DTU Skylabs DesignLab and 32 test participants was recruited. In the experiment the test subjects had to operate the gaze controlled telepresence robot through a maze. The experiment consisted of a pretrial, training session (5 trials) and a final trial. During the pre- and final trial the test subjects would receive 3 types of tasks (saccade test, SPAM-queires and person interaction). The experiment examined 3 independent variables: *Training environments*, *maze layout* and *trial order*. The dependent measures was: Task performance (taks completion time and number of collisions), workolad (NASA-TLX), situational awareness (SPAM), saccade test results (pop-up pro-saccade test), self-assessment (SAM) and recollection (estimated performance, user comments ect.).

The source of errors during the experiment was: The VR-environment (controls varied compared to the real robot), experiment conductor errors (some measurements was dependent on the conductors performance- SPAM-queries) and park-mode (eye tracking was poor in the lower part of the FOVE HMD - location of the park button).

The experiment was carried out successfully according to the procedure and design and the data was collected as planned. Figure 20 presents a selection of pictures from the experiment. Figure 20a shows a test subject controlling the robot while the experiment conductor are taking notes and activating the different tasks. Figure 20b shows the experiment conductor explaining the experiment procedure to a test subject. Figure 20c shows the experiment from inside DesignLab.

(a) Test setting outside DesignLab; Test subject (left) and experiment conductor (right)



(b) Experiment conductor (right) explaining the experiment to the test subject (left)

(c) Picture of DesignLab, that shows the robot and how the maze was placed on the floor

Figure 20: Pictures from the the experiment

5 Data Analysis

This section presents the results of the experiment and how the data analysis was carried out. In the experiment various types of data was gathered, and thus there was a need to set up a standard procedure for analyzing the data, which is presented in this section. The section only presents the relevant results of the data analysis, all the results of the data analysis can be seen in appendix C.1. First the results of all 32 test participants is presented. In the end of the section a analysis between the best and worst robot performers is conducted, to see if the saccade test could be used to divide people into groups based on their robot performance.

5.1 Approach: Theory and expectations

The collected data throughout the experiment, was all gathered in a Google sheet document and formatted correctly for the data analysis. The data analysis was conducted using R-studios.

All the dependent variables measured (non ordinal) for the pre- and final trial was first analyzed using a three-way ANOVA test, where *trial order*, *training environment* and *maze layout* was treated as independent variables, to see what learning effect the training had. Afterwards a two-way ANOVA analysis was conducted on the two independent variables *training environment* and *maze layout* to analyze the the training session. A one-way ANOVA test was used to analyze the results between the best and worst robot performers. The ordinal data collected from the self-assessment was analyzed using the non-parametric test Mann–Whitney U test (Wilcoxon rank-sum test).

Before conducting the different ANOVA tests, it was first assured that none of the assumptions for a ANOVA test was violated [24]. We tested for:

- Normality - using the Shapiro-Wilk test
- Homogeneity of variance - using a Levene's test
- The dependent variables must be a continuous measurement.
- Sample Independence - each sample must be independent of each other.
- The independent variables must contain at least two or more categorical groups.

The overall process for the data analysis can be seen in figure 21. If the measured dependent variable failed either the normality or homogeneity of variance assumptions, we used data transformation on the measured value and checked both assumptions again.

If the transformed value did not fail any of the assumptions, this would be used as a the value of interest in the analysis of variance. If the transformed data did not satisfy the assumption of normality or homogeneity of variance, the Mann–Whitney U test was used instead.

The data transformation we applied throughout the data analysis was the Box-Cox transformation:

$$y(\lambda) = \left\{ \begin{array}{c} \frac{y^{\lambda}-1}{\lambda}, \text{ if } \lambda \neq 0 \\ \log y, \text{ if } \lambda = 0 \end{array} \right\} \quad (1)$$

To do the Box-Cox transformation we used the MASS package for R-studios. The library has a built-in function that finds and suggest the most optimal λ. This is done by suggesting a lambda value based on a Log-likelihood in 95% confidence interval. The lambda is then extracted and the transformation conducted.

The transformed value was then inserted into the linear model used for the ANOVA-analysis. If the results showed no difference in significance compared to the untransformed ANOVA-model, the results from the untransformed ANOVA-analysis was reported. This was done based on how the normality issue is addressed by MacKenzie in [26] chapter 6.4.

If either the two- or three-way-ANOVA test did not show any significance of interaction the ANOVA analysis was conducted again, without analyzing the interaction effects. Finally if any of the dependent measures showed any sign of significance, a Bonferroni correction was conducted between the measure and the independent variable that showed a significant main effect.

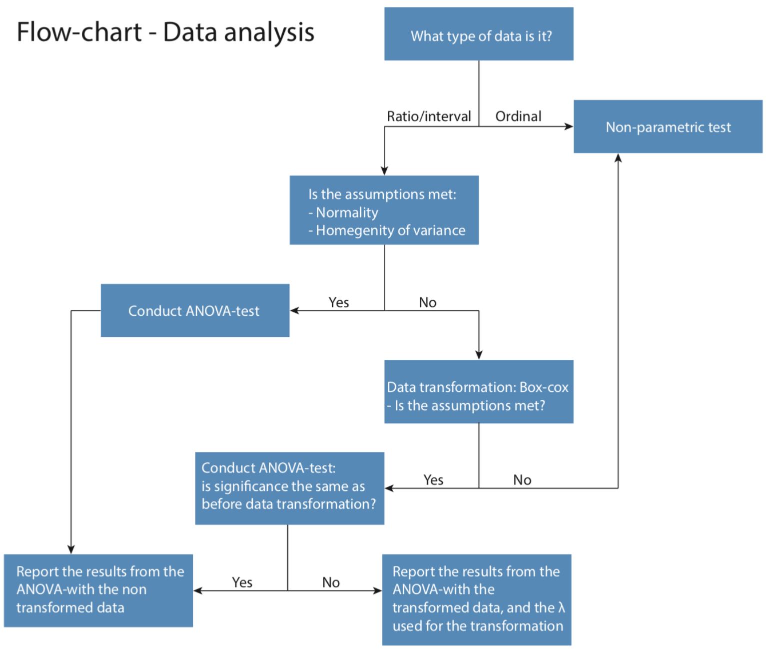


Figure 21: Process diagram for the data analysis, showing the selection of analysis model

10 first vs. 10 last: A one-way ANOVA was conducted between the first 10 and the last 10 test participants, to see if there was any significant difference between the groups. The results showed that there was no significant difference between the groups on any of the dependent measures. Thus, there was no *order effect* of when the test subject participated in the experiment, and the introduction/description of the experiment ca be seen as equal for all.

5.2 Results

5.2.1 Performance

Task completion time

The task completion time is the driving time without the duration of the different tasks included. With a three-way ANOVA followed-up by a pairwise comparison with a Bonferroni correction, significant main effects of task completion time was found on *trial order* $F(1,60) = 34.91, p < 0.001, \eta^2 = 0.366$, see figure 22. No significant main effects of task completion time was found on *training environment*, *maze layout* or any of the interaction effects. This means that neither *training environment* or *maze layout* had any effect on how much a test subject improved their driving abilities.

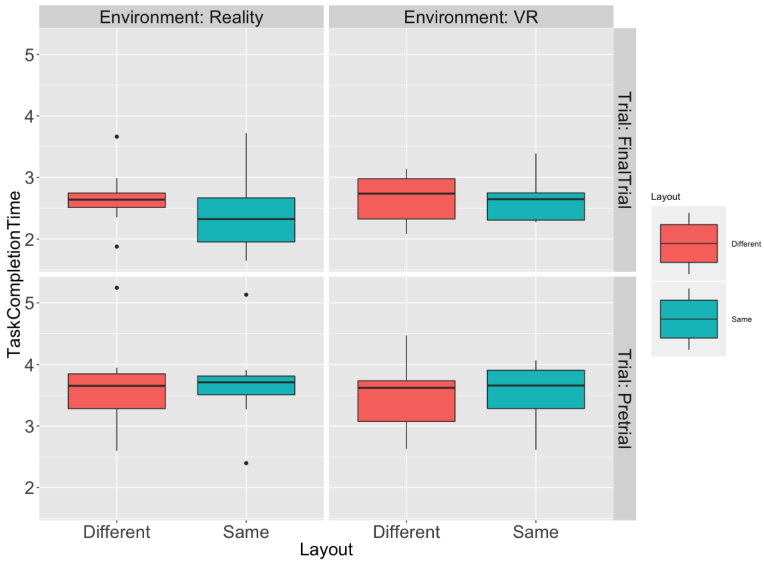


Figure 22: Task completion time

Number of collisions

With a three-way ANOVA followed-up by a pairwise comparison with a Bonferroni correction, significant main effects of number of collisions on *training environment* $F(1,60) = 6.69, p < 0.05, \eta^2 = 0.076$ and *trial order* $F(1,60) = 20.95, p < 0.001, \eta^2 = 0.239$. The mean number of collisions for the pretrial was 5.06 (SD=4.06) and 1.45 (SD=2.20) for the final trial, see figure 23.

Training session

We analyzed the task completion time and number of collisions in the training sessions using a Mann–Whitney U test, with *training environment* and *maze layout* as the two independent variables, because the data did not satisfy the normality assumptions, and after data transformation not all measurements satisfied the normality assumption. Thus, a Mann-Whitney U test was chosen.

In all the training sessions we saw a significant effect of both collisions and task completion time on the *training environment* but no effect of *maze layout* was found. This difference between training environments did not effect the learning curve between groups. Figure 24 and Figure 25 both illustrates the learning curve of task completion time and number of collisions between training trials.

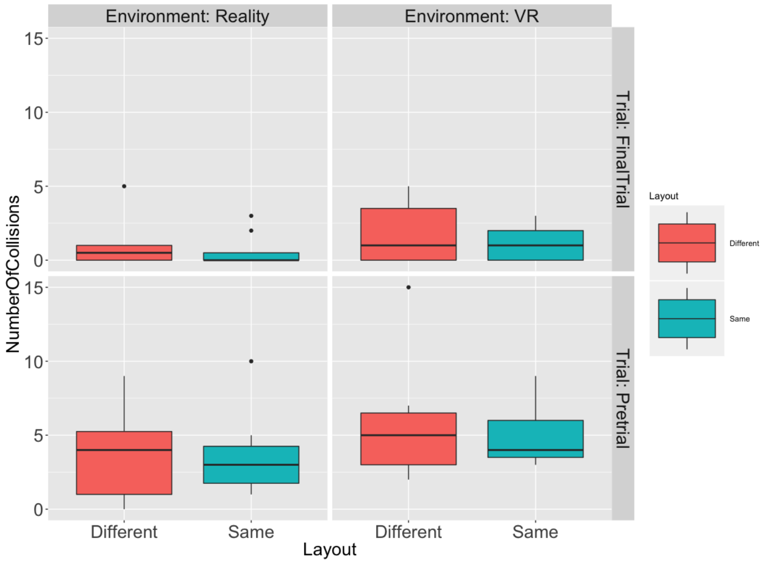


Figure 23: Number of collisions

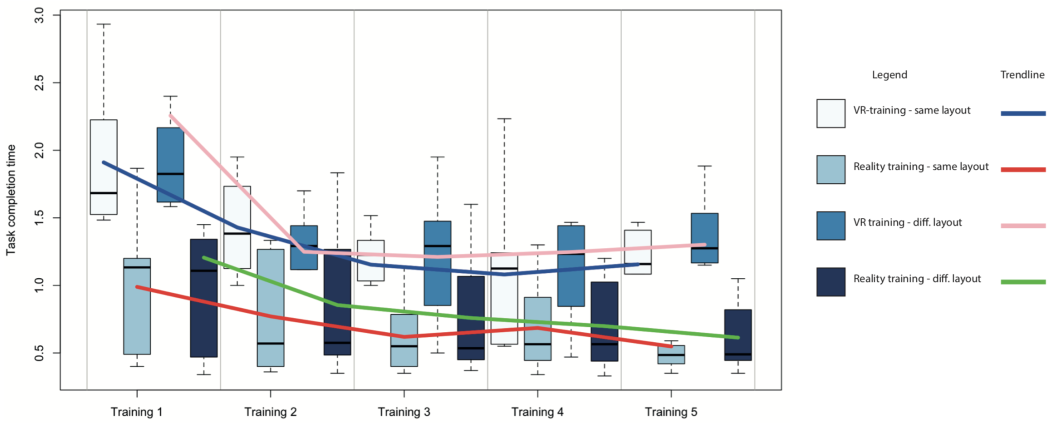


Figure 24: Duration of each training session between groups

5.2.2 Perceived workload - NASA TLX

6 three-way ANOVA's was conducted based on the responses to the NASA TLX we received after the first and final trial. The results showed no significant main effect of perceived mental, physical and temporal demand of the task on any of the independent variables or any of the interaction effects.

With a three-way ANOVA followed-up by a pairwise comparisons with a Bonferroni correction, significant main effects of perceived performance, effort and frustration was all found on *trial order*. Perceived performance - $F(1,60) = 7.88, p < 0.01, \eta^2 = 0.108$, perceived effort - $F(1,60) = 8.87, p < 0.01, \eta^2 = 0.128$ and perceived frustration - $F(1,60) = 8.22, p < 0.01, \eta^2 = 0.118$. These findings indicate that the test subject experienced a lower workload in the final trial than the pretrial. These effects can all be seen in figure 26.

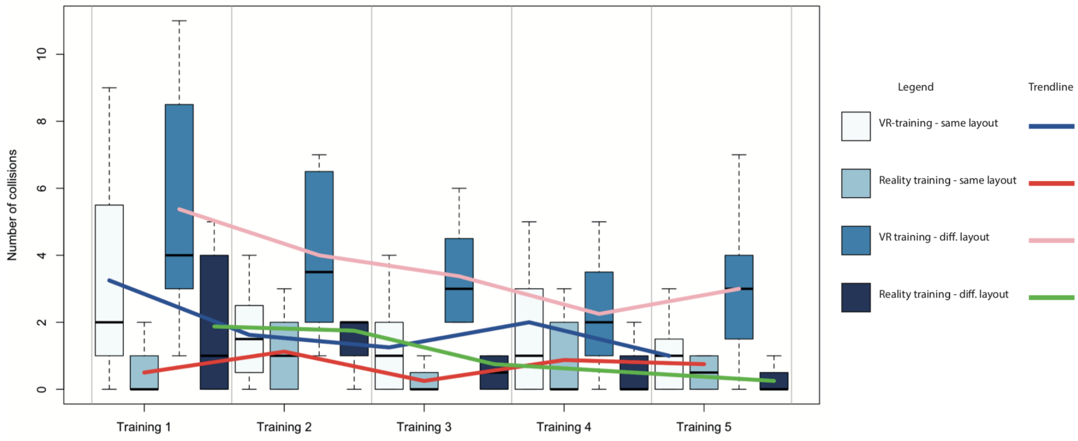


Figure 25: Number of collisions in each training session between groups

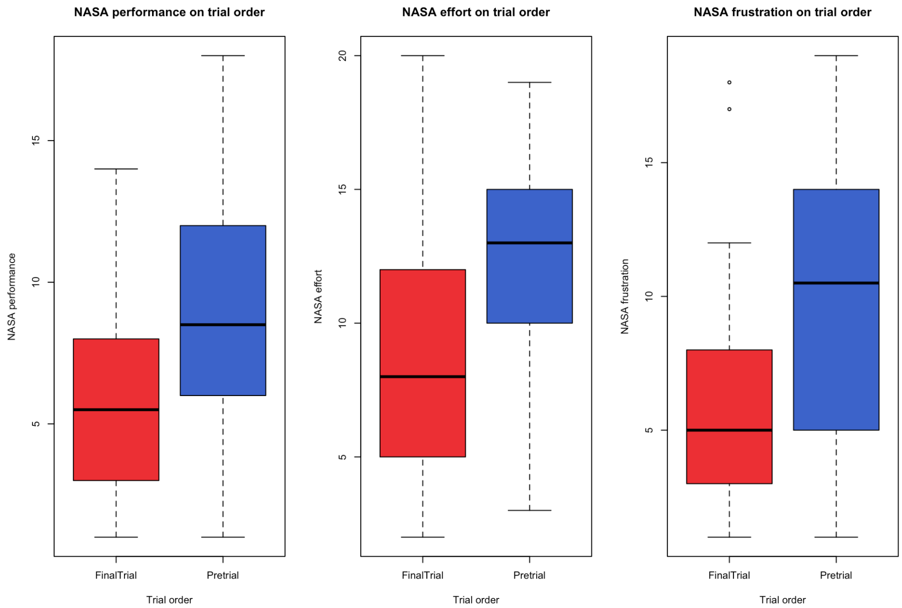


Figure 26: NASA TLX - illustrates the perceived workload performance, effort and frustration.

5.2.3 Situational awareness

SPAM results: Using a Chi-square test no significant main effects of the correct answers in the SPAM-queries was found on *training environment*, *maze layout* or *trial order*.

SPAM response time: The participants response time on the SPAM-queries was measured and analyzed. With a three-way ANOVA, we found no significant main effects of response time on the comprehension- and projection related questions on any of the independent variables or their interactions effects. With a three-way ANOVA a significant main effect of the response time on the perception related question was found on *trial order* ($F = (1, 50) = 5.19, p < 0.05, \eta^2 = 0.091$. The mean of the pretrial being 5.97 s(SD=4.46 s) and final trial 9.45 s(SD=6.28 s). It shows that the response time was slower on the final trial than the pretrial.

Preliminary question: The residuals of the average response time on the preliminary question ("are you ready to answer a question") was not normally distributed, and the significance was not the same after data transformation. Thus, we corrected the data using the box-cox transformation with $\lambda = -1.2$. Using a three-way ANOVA on the transformed dataset we found a significant main effect of the average response time on *trial order* $F(1,50) = 19.94, p < 0.001, \eta^2 = 0.215$.

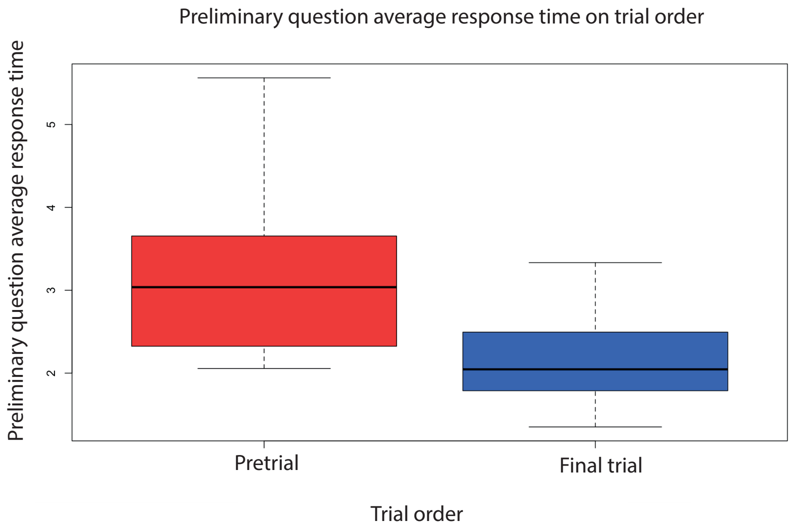


Figure 27: The plot shows how the test subject reaction time decreased on the preliminary question from the pretrial to the final trial

5.2.4 Self assessment - SAM

To see the difference in response of the self assessment manikin after the first and final trial, three Mann-Whitney's U test was conducted, one for each parameter. The test showed no significant main effects of self assessed arousal between trials on any of the independent variables or their interaction effects.

Pleasure: With the Mann-Whitney's U test a significant main effect of self assessed pleasure on the *training environment* was found (The mean ranks of VR-training and reality training were 27.52 and 37.48, respectively; $U = 670, Z = 2.2, p < 0.05, r = 0.275$). The medians of VR-training and reality training were both 4. No significant main effects of self assessed pleasure on *trial order* and *maze layout* was found.

Dominance: With the Mann-Whitney's U test a significant main effect of self assessed dominance on *trial order* was found (The mean ranks of the pretrial and final were 25.95 and 39.05, respectively; $U = 300, Z = -3, p < 0.01, r = 0.375$). The medians of the pretrial and final trial were 3 and 4, respectively. No significant main effects of self assessed dominance on *training environment* or *maze layout* was found.

Self-assessment after the training session: The self-assessment of dominance after the training session was analyzed and with the Mann-Whitney's U test a significant main effects of self assessed dominance on *maze layout* was found, the medians of same maze layout and different maze layout was 4 and 3, respectively; (The mean ranks of same maze layout and different maze layout was 20.34 and 12.65 respectively; $U = 66.5, Z = -2.512, p < 0.05, r = 0.444$). No significant main effect of self assessed dominance was found on training environment. Which shows that test subjects having their pretrial in the same maze layout as their training session reported a higher feeling of dominance.

5.2.5 Recollection

The recollection is based on the answers the test subjects provided in the post-trial interviews. After each trial the participants estimated the overall duration of the trial and the number of collision. Their answers was compared with the real number.

Duration - Estimated duration: With a three-way ANOVA, no significant main effects of duration (|estimated-real|) was found on *maze layout*, *training environment*, *trial order* or any of the interaction effects. This indicates that independent of the *training environment* the test subjects time perception was the same.

Number of collisions - estimated number of collisions: The number of collisions and estimated collisions, |estimated-real|, was compared in the same manner, and with a three-way ANOVA a significant main effect of the difference was found on *trial order* ($F(1, 58) = 11.996, p < 0.01, \eta^2 = 0.165$). The mean of the difference in the estimation of collisions compared to the actual number was for the pretrial 2.97(SD=2.89) and final trial 1.04(SD=1.18) .

Estimated confidence: Finally the participants confidence was evaluated using a Mann-Whitney U-test, since the normality assumption was not met either before or after data transformation. A significant main effect of estimated confidence was found on *trial order*, with median of the pretrial and final trial being 3 and 4, respectively (the mean rank for the pretrial and final trial was 24.05 and 40.95, respectively; $U = 210, Z = -3.8, p < 0.001, r = 0.475$)

5.2.6 Saccade tests

With a three-way ANOVA no significant main effects was found of *trial order*, *maze layout* and *training environment* or any of their interactions on saccade amplitudes, latency or number of successful trials.

5.2.7 User comments

Based on the post-trial interview the most notable and frequent comments from the test subjects was gathered. The comments was groups into basic categories:
Technical: *"Felt like the calibration got worse over time", "Biggest issue is delay/latency of video stream, resulting in over steering", "Bad resolution, couldn't see anything".*
Robot control: *"Hard to know when driving and turning the robot, no feedback", "Impossible to hit parking button", "Robot and room size would have been nice to know", "Would have liked a more thoroughly explanation of the control functions as what areas to turn, and the buttons explained, the training could to be about introducing controls as a game", "Control grid was big,*

far from one side to the other", "Hard to gain control many disruptions".
Eye input *"Would like the middle of the screen to have a neutral zone, it should be easier to look around, without controlling the robot", "Eyes became tired as a result of over starring"*
VR experience *"Felt dizzy after 5 minutes"(ended up getting really sick), "VR-training was very long and tiresome, it is a very long time wearing VR-glasses".*

5.3 8 best and 8 worst performers

In this subsection the best and worst performers based on robot operation performance was divided into two groups. This was done to see if the results of the saccade test could be used to create this division in robot performance and how the results varied between good and bad robot performers.

The grouping was done by taking the 8 best and 8 worst robot performers, based on each participants task completion time and average number of collisions in robot operation. To compare these two values, that are based on different measurements, they were normalized. This was done by doing a min-max feature scaling (normalization) on the data, so each value was between 1 and 0. The following equation was used:

$$Value\_normalized = \frac{Value - MIN(value)}{MAX(value) - MIN(value)} \tag{2}$$

This section only presents the relevant results, but all the results of the grouped data analysis can be found in C.1.

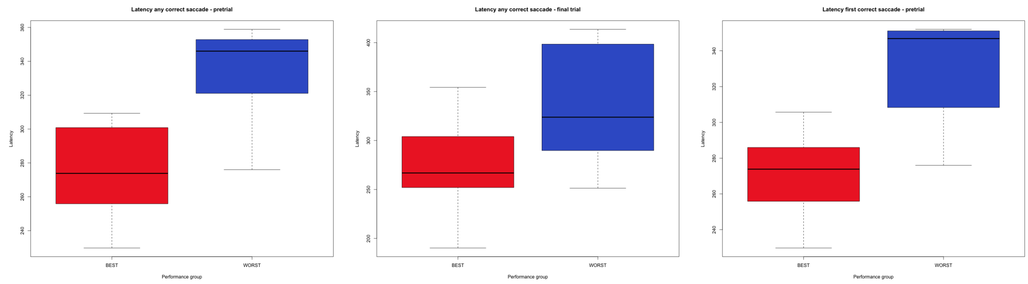
5.3.1 Saccade test

Number of correct saccades: With a one-way ANOVA followed-up by a pairwise comparisons with a Bonferroni correction, no difference in number of correct pro-saccades between the best and the worst robot performers was found.

Latency: Data transformation was used since the residuals of the latency of any correct saccade on both the pretrial and the final trial was not normally distributed. The lambda values for the pretrial and the final trial was so different (-0,9 and -2,4) on the same measured parameter, that a non parametric Mann-Whitney U-test was used instead.

With a Mann-Whitney U-test a significant main effect of the Latency of any correct saccade for both the pre- and final trial was found between groups. In the pretrial the mean ranks for the best and worst performers was 5 and 12, respectively; $U = 4, Z = -2.7775, p < 0.01, r = 0.6943$, the medians of best and worst performers, was 273.84 ms and 346 ms, respectively; see figure 28a. In the final trial the mean rank for the best and worst performers was 6 and 11 respectively, $U = 12, Z = -2.1004, p < 0.05, r = 0.5251$) the medians for best and worst performers was 266.85 ms and 323.83 ms, respectively. These findings indicate that the best performers had shorter reaction times on the saccade test during the entire experiment. The results can be seen in see figure 28.

Lastly using a Mann-Whitney U-test there was found a significant main effect of the latency first of the first correct saccade in the pretrial on performance group. The mean rank for the best and



(a) The latency of any correct saccade on the pretrial (b) The latency of any correct saccade on the final trial (c) The latency of the first correct saccade on the final trial

Figure 28: Latency

worst were 4.125 and 12.18, respectively, $U = 2.5, Z = -2.95, p < 0.01, r = 0.73$ The median for the best and worst performers was 273.8 ms and 346.8 ms, respectively; see figure 28c.

5.3.2 NASA:

Effort: With a one-way ANOVA followed-up by a pairwise comparison with a Bonferroni correction a significant main effect of perceived effort after the pretrial was found between groups $F(1, 14) = 5.141, p < 0.05, \eta^2 = 0,2686$, see figure 29.

Physical: With a one-way ANOVA followed-up by a pairwise comparison with a Bonferroni correction a significant main effect of perceived physical demand after the training session and after the final trial was found between groups, respectively $F(1, 14) = 6.13, p < 0.05, \eta^2 = 0.304$ and $F(1, 14) = 11.66, p < 0.01, \eta^2 = 0.454$, see figure 30.

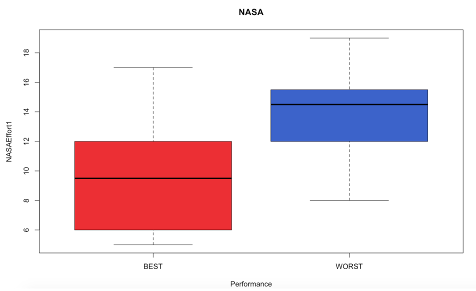


Figure 29: Pretrial: The plot shows that the best group experienced less effort driving the robot in the pretrial

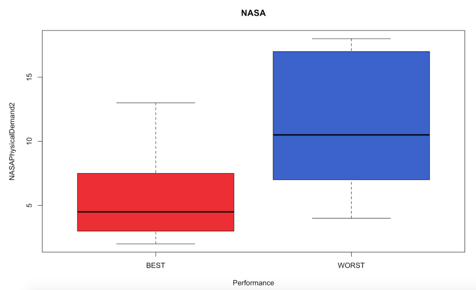


Figure 30: Training: the best performers experienced a lower physical demand during training and after the final trial (graph is during training)

5.3.3 Situational Awareness:

With a one-way ANOVA followed-up by a pairwise comparison with a Bonferroni correction a significant main effect of the test subjects reaction time on the second projection related query was found between groups $F(1, 9) = 6.589, p =< 0.05, \eta^2 = 0.351$ (the mean was for best and worst respectively, 7.57 s (SD=2.28 s) and 12.05 s (SD = 5.42 s)).

With a one-way ANOVA followed-up by a pairwise comparison with a Bonferroni correction a significant main effect of the test subjects reaction to the preliminary question for the first projection related question was found between groups $F(1, 9) = 6.163, p =< 0.05, \eta^2 = 0.406$ (the mean for the best and worst respectively, 1.83 s(SD=0.45 s) and 9.84 s(SD=15.65 s)).

5.3.4 Self assessment

Using a Mann-Whitney U-test between the best and the worst robot performers on the participants self assessment significant main effects was only found on pleasure after the pre- and final trial. No significant main effects was found for self-assessed arousal or dominance between the best and worst robot performers. The statistical results can be found in appendix C.1.

5.4 Summary data analysis

This section 5 presents the procedure of how the data analysis was carried out. The data analysis presents the most important results of the analysis and all the results of the analysis can be found in appendix C.1. The analysis was divided into two parts. The first part is a general data analysis which included all 32 test participants and the second part is an analysis between the 8 best and 8 worst robot performers. The subdivision was created to see if the pro-saccade test results could be used to group bad and good robot performers. The main results will be explained and discussed in the discussion section.

6 Discussion

This section presents the discussion of the results from the experiment and how it relates to the prior research presented in the first part of the report. The discussion will be divided into two subsections. The first part will discuss the results of the data analysis from all 32 test subjects and the second part will discuss the results of the analysis about the best vs. the worst performers mainly focusing on the results of the pro-saccade test. In the end the limitations associated with conducting an experiment like this is presented.

6.1 General discussion

The results from the experiment showed that *training environments* did not affect the task completion time, neither did the *maze layout* even though it was self assessed by the test subjects that training in a different *maze layout* reduced their feeling of dominance when operating the telepresence robot. Thus, the test participants felt less in control when operating the robot in unfamiliar environments even though this was not the case with task completion time as a measure.

It was shown that all the performance related measures improved between trials, showing that there was a learning effect as a result of training. Additionally the perceived workload between trials related to effort, frustration and performance all decreased.

The only performance related difference between *training environments* was the number of collisions. The test subjects who trained in VR had a higher number of collisions than the test subjects who trained in reality, see figure 23. This could be an effect of training in a virtual environment. When training in a virtual environment, the consequences of a collision is not as serious as with the real robot. This could both be positive and negative, since it might make the subject more curious(risk-willing), but it might also cause the effect of becoming more of a video-game than actually a replacement for training in reality. This might have been influenced by some of the errors in the virtual environment mentioned in section 4.7. The errors also shows exactly what virtual environments could be used for. The fact that the controls varied during the VR-training, and did not show any difference in task completion time in the final trial between the groups trained in different *training environments*, suggests that VR-environments could be ideal to create prototypes in.

The test subjects also performed (task completion time and collisions) significantly worse in virtual reality than in reality when trained with the robot. But the learning curves (task completion time and collisions) during the training sessions decreased in all groups and followed a very similar structure, see figures 24 and 25.

The pleasure of operating the telepresence robot in virtual reality was also significantly lower than in reality. This could be due to the longer training sessions in virtual reality, and thus longer time wearing the HMD, or it could be an effect of the amount of presence the test subjects felt. The training sessions was rarely interrupted and lasted around 30-60 seconds more for each trial. Thus, the participants ended up spending more time wearing the HMD. This could be one of the reasons that participants trained in VR had a higher tendency to report cybersickness symptoms than the participants trained in reality. In the end the two different training environments was rated as being equally exciting.

The test subjects confidence and feeling of control(dominance) both increased from the pretrial to the final trial. This could be seen by the actual number of collisions compared to the reported number of collisions between trials. The test subjects had a tendency to estimate a higher number of collisions than the actual number of collisions in the first trial, this was not the case in the second trial. This could in general indicate that the test subject got more aware of their surrounding with training, suggesting a learning effect happens where the test subjects got better at *perceiving* the information they received.

The primary finding in the above paragraphs is that a virtual environment can be used to train robot operators, even though the VR environment resulted in more time wearing the HMD, lower pleasure and an increased number collisions. Future work could therefore be in the development of virtual training environments that simulates extreme situations (emergencies etc.).

The fact that the trial order did not have any effect on the saccade performance (correct saccades) for all test participants (no learning effect), suggest that robot performance is independent of the saccade performance and that the test will provide the same independent result for the test participant given the same situation. The pro-saccade tests was activated at the same location in the maze layout in both trials. This suggest that saccade performance does not vary when the test is done under the same circumstances.

The situational awareness queries did not show any difference in the answers between groups, but the response time for the perception related query did change between trials. The response time actually became slower, this is probably due to the difference in the perception questions structure (A.5). In the pretrial the question was a yes/no question to if the user had seen someone in the room, where in the last trial the participant had to give the direction they were facing (north/south/east/west). The two question vary a lot in difficulty, and thus the results of this should be neglected.

The fact that no significance was found between the groups in the SA queries nor the saccade test could have mixed importance. If a significance between the groups on either one of the measures but not on the other was found, it would be more certain that there was no correlation between the two. It was expected to see a difference in SA between the pre- and final trial, as it was expected that the test subject would become better robot operators and therefore obtain a higher level of SA. This missing result could be explained in the difference in the questions asked in the SA queries and the sources of errors regarding the SPAM-queries mentioned in the experiment section. It was shown that the average response time on the preliminary question in average reduced significantly between the trials. This indicated that the perceived workload reduced with training, based on the SPAM measures, see figure 27. This is backed up by the test subjects indication of increased perceived performance, decreased effort and frustration from the pretrial to the final trial, see figure 26.

6.2 Discussion 8 best vs. 8 worst performers

The results of the saccade test showed that the best performers had shorter latency on any of their correct saccades on both trials. This result indicates that the best robot performers had a higher alertness to any surprising stimuli such as a pro-saccade test, see figure 28. It was also found that the first correct saccade also was significantly quicker for the best robot performers. This could be argued to provide the same measure of workload as the SPAM-preliminary question. This is partly backed up by the response time on the preliminary question to the 3. projection related query. Here the best performers was quicker at answering the suddenly appearing question. This again suggest that the best performers experienced a lower workload than the worst performers. The significant difference in reaction time to the preliminary question could be explained much like the latency of the saccades was explained, simply that the best performers had an excess of mental capacity to register that an object appearing in front of them, and therefore were faster at responding. This is backed up by method of how to measure workload in the SPAM query [40]. Furthermore the increased perceived physical demand and effort for the worst group indicate that they also had a harder time operating the robot. The extra perceived physical demand could also be a result of operating the robot for a longer duration possibly resulting in tired eyes (fatigue). This would also effect how fast they were able to react to some task that did not have any thing to do with the primary driving task.

We saw no significant effect between the best and worst robot performers in their ability to perform a saccade ("number of correct trials") this result further backs up the results presented in the general discussion, that the saccade performance is not affected by a learning effect or the robot performance.

The experiment showed that it is possible to obtain results from a *forced* pop-up pro-saccade test where the test subjects main task is paused without causing breaks in presence. This principle could potentially be used when monitoring peoples mental state when operating telepresence robots with their eyes.

The test showed that the best performers had a faster reaction time on the second projection related question, indicating that they obtained a higher level of SA. This is also backed up by the best performers indicating a lower physical demand operating the robot. The best performers also experienced more pleasure driving the robot during the entire experiment.

From the plots provided by the SaccadeMachine software we could see a clear tendency that the worst performers had a more *flickering* eye movement during the saccade test. This observation was very hard to quantify but could be a subject for further investigation. See figure 31, for an visual explanation.

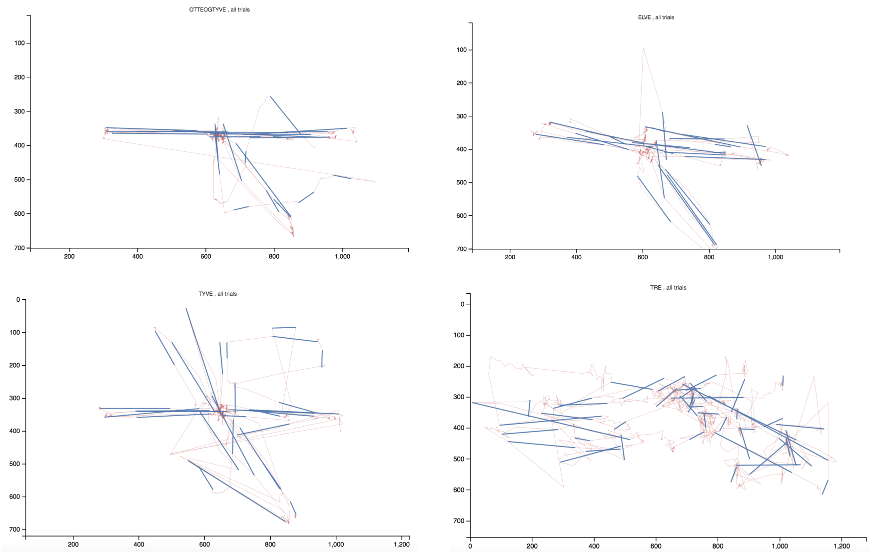


Figure 31: Eye movements in the saccade test. The top plots are from the best robot performance group, and the bottom from the worst

6.3 Limitations of study/results

This study and the results has its limitations which is important to take into consideration before conducting studies based on it or studies alike.

The experiment was a lab based experiment, which is a good way to simulate real use scenarios and makes the study easy to replicate. A limitation to lab-based studies is the test settings. The fact that test subjects are being observed could result in unnatural behaviour that would not directly reflect user behaviour in a real life settings [30]. The number of test participants in the study was only 32 and does not necessarily represent a specific target group. Besides this the test participants was all able-bodied and the target user group is ALS-patients or other patients with motor disabilities. Thus, user studies in their saccadic behavior would have to be conducted before implementing a test in a robot system designed for this group.

The number of studies conducted on gaze controlled telepresence is very limited. Thus, most of the literature gathered has been in research areas closely related to the one of gaze controlled telepresence robots.

The literature about saccade tests, and how it previously has been used is all about, more static test in simulators or just saccade tests on a normal computer screen. It is not certain that the theory related to these less dynamic test, reflects the same as our test even though some of the results might indicate that. We were not able to find a saccade test conducted while users wear a HMD, which created a challenge for us in the implementation, as we didn't have any literature to base our approach on.

Finally the experiment has its sources of errors as listed in the section 4.7. These was related to the experimenters involvement in the experiment. This could have been avoided by creating an alternative way of testing and gathering data about the aspects or adding a autonomous aspect.

6.4 Summary discussion

The data analysis showed that there was no difference in task completion time between training environments only the number of collisions was higher for participants trained in VR than reality. The VR-environment also resulted in less pleasure and longer duration of the training. The longer duration was due to different controls in VR than reality. Even with different controls in training the groups still performed and improved equally. Based on these results we suggest that a virtual training environment theoretically could replace reality training when operating a telepresence robot using gaze, if implemented correctly. But more important virtual environments could be used for prototyping. Often a slight change in functions will have functional requirements to the robot, such as sensors and actuators, this could be tested and implemented much easier in a virtual environment. This allow for testing of new interfaces, new functions etc. almost any aspect of the telepresence robot could be tested.

It was observed that neither training environment, trial order or the test subjects ability to operate the robot had an influence on how well a person was able to perform a saccade. This suggest that under the same conditions a persons pro-saccade performance would be the same. This can potentially be used to create a *baseline* for the operator, so any deviation from the baseline within a threshold could indicate some changes in the the environment/situation.

Based on the results from the data analysis it was not possible to find a clear correlation between situational awareness and pro-saccade test results. However it is not possible to reject that they are not connected.

By grouping the data into the best and worst performers it was observed that there was a clear difference between the groups latency on their saccades, indicating a higher alertness for the best performers. The results also indicate that the best performers felt a lower workload than the worst performers along with a lower physical demand. These findings suggest that it is possible to evaluate something about robot operators performance based on their eye movements.

7 Concept development

7.1 Concept description

Based on the empirical research and the results from the experiment 5 concepts was developed. The 3 first concepts are based on the idea of implementing a saccade test (both active and passive) to measure different aspects about a robot operator. The 4. concepts is based on the primary finding from the experiment regarding the use of virtual environments to train robot operators. The final concept is an idea that are inspired by some of the user comments in the experiment and the prior work section (2).

7.1.1 Passive test

Concept description: The passive test should be a completely none invasive test that runs constantly in the background, measuring the users eye movements isolated and in relation to other elements in the interface, see figure 32. It would in real time determine when a user was performing a saccade and the amplitude of that saccade. If messages, warnings and other notifications appeared in the interface, the test would determine the latency from the moment the notification appeared to the saccade was performed and how *correct* is was performed. This data would be analyzed in real time by an algorithm that would compare the current data stream with the user's previous data. This way the algorithm could update the individual user's threshold for the amplitude, latency and correctness of their saccades, in relation to a predetermined baseline.

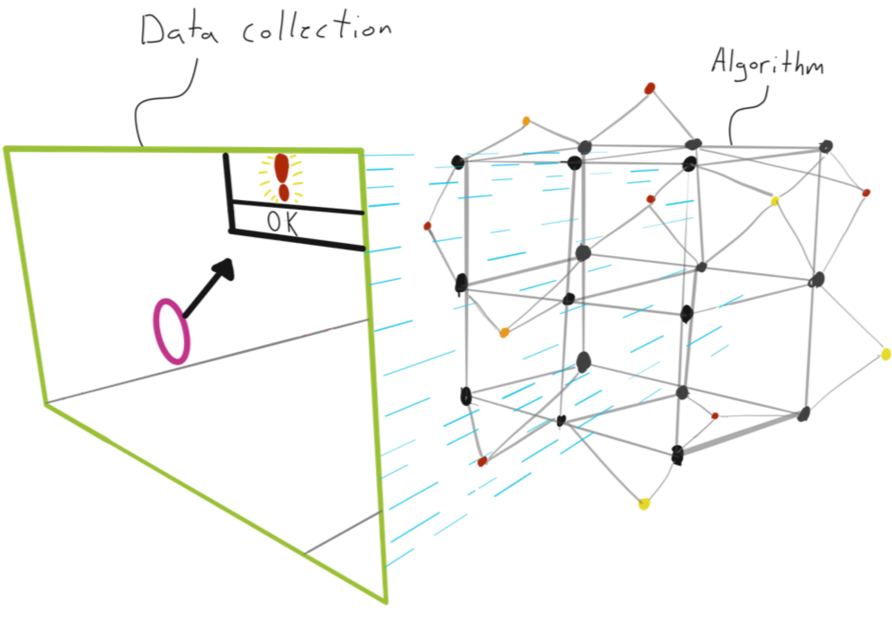
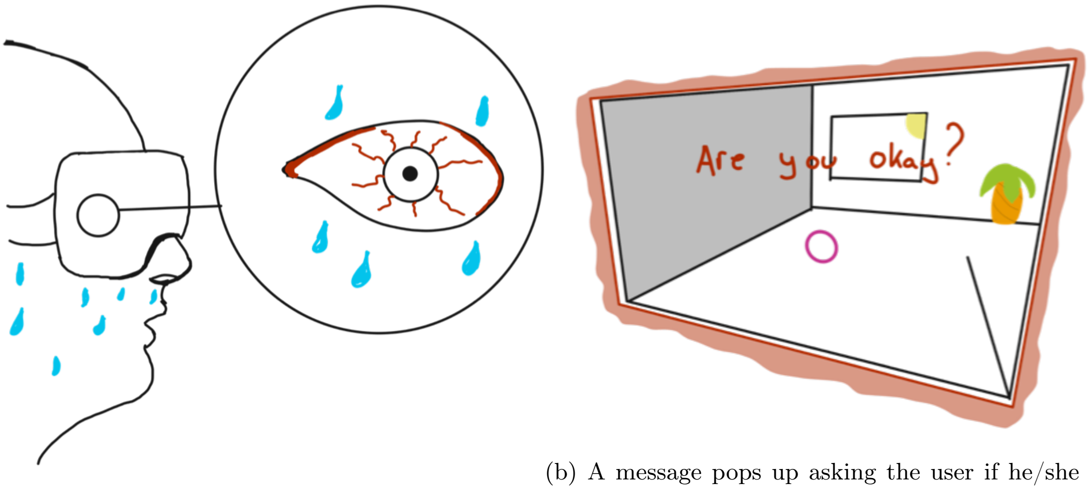


Figure 32: Passive test measurement example: A warning appears as a target in a pro-saccade test. The results of the saccade made towards the warning is analyzed.

This concept could potentially be used when ALS patients operates the robot and are feeling distressed. The algorithm would be able to alert assistance that something was wrong.



(a) The user are in distress

(b) A message pops up asking the user if he/she is okay

Figure 33: Passive test that constantly monitors the operators eye movement patterns and search for deviations

Arguments: The passive test would use the results from the experiment, where it was shown that the best performers had a longer latency when a target appeared onset. As the passive test is not an active test with a target appearing on-set some other way of measuring the operators saccade latency should be used. This could be done when notifications or messages appeared in the interface. Here it would be possible to know exactly when the notification appeared onset and when the operator performed a saccade towards the target. Here the operators alertness would be analyzed, see figure 32.

The focus in this project have been on saccadic eye movements but the empirical research showed that other different eye movements also could provide insight into the user's mental state. Measurement of eye movements such as blinks and pupil size could also be implemented in the test to give the algorithm more data streams to evaluate the user. This will be further discussed in the future work section 8.

A number of users indicated that the saccade test and the SPAM - queries was disturbing for the driving task. This concept would eliminate this.

Use case: 1) The operator is driving around with the robot.

2) Their eye movements are being measured. If some notification appears the latency from the notification appear to the operator performs a saccade is measured. further more other eye movement such as the amplitude of the operators saccades, blinks, pupil size etc. are measured and analyzed in by the algorithm, see figure 32

3) If the algorithm detects that the operator is experiencing a higher workload or are less alert the operator is asked if they are okay, and maybe should take a break, see figure 33.

Strengths: This concept has potential to be a non-invasive test to measure the mental state of the user. This could be a future alternative to the more invasive test such as the SPAM queries used in the experiment or a post-trial interview, that will cause breaks in presence.

As new aspects about the human mind gets discovered this could continuously be used to improve the algorithm and thereby make better estimations of the users mental state. Furthermore, the continuous improvements within machine learning etc. could further enhance the accuracy of the test.

Weaknesses: The conclusions from the experiment show promising result for this concept to function. With that said more research needs to be conducted to verify the findings.

If the concept gets implemented a series of ethical issues needs to be addressed. By constantly measuring the user the action could be perceived as surveillance, especially in an employment situation.

7.1.2 Prolonged calibration

Concept description: After taking on the HMD the user needs to go through a calibration process where the software registers the position of their eyes. This process takes around 20-120 seconds depending on the users experience with the calibration process. The calibration process is in many ways much like a pro saccade test where the user follows a dot around on the screen. The concept is therefore a hidden saccade test immediately after the user has finished the calibration.

This *prolonged calibration* - saccade test will be looking at the same parameters as the pro-saccade test conducted in the experiment.

The prolonged calibration could also be used as an initial assessment of the users ability to drive the robot. This principle has been used in accessing elder's inability to drive a car by comparing their anti saccade performance with their driving abilities [35]. This indicated a high correlation between bad drivers and performers on the anti-saccade test.

Arguments: From the experiment it was found that it was possible to get results about the alertness of a user from a active pro-saccade test. This was seen in how the latency of the saccades varied with driving performance. This measure could be one of the parameters determining the users ability to drive.

The experiment also indicated that under the same conditions the performance on a pro-saccade test would be constant. The calibration process would be expected to be the same every time. Thus, if a deviation in the user's saccade was registered, some other factor might be the cause. This aspect will be discussed in the future work section 8.

Use case:

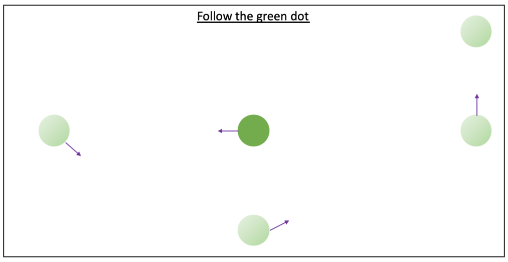
1) The user takes on the HMD.

2) A calibration process is initiated that will determine the position of the user's eyes, see figure 34b.

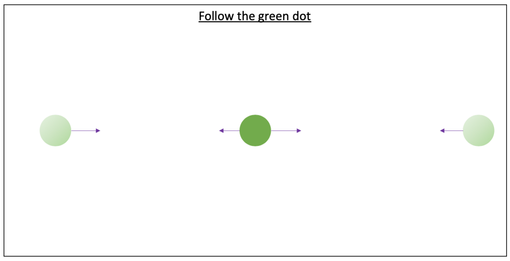
3) The user will perform a pro-saccade test without they know it - as they will think it is still the calibration, see figure 34b.

4) If the user do not show any signs of inability to drive, he/she gets access to the robot. If he/she shows sign of inability to drive they are ask to wait a couple of minutes or some assistance is called for, see figure 34c and 34d.

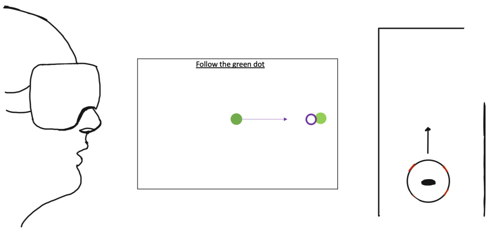
Strengths: The concept works as an active test, but is still hidden, so the user does not know they are being measured. This makes it less invasive for the user. A pro-saccade test is relative



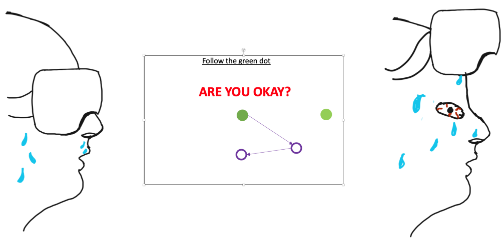
(a) User is being calibrated



(b) The user performs a pro-saccade test



(c) If the saccade test did not show any deviations



(d) If the saccade test shows deviations

Figure 34: The use scenario of the prolonged calibration

simple in nature, therefore it is not that complex to implement in a calibration system. The prolonged calibration can be used as a baseline for the *passive test*.

Weaknesses: The concept will *prolong* the calibration for the user, which can be seen as an annoying factor.

As in the passive test the measurement of an individual can cause a series of ethical issues that needs to be addressed, especially in a working situation.

More research needs to be conducted to verify the findings from the result. These will be discussed in the future work section.

7.1.3 Active pro-saccade test in robot operation

Concept description:

The operator would be given a pro-saccade test if they show significant deviation from their average performance when controlling the robot. To monitor the operators performance in robot operation, the robot would need to be equipped with ultrasonic sensors. These would measure, the distance to obstacles in the environment and if any collision might happen. It would also be possible to measure how much the operator is *wobbling* around by looking at how the wheels are turning on the robot. Besides this the users confidence level could be measured in how fast they were driving the robot and if the driver for example often changed between driving really fast to really slow in a straight line. If the operators driving performance deviated a lot from some predetermined baseline they would be given a pro-saccade test. The test would be looking at many of the same parameters as the one used in the experiment. The pro-saccade test would also be used as a recalibration of the user, as bad calibration often influences driving performance.

The change in robot performance or eye-movement patterns could be due to bad calibration, the eye-tracking quality or physical symptoms such as fatigue. The pro-saccade test would therefore not only be used as a way to measure the users mental state as done in the experiment, but also as a way the recalibrate the user. Wearing a HMD for a longer period can be tiring for the users eyes, heavy for the neck and in general be uncomfortable to wear. Thus, users tend to adjust the HMD during robot operation, which results in calibration offsets. A good indication of poor calibration was to look at the number of times the operators gazed outside the *control input border* and back in again, see figure 13.

As an extra element to the concept the passive test-concept could be implemented as well. By looking at both the operators robot performance and their eye movements a more detailed picture of the users mental state could be formed. The active saccade test would here be used as a validation of the passive test and if the poor robot performance was connected to the users mental state or if was some outside factors.

Arguments:

The results from our experiments indicated a correlation between driving performance, workload and especially latency of saccades. Thus, we suggest that the concept could be used to track if the operator needs a break in robot operation. By combining the pro saccade with a re calibration process assures that the user is not performing worse because of the calibration. Bad calibration was one of the most frequent comments from test subject when asked what reduced their performance.

Use case:

1) The user are being measured on their driving performance and potential different eye movements.

2) If the users driving performance seems poor, which indicate that they are experiencing high workload, the user is asked to recalibrate, where their mental state are measured using a pro-saccade test, see figure 35.

3) If the user passes the test by not exceeding some predetermined threshold on the pro-saccade test they can continue driving. If they fail they are asked to take a break, see figure 36.

Strengths: The measurement would most of the time be non invasive, and when the test gets invasive the user will only believe they are recalibrating, which is in the users interest as well.

Weaknesses: By measuring the driving performance, there are a risk that the test would appear in unwanted situations. Imagine the case where a perfectly capable operators working on highly complex task, for example driving through narrow hallways avoiding hitting different obstacles. The measure of the individuals driving performance could unnecessarily suggest that they should consider taking a break. Therefore a "snooze" function should be implemented to allow the user to bypass the test. This could be the user pushing a button by dwelling on it or maybe perform a saccade to a specific target.

7.1.4 VR-tutorial:

Concept description A tutorial in a virtual environment that allows the user to operate the robot in all sorts of terrains, both realistic operation environments and very extreme environments.

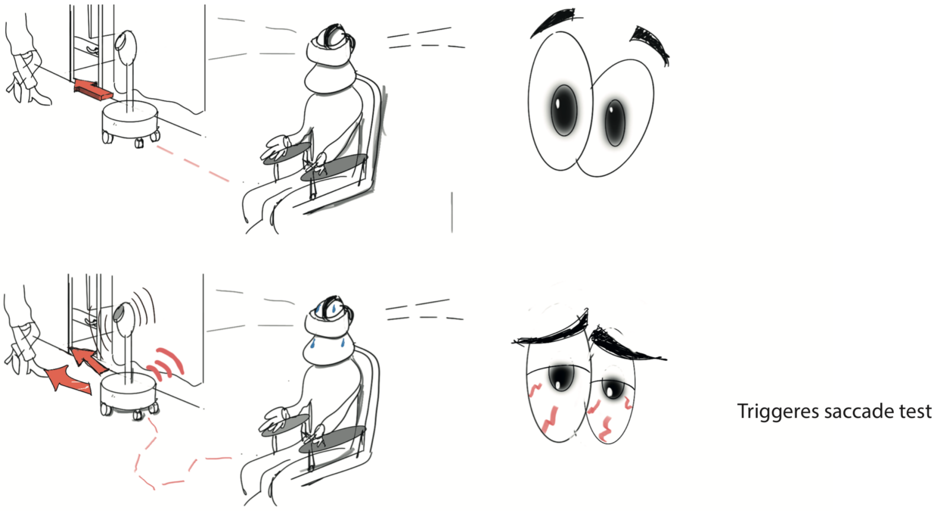


Figure 35: Variations in the operators robot performance, might be a sign of a bad calibration or a sign that something is wrong

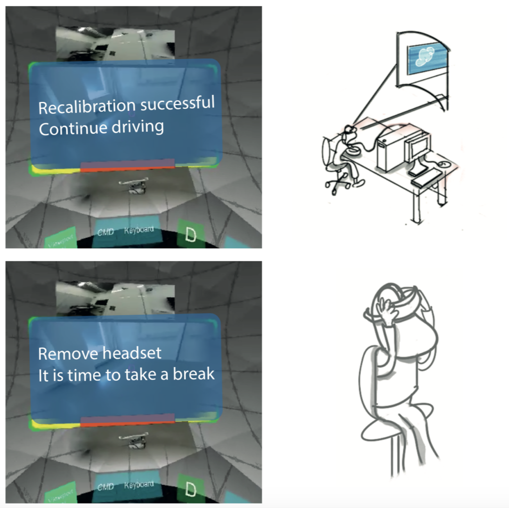


Figure 36: Depending on the results of the pro-saccade test, the user will be recalibrated (top) or asked to take a break (bottom)

The tutorial would be interactive and provide an overall introduction to the control mapping in the interface, and allow users to test different interfaces out. The VR-tutorial could incorporate more than just the basic control, an introduction to the keyboard for communication and a virtual person that interacts with the user in the tutorial, could make it more natural for the operator to interact with people through a telepresence robot, see figure 37 for an illustration of the tutorial.

After a introduction of the basic controls and functions of the robot a normal driving scenario of the robot can be chosen. This will challenge the operator in different ways.

The tutorial should be skippable and accessible at all times during robot operation. Thus, in need of a specific function it could be found in a menu. This should also be done not to force the user to go through every part of the tutorial, if only a certain control is of interest.

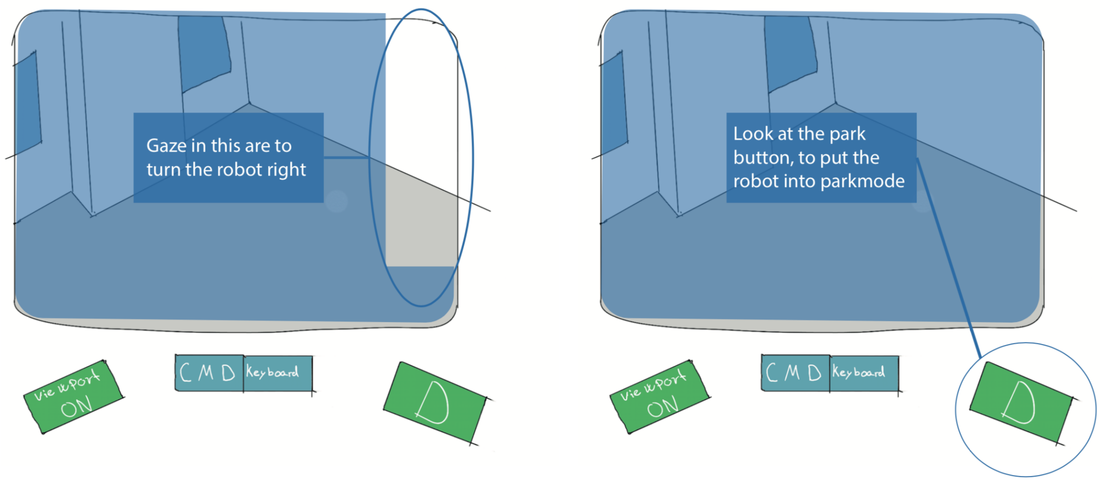


Figure 37: VR interactive tutorial explaining the functions and control mapping

Arguments: The results from the experiment showed that training environments did not affect the performance, and that test participants trained in both environments, had decreasing learning curves of task completion time and number of collisions during training. The training in general was not very instructive, the operators, was just told to do a simple task 5 times. The test subjects only received information regarding controls before their pretrial and no instructions in how to operate the robot was given during the training session. Since the results show that the VR-environment can be used to train robot operators, we suggest that the training could be like a game-tutorial. Some test subjects also argued in the post trial-interview, that they would have liked a more thoroughly explanation of the controls.

We suggested that the results of a pro-saccade test could be dependent on the situation that the operator is in. VR-environments could be ideal to test if this claim holds, we could design different driving situations of varying difficulty in VR and use the pro-saccade test to compare the users performance. The use of mentally demanding situations in a tutorial, could also provide us a with more data to analysis the change in eye-movement patterns in different work situations and difficult operation environments by implementing a test like the passive test suggested in 7.1.1.

By creating an interactive tutorial in the VR-environments, the control mapping and the buttons in the interface could be introduced and used in a specific relevant context to make the user feel safe about the functions. Creating an interactive tutorial instead of a cinematic explaining the controls, will provide better feedback to whether or not the user understands the tutorial and control. Allowing the user to interact with the environment during a tutorial also provides feedback to the user about how the controls work in the physical environment.

Use case:

1) The first part of the tutorial explains the basics controls of the robots interface, see figure 37.

2) After an introduction to the basic controls, the user can drive the robot in different scenarios of various diffucility, see figure 38.

Strengths: Having a virtual environment that could reflect real robot operation would shorten the testing process. Not having to implement all the features needed for a test on the real

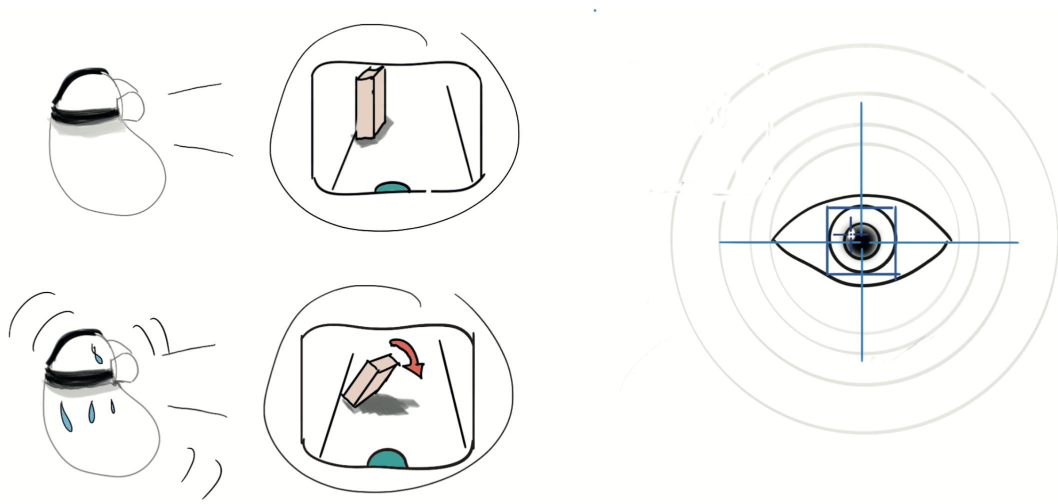


Figure 38: Use scenario - of how the users eye movement patterns could be tracked in two different situations, that vary in difficulty.

telepresence robot makes the VR-environments an ideal tool for prototyping and testing of new features.

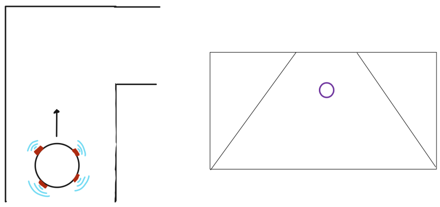
Secondly the environment could be used to gather much more data about the saccade tests, by implementing them is virtual robot operation situation of varying difficulty.

Weaknesses: A weakness is the quality of simulation, how well can we actually create a real work-scenario and is it the optimal way to do it? Would it be preferable that it was more like a game with virtual unrealistic characters, or should we try to recreate the real working environment with realistic representations of the simulated environment. This is a trade-off that needs to be considered.

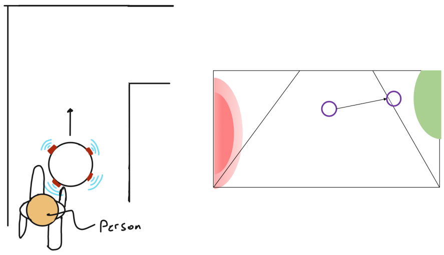
7.1.5 Saccade priming interface

Concept description: The interface will use targets of interest to prime the user to steer the robot in the correct direction. The concept requires that the telepresence robot is equipped with distance measuring sensors (ultrasonic sensors). The idea to give the operator an extra element of information or add a layer of autonomy when driving is an idea that have been suggested and implemented in previous concepts [10]. When the sensors registers an obstacle a *distractor* will be displayed in red on the interface that indicates in which direction the obstacle is located, see figure 39a. This would help the robot operator to gain an increased level of spatial awareness. The concept is thought of as the warning in a first person shooter game (were the screen lights up in direction you are shot from). In the mirror position of the interface a *target* will appear in green, suggesting that the user should look in that direction to avoid the obstacle. This way the user is primed to perform a saccade in the mirror position of the appearing *distractor*.

Arguments: One of the principles from the SA oriented design, is to only implement autonomy if necessary [11]. This concept allows the user to be in control of the robot instead of an automated system taking control of the robot. This increases the chances of the operator keeping their SA at



(a) The robot operator driving down a hallway with no distractions nearby



(b) A obstacle is detected by the robot and a *distractor* (red area) an *target* (green area) appears

Figure 39: Use case scenario of the Saccade priming interface. The robot can be seen from bird perspective on the left in both figures and a picture of the interface can be seen on the right

a higher level. The concept could help the users to feel more in control when navigating unfamiliar environments. In the post-trial interviews some test subjects said that it was hard for them to determine the size of objects in the environments when operating the robot, the feedback in the interface would give the user a better awareness of the dimensions of their surroundings.

The concept would have some requirements to the robot in terms of sensors. Instead of equipping a robot with these sensors just to test the interface, the concept could be implemented in a VR-environment to save time and money.

Use case:

1) The user drives around with the robot, see figure 39a.

2) When an obstacle appears the sensors register from which direction the obstacle is and displays a *distractor* as a warning in the interface and a *target* in the mirror position, see figure 39b.

3) The user is primed to perform a saccade in the opposite direction of the obstacle.

Strengths: The stimuli will function as a visual feedback for the user, indicating where a obstacle appears giving them a better awareness of their environment.

The concept will train peoples ability to user their eyes for robot control.

Weaknesses: The concept is only tested in game situations (shooting and racing games) the use is partly unknown in the proposed scenario.

8 Future work

This section suggests which aspects and measurements that should be further investigated to validate the findings of the project. Many aspects of the concepts also depends on measures that have not been examined in this project. This leads to the following recommendations for potential future work.

In the *passive test* (section 7.1.1) it is suggested that other measures could be used to determine the user's mental state. It is suggested to look at blink duration, blink rate and pupil size as these parameters have been shown to provide information about a users experienced workload and SA

[2, 1, 19, 49]. If this had to be implemented in the passive test, each of those measures would need to be tested. This should be done to see if the same conclusions drawn in each of the studies apply when operating a gaze controlled telepresence robot.

We observed no significant difference between the best and worst performers in the amplitude of their saccades. Research conducted by Cardona and Quevedo in [2] suggest that amplitude of a saccade might provide information about the workload a operator is experiencing. This could therefore be investigated in a dynamic test where the saccade will not be performed from one fixation to a target.

The results of the pro-saccade test did not vary between trials and was activated at the the same time in the same situation in each trial. Thus, it was suggested that the pro-saccade test has no learning effect and can be seen as a measure of the situation. Furthermore the results from the experiment did not show a direct correlation between an individuals pro-saccade test and SA. Future work could therefore be conducted with the primary focus on investigating this correlation. This could be done by giving different saccade test in a range of situations, along with other SA measures described in the SA section.

In the *passive test*, *prolonged calibration* and *Active pro-saccade* concepts it is interesting to know if the operator is experiencing fatigue or feeling sick. In the experiment we observed that the worst performers was feeling a higher physical demand. It could therefore be interesting to investigate if the bad performers showed any characteristics in their eye movement that could indicate if they were tired. In the discussion an observation was mentioned regarding the flickering eye movement for the worst performers during the pro-saccade test (see figure 31). This could be a interesting observation to further investigate.

9 Summary

This is a gathering of the short summary's that rounds of each section in the report. Thus, the content will be a repetition that lead to the conclusions in the next section.

Research question

The research question of this project was presented in section 1.1 was:

How does training environments affect operator performance, awareness and alertness in operating a gaze controlled telepresence robot? Could operator performance, awareness and alertness be evaluated without intervening in the operators primary task - using a saccade test?

How could a dynamical test be incorporated in the design of a new interface for a gaze controlled telepresence robot?

Prior research

Section 2 provides a review of the the empirical material that establishes the background knowledge for this project. The literature presented shows that it is possible to control a teleprensece robot using gaze as a control input. Gaze controlled telepresence robots could create big value to the lives of motor disabled persons. Allowing motor disabled to be physically embodied in a robot could help them experience events they would not usually attend due to accessibility problems (concerts, conferences etc.) and most importantly it could provide motor disabled with jobs as OryLab showed [3].

The use of gaze to control a telepresence robot is followed by a series of problems. One of the main recurrent problems in telepresence is the technical problems as lag, latency etc. resulting in bad image quality that leads to a reduced feeling of presence and a decrease in spatial awareness with the missing depth perception in the video image. Another problem presented is that navigating a regular teleprensence robot in unfamiliar environments often is very cognitively demanding, and that gaze control is rated as more cognitively demanding than the use of a traditional control inputs, but at the same time it serves as the only input method for some motor disabled. It was also shown that an ALS-patient could operate his wheelchair using gaze as a control input with the help of autonomous assistance - autonomous functions should still be implemented with care since it might reduce the operators spatial awareness and in the end SA. Another problem in the robot operation is that operators has to wear a HMD. Thus, measuring presence using classic measurements the user will have to take off the HMD to fill in a questionnaire, and risks the break in presence.

Based on the knowledge gathered about what different eye-movements can tell about a persons experienced workload, awareness and alertness, and the use of saccade tests, we suggest that a saccade test might be able to provide us with information regarding the presence and SA of the user, which was described as closely related. Thus, for the experiment a pop-up pro-saccade test have been implemented in the interface of the robot, that is very similar to the structure of the SPAM-queries.

Implementing pro-saccade test

Section 3 explains how a pro-saccade test was implemented in Unity. The problems and solutions associated with implementing a saccade on a 2D screen in a 3D environments is also explained. The main problems/errors of implementing a pro-saccade in the test setup used in this study was determining spatial measures of the saccade pop-up test in Unity and the low frame rate of 70 Hz of the FOVE HMD. Most of the measurements that needed to be collected during the pro-saccades could be collected through the FOVE-headset and Unity environment, such as the coordinates of the gaze position relative to the target position. The only measurement we needed to create outside the standard measures was a saccade detection algorithm. A simple dynamic saccade detection algorithm that detects the saccade based on previous gaze position was implemented and is presented in the section. A brief presentation of the SaccadeMachine software which was used to analyze the data gathered in the pro-saccade tests can be found in the section [29].

Experiment

Section 4 explains how the experiment was designed, it's procedure and how it was carried out. The aim of the experiment was to examine how virtual environments, familiarity of the test environment and trial order affected the robot performance and what a pop-up pro-saccade test during the robot operation task could say about the test subjects.

The experiment was a between-groups design. The experiment was conducted in DTU Skylabs DesignLab and 32 test participants was recruited. In the experiment the test subjects had to operate the gaze controlled telepresence robot through a maze. The experiment consisted of a pretrial, training session (5 trials) and a final trial. During the pre- and final trial the test subjects would receive 3 types of tasks (saccade test, SPAM-queires and person interaction). The experiment examined 3 independent variables: *Training environments*, *maze layout* and *trial order*. The dependent measures was: Task performance (taks completion time and number of collisions), workolad (NASA-TLX), situational awareness (SPAM), saccade test results (pop-up pro-saccade test), self-assessment (SAM) and recollection (estimated performance, user comments ect.).

The source of errors during the experiment was: The VR-environment (controls varied compared to the real robot), experiment conductor errors (some measurements was dependent on the conductors performance- SPAM-queries) and park-mode (eye tracking was poor in the lower part of the FOVE HMD - location of the park button).

Data analysis

Section 5 presents the procedure of how the data analysis was carried out. The data analysis presents the most important results of the analysis and all the results of the analysis can be found in appendix C.1. The analysis was divided into two parts. The first part is a general data analysis which included all 32 test participants and the second part is an analysis between the 8 best and 8 worst robot performers. The subdivision was created to see if the pro-saccade test results could be used to group bad and good robot performers.

Discussion

Section 6 presents a discussion on how the results of the experiment can be interpreted and what the limitations is to the study. The main findings of the data analysis and discussion is four elements:

- The data analysis showed that there was no difference in task completion time between training environments only the number of collisions was higher for participants trained in VR than reality. The VR-environment also resulted in less pleasure and longer duration of the training. The longer duration was due to different controls in VR than reality. Even with different controls in training the groups still performed and improved equally. Based on these results we suggest that a virtual training environment theoretically could replace reality training when operating a telepresence robot using gaze, if implemented correctly. But more important virtual environments could be used for prototyping. Often a slight change in functions will have functional requirements to the robot, such as sensors and actuators, this could be tested and implemented much easier in a virtual environment. This allow for testing of new interfaces, new functions etc. almost any aspect of the telepresence robot could be tested.

- It was observed that neither training environment, trial order or the test subject ability to operate the robot had an influence on how well a person was able to perform a saccade. This suggest that under the same conditions a persons pro-saccade performance would be the same. This can potentially be used to create a *baseline* for the operator, so any deviation from the baseline within a threshold could indicate some changes in the the environment/situation.

- Based on the results from the data analysis it was not possible to find a clear correlation between situational awareness and pro-saccade test results. However it is not possible to reject that they are not connected.

- By grouping the data into the best and worst performers it was observed that there was a clear difference between the groups latency on their saccades, indicating a higher alertness for the best performers. The results also indicate that the best performers felt a lower workload than the worst performers along with a lower physical demand. These finding suggest that it is possible to evaluate robot operators performance based on their eye movements.

10 Conclusion to research question

The results of the experiment shows that training in a virtual reality only affects the test subjects performance in terms of collisions when operating the robot after training. This was argued to be an effect of how the virtual environment was designed, but it might also be a negative effect of training in a virtual environment where collisions does not matter. The *maze layout* had no effect on the performance, but produces a feeling of less control when controlling the robot in unfamiliar environments. The data analysis showed no effect of *training environments* and *maze layout* influencing the awareness and alertness of the test subjects.

The pro-saccade test showed no difference in number of successful trials on performance, but an interesting finding was that saccade performance and saccade latency did not improve between trials. Another interesting finding was that the best robot performers had significantly lower saccade latencies than the worst robot performers, which indicates a higher alertness for the best robot performers. Based on these results we argued that the results of a pro-saccade test might be individual for each user and only depended on the situation and that the latencies of an individuals pro-saccade test can be used to group bad and good performers.

The results from the pro-saccade test showed interesting results. Thus, we recommend to further investigate how it can be used to assess workload in different situations. Based on the results from the experiment and the prior research several ways of implementing a dynamical test have been suggested, these should all be well tested since our results can not justify the implementation alone. For testing of dynamic saccade tests we suggest the use of a virtual environments, since the results showed no difference in performance regardless of training environment and it would be less comprehensive to setup an experiment in VR. Implementing a dynamic saccade test could be a great way to avoid the interfering with the operators task and to avoid breaks in presence.

References

[1] Per Bækgaard, Shahram Jalaliniya, and John Paulin Hansen. Pupillary measurement during an assembly task. *Applied ergonomics*, 75:99–107, 2019.

[2] Genís Cardona and Noa Quevedo. Blinking and driving: the influence of saccades and cognitive workload. *Current eye research*, 39(3):239–244, 2014.

[3] cnet. Cafe opens in tokyo staffed by robots controlled by paralyzed people. `https://www.cnet.com/news/paralyzed-people-hired-to-control-robot-waiters-in-tokyo-cafe/5`, 2018.

[4] E.S. Dalmaijer. Open source eye-tracking software and more. `http://www.pygaze.org/`, 2018.

[5] Diane Damos. *Multiple task performance.* CRC Press, 1991.

[6] Jennifer H Darrien, Katrina Herd, Lisa-Jo Starling, Jay R Rosenberg, and James D Morrison. An analysis of the dependence of saccadic latency on target position and target characteristics in human subjects. *BMC neuroscience*, 2(1):13, 2001.

[7] Munjal Desai, Katherine M Tsui, Holly A Yanco, and Chris Uhlik. Essential features of telepresence robots. In *2011 IEEE Conference on Technologies for Practical Robot Applications*, pages 15–20. IEEE, 2011.

[8] DTU-R3. dtu-r3.github.io. `https://dtu-r3.github.io/`, 2018.

[9] Francis T Durso, Carla A Hackworth, Todd R Truitt, Jerry Crutchfield, Danko Nikolic, and Carol A Manning. Situation awareness as a predictor of performance for en route air traffic controllers. *Air Traffic Control Quarterly*, 6(1):1–20, 1998.

[10] Mohamad A Eid, Nikolas Giakoumidis, and Abdulmotaleb El Saddik. A novel eye-gaze-controlled wheelchair system for navigating unknown environments: case study with a person with als. *IEEE Access*, 4:558–573, 2016.

[11] Mica R Endsley. Situation awareness-oriented design. In *The Oxford Handbook of Cognitive Engineering.*

[12] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human factors*, 37(1):32–64, 1995.

[13] Mica R Endsley and Daniel J Garland. *Situation awareness analysis and measurement.* 2000.

[14] John Findlay. Human saccadic eye movements. `http://www.scholarpedia.org/article/Human_saccadic_eye_movements`, 2012.

[15] International Society for Presence Research. Presence defined. https://ispr.info/about-presence-2/about-presence/.

[16] Dorrit Givskov. Gazeit. `http://www.innovation.man.dtu.dk/english/research/design/research-projects/gazeit-in-danish/`, 2017.

[17] John Paulin Hansen, Alexandre Alapetite, I Scott MacKenzie, and Emilie Møllenbach. The use of gaze to control drones. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 27–34. ACM, 2014.

[18] John Paulin Hansen, Diako Mardanbegi, Florian Biermann, and Per Bækgaard. A gaze interactive assembly instruction with pupillometric recording. *Behavior research methods*, 50 (4):1723–1733, 2018.

[19] Eckhard H Hess and James M Polt. Pupil size in relation to mental activity during simple problem-solving. *Science*, 143(3611):1190–1192, 1964.

[20] Samuel B Hutton. Cognitive control of saccadic eye movements. *Brain and cognition*, 68(3): 327–340, 2008.

[21] Jason Jerald. *The VR book: Human-centered design for virtual reality.* Morgan & Claypool, 2015.

[22] Dooho Jung, Seongsik Jo, and Rohae Myung. A study of relationships between situation awareness and presence that affect performance on a handheld game console. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, pages 240–243. ACM, 2008.

[23] Annica Kristoffersson, Silvia Coradeschi, and Amy Loutfi. A review of mobile robotic telep-

resence. *Advances in Human-Computer Interaction*, 2013:3, 2013.

[24] Lærd. Three-way anova in spss statistics. `https://statistics.laerd.com/spss-tutorials/three-way-anova-using-spss-statistics.php`, 2018.

[25] Simon Liversedge, Iain Gilchrist, and Stefan Everling. *The Oxford handbook of eye movements*. Oxford University Press, 2011.

[26] I Scott MacKenzie. *Human-computer interaction: An empirical research perspective*. Newnes, 2012.

[27] Päivi Majaranta. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies: Advances in Assistive Technologies*. IGI Global, 2011.

[28] Päivi Majaranta. Communication and text entry by gaze. In *Gaze interaction and applications of eye tracking: Advances in assistive technologies*, pages 63–77. IGI Global, 2012.

[29] Diako Mardanbegi, Thomas Wilcockson, Peter Sawyer, Hans-Werner Georg Gellersen, and Trevor Jeremy Crawford. Saccademachine: Software for analyzing saccade tests (anti-saccade and pro-saccade). In *The Symposium on Communication by Gaze Interaction (COGAIN) at ETRA'19*. ACM, 3 2019.

[30] Saul McLeod. Experimental method. `https://www.simplypsychology.org/experimental-method.html`, 2012.

[31] Katsumi Minakata, John Paulin Hansen, Per Bækgaard, I Scott MacKenzie, and Vijay Rajanna. Pointing by gaze, head, and foot in a head-mounted display. In *ETRA ´ 19*, 2019.

[32] Marvin Minsky. Telepresence. 1980.

[33] Emilie Møllenbach, Martin Lillholm, Alastair Gail, and John Paulin Hansen. Single gaze gestures. In *Proceedings of the 2010 symposium on eye-tracking research & applications*, pages 177–180. ACM, 2010.

[34] Fiona Mulvey. Eye anatomy, eye movements and vision. In *Gaze interaction and applications of eye tracking: Advances in assistive technologies*, pages 10–20. IGI Global, 2012.

[35] DP Munoz, JR Broughton, JE Goldring, and IT Armstrong. Age-related performance of human subjects on saccadic eye movement tasks. *Experimental brain research*, 121(4):391–400, 1998.

[36] Carman Neustaedter, Gina Venolia, Jason Procyk, and Daniel Hawkins. To beam or not to beam: A study of remote telepresence attendance at an academic conference. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 418–431. ACM, 2016.

[37] Holger T Regenbrecht, Thomas W Schubert, and Frank Friedmann. Measuring the sense of presence and its relations to fear of heights in virtual environments. *International Journal of Human-Computer Interaction*, 10(3):233–249, 1998.

[38] Jennifer M Riley, David B Kaber, and John V Draper. Situation awareness and attention allocation measures for quantifying telepresence experiences in teleoperation. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 14(1):51–67, 2004.

[39] Double robotics. Website double robotics. `https://www.doublerobotics.com/pricing.html`, 2019.

[40] Paul M Salmon, Neville A Stanton, Guy H Walker, Daniel Jenkins, Darshna Ladva, Laura Rafferty, and Mark Young. Measuring situation awareness in complex systems: Comparison of measures study. *International Journal of Industrial Ergonomics*, 39(3):490–500, 2009.

[41] Kai-Uwe Schmitt, Rolf Seeger, Hartmut Fischer, Christian Lanz, Markus Muser, Felix Walz,

and Urs Schwarz. Saccadic eye movement performance as an indicator of driving ability in elderly drivers. *Swiss medical weekly*, 145:w14098, 2015.

[42] Valentin Schwind, Pascal Knierim, Nico Haas, and Niels Henze. Using presence questionnaires in virtual reality. 2019.

[43] Aaron Steinfeld, Terrence Fong, David Kaber, Michael Lewis, Jean Scholtz, Alan Schultz, and Michael Goodrich. Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40. ACM, 2006.

[44] STU. Eye movements can open doors. https://www.dtu.dk/nyheder/2018/01/en-bevaegelse-med-oejet-kan-aabne-doere-via-mobilen?id=e24abc17-5a19-4708-b62c-7ca788893001.

[45] Martin Tall, Alexandre Alapetite, Javier San Agustin, Henrik HT Skovsgaard, John Paulin Hansen, Dan Witzner Hansen, and Emilie Møllenbach. Gaze-controlled driving. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4387–4392. ACM, 2009.

[46] MD Timothy C. Hain. Saccade tests. `https://www.dizziness-and-balance.com/practice/saccades/saccade.htm`, 2019.

[47] TOBII. Tobii eye tracking debuts in esports. `https://blog.tobii.com/tobii-eye-tracking-debuts-in-esports-4162b10d5e049`, 2017.

[48] Katherine M Tsui, Munjal Desai, Holly A Yanco, and Chris Uhlik. Exploring use cases for telepresence robots. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 11–18. ACM, 2011.

[49] Michael A Vidulich, Michael Stratton, Mark Crabtree, and Glenn Wilson. Performance-based and physiological measures of situational awareness. *Aviation, Space, and Environmental Medicine*, 1994.

[50] Wikepedia. Eye tracking. `https://en.wikipedia.org/wiki/Eye_tracking#Eye-tracking_vs._gaze-trackingr`, 2019.

[51] Wikipedia. Nasa-tlx. `https://en.wikipedia.org/wiki/NASA-TLX#cite_note-Colligan2015-1`, 2019.

[52] Bob G Witmer and Michael J Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3):225–240, 1998.

[53] Guangtao Zhang, John Paulin Hansen, and Katsumi Minakata. Hand- and gaze-control of telepresence robots. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, pages 70:1–70:8, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6709-7. doi: 10.1145/3317956.3318149. URL `http://doi.acm.org/10.1145/3317956.3318149`.

[54] Guangtao Zhang, John Paulin Hansen, Katsumi Minakata, Alexandre Alapetite, and Zhongyu Wang. Eye-gaze-controlled telepresence robots for people with motor disabilities. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 574–575. IEEE, 2019.

Appendices

A Appendix - Experiment

A.1 Experimental Design

Hypothesis
When operating a telepresence robot wearing a HMD the users show no difference in their task performance, workload, situational awareness, saccade test results and self-assessment independent of the training environment they trained in and maze layout in the pre-and final trial.

Experimental Method
Between-group design

Design
Independent Variables:

- Training environment (virtual robot in VR & robot in real world)
- Layout of training environment (same & different)
- Trial order (pretrial and final trial)

Dependent Variables (measures):
Performance (log data from robots and *GamesOnTracks*), workload (*NASA-TLX*), situation awareness (Pop-up: SPAM & saccade test), self-assessment (SAM), eye behavior (log data from *FOVE*), recollection.

Conditions
Training types:
1. Telerobot Test + 5 x VR training with same layout + Telerobot test;
2. Telerobot Test + 5 x VR training with different layout + Telerobot test;
3. Telerobot Test + 5 x Reality training with same layout + Telerobot test
4. Telerobot Test + 5 x Reality training with different layout + Telerobot test

Room layouts:
Same layout with real scenario
Different layout with real scenario
VR mazes for training, same with real scenario or not?

Participants
32 participants (2 * 16)

Ideally balanced latin square:
Participants performed the experiment in a single session lasting around 60 minutes.
The order in which participants were exposed to each maze was counterbalanced by a Latin square.

Apparatus and Setup
Outside Design Lab:

- A PC with Unity, connected with the telerobot;
- A laptop with limesurvey local server, connected with *GamesOnTracks (GOT) sensors;*
- A *FOVE* headset and a joystick connected with the PC;

Inside Design Lab:

- A telerobot with a 360 degree camera and a microphone, connected with the headset and a joystick;
- 5 GOT sensors for telerobot's indoor positioning measurement;
- Maze built inside the room with white sticks on the floor:

- A *Ricoh theta V* camera

Tasks
Test persons sitting outside Design Lab:

- A test person sits in the lab and wears the FOVE headset;
- The test person is introduced about devices and tasks;
- Each test person is tasked with remotely driving a gaze-controlled robot around an obstacle course, meeting a person there, and reaching to the entrance of Design Lab. Afterwards, the test person has training in VR/reality for 5 times. In each training session, when they have two collisions, the training session ends. The test person who has the fewest collision will get a extra bonus. After training session, the test person is asked to repeat remotely driving the gaze-controlled robot again.
- During the first and last driving task, SA queries and saccade test appear as pop-up.
- When finishing the task, they have an interview with the conductor to answer questions.

Experimenter outside Design Lab:

- Sitting next to the test person;
- Follow the instruction to change training type and maze layout inside Design Lab;
- Control the SA queries and saccade test by using keyboard of the PC;
- Record times of collisions.

Person walking inside Design Lab:

- Provide oral information about himself/herself, ask the test person to drive the telerobot to entrance point.

Data collection
Performance (log data from robots and *GamesOnTracks*), workload (*NASA-TLX*), situation awareness (Pop-up: SPAM & saccade test), self-assessment (SAM), task confidence, eye behavior (log data from *FOVE*), recollection.

Log data from Telerobot including timestamp, position (x, y) on the floor, linear&angular velocity;
Log data from *GamesOnTracks,* including timestamp, position (x, y);
Log data from Unity, SPAM (timestamp, query type, response time);
Log data from *Fove*;
Video record using *Ricoh theta V*;
Reponse to questionnaires on *limesurvey*: demographic information, NASA-TLX, SAM, task confidence, and post-trial recollection/estimation.
Post-test interview;

Attachments:

- Schedule (for Test Persons)
- Experiment Protocol
- SA Queries (SPAM) and saccade test procedure
- Task Load Index (NASA-TLX)
- Self-Assessment Manikin scale (SAM)
- Consent Form
- CounterbalancingScheme (with maze layouts)
- Pre-Test Questionnaire
- Post-Test Interview

A.2 NASA TLX

Figure 8.6

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

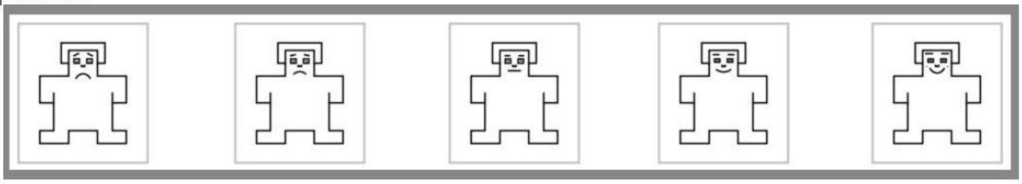
Very Low — Very High

A.3 Self Assessment Manikin - SAM

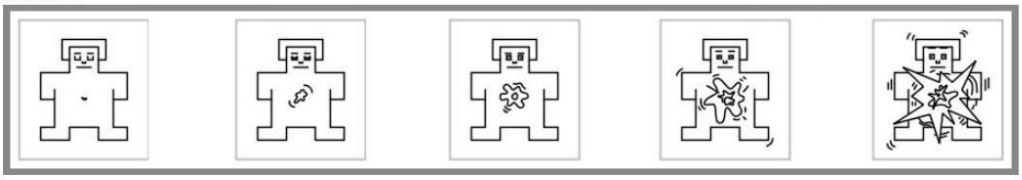
Pre-Test:Self-Assessment Manikin (SAM)

Please self-reported your momentary feelings of pleasure, arousal and dominance using the pictorial rating scale below:

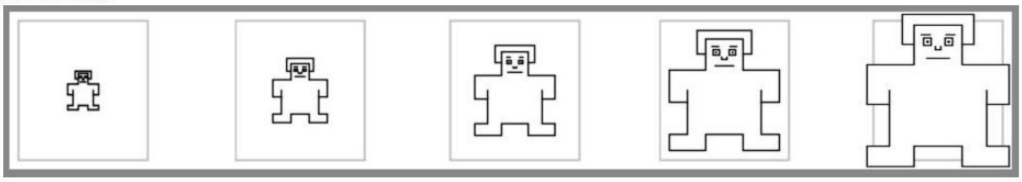
Pleasure



Arousal



Dominance



(the Self-Assessment Manikin or SAM: Bradley & Lang, 1994)

A.4 Consent Form

ID: _____

Collection of Research Data Consent Form

You are invited to participate in a research study on gaze controlled telepresence robot. From the information collected and studied in this project, we hope to improve our understanding of user-centered design in our research.

1. **Procedures:** With your permission, we would like to have you participate in a prototype test session. In this session you will be asked to use a version of our prototype and share any positive or negative feelings you have about our prototype. Video of this session may be recorded. During and after the test, you will be asked some questions related to your test. Audio of this session may be recorded.
2. **Risks & Benefits:** There are no anticipated risks associated with this study. We cannot and do not guarantee or promise that you will receive any benefit from this study.
3. **Time Involvement:** If you agree to participate, your participation in this study will not require more time from you other than this instance where you evaluate the prototype. We estimate that it will take no longer than fifteen minutes
4. **Participant's Rights:** If you have read this form and have decided to participate in this project, please understand your participation is voluntary and you have the right to withdraw your consent or discontinue participation at any time. Your identity will not be disclosed in any published and written material resulting from the study.

Authorization to Use Your Results for Research Purposes

Because information about you is personal and private, it generally cannot be used in a research study without your written authorization. If you sign this form, it will provide that authorization.

The form is intended to inform you about how your results and information will be used or disclosed in the study. Your information will only be used in accordance with this authorization form and the informed consent form and as required or allowed by law.

Please read the following carefully before signing this authorization form:

1. This research project seeks to improve our understanding of user-centered design.
2. You do not have to sign this authorization form. But if you do not, you will not be able to participate in this research study.
3. If you decide to participate, you are free to withdraw your authorization regarding the use and disclosure of results information (and to discontinue any other participation in the study) at any time. After any revocation, your results will no longer be used or disclosed in the study, except to the extent that the law allows us to continue using your information (e.g., necessary to maintain integrity of research).
4. If you wish to revoke your authorization for the research use or disclosure of your survey information in this study, you must do so in writing.
5. Your name from this research study will not disclosed to anybody.
6. The only personal information that will be shared is your familiarity with our devices

and prototype. This information will be shared anonymously.

7. Your observations and comments relating to our prototype will be shared anonymously.
8. The following research group are authorized to use your result information in connection with this research study as described above:
   Innovation Division, DTU Management, Technical University of Denmark.

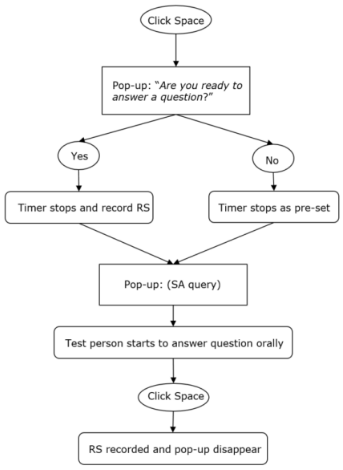
____

Signature of Participant / Date

A.5 Pop-up queries(SA) - SPAM

Situation Present Assessment Method (SPAM) will be used as a Method of measuring test person's situation awareness (SA) during the test.

Process:



Queries:

The following queries will show up as pop-up:

Hotkey	Time	Query	Aspect
-	Before questions of each session	*"Are you ready to answer a question?"*	Mental workload
Q	*When approaching the person*	*"Have you seen a person in the room?"*	SA-perception
	After talking with the person	*"What is the name of the person who is talking with you?"*	SA-comprehension
	Close to the end	*"Can you estimate the distance between you and the position where you will be finished with the task?"*	SA-projection
W	*When approaching the person*	*"Which direction are you facing now?"*	SA-perception
	After talking with the person	*"What kind of information did the person tell you?"*	SA-comprehension
	Close to the end	*"Can you estimate the distance between you and the position where you will be finished with the task?"*	SA-projection

A.6 Script for indtroducing the task

Welcome + description:
Welcome to skylab where we are going to conduct the experiment.

Please have a seat here. (hand them consent form) Can you please read this consent form and sign it if you agree. The main focus with this is that we will collect data, which all will be anonymous and will not be shared with any 3. party. At any point, for any reason, you can end your participation in the experiment.

You are going to drive a robot in the next room. You are going to where a VR headset with build in eye-trackers. This will track where you look in the Virtual environment. When you enter the Virtual environment you will see a square with a direct livestream from the robots camera that follow you around. You will also see another square with an red border. The robot will only take your driving instructions when you look inside this square. When you look in the middle of square the robots move forward, look left, it drives left, look right, it drives right. the further upwards you look the faster it drives. At anytime you can look outside the square and the robot will stop. You can look around in the virtual space by moving your head.

Before we start the test could you please fill in this form (refer to computer)

And fill out this self-Assesment scheme.
Pleasure:
Are you in a good mood?

Arousal:
Are you excited?

Dominace:
How much in control do you feel?

The experiment:
The experiment will be divided into 3 parts. one pretrial, training and Exam.

First you will have one run with the robot to measure how you control it before being trained. Then we will conduct 5/8 training sessions before you do your final run.

In the first and final run of the robot you will have to complete three type of tasks, while operating the robot.

Saccade test:
Test nr. 1 will appear on the screen as a blured square with a moving red dot on it. It is your job to follow the moving dot. Try to keep your head steady and only let your eyes moving.

Pop-up querries:

Test nr. 2 appears on the screen as a blue coloured box and the robot control stops, a question will be asked that you can answer orally and robot operation starts after. Respond as fast as possible.

After each test move your head a little to be able to move again.

Person interaction:
First you will have to operate the robot towards the person in the experiment room. Avoid hitting the “walls” of the maze on the floor. pause the robot by going into park mode when reaching the person, this is done by looking down on the park botton.
The person will then provide you your next destination.

Pre-info to the trial run:
When you start operating the robot you are facing south.

After the test is conducted:

Can you please fill in this Post-trial schema.

A.7 Experimenter note book(each experiment

ID:
Date(mm.dd.hh:mm):

	1st Trial	Training 1	Training 2	Training 3	Training 4	Traning 5	Exam	Notes
Hit								
Coverage (%)								
SA1								
SA2								
SA3								
Notes								

Notes for Interview:

A.8 Post trial interview

ID _____

Trial _____

Post-Trial Interview

0 1 2 3 4 5 6 7 8 9 10 >10

1. Can you estimate how many minutes you have spent on driving the telerobot?

2. Did you experience any technical or other problems that reduced your performance?

3. What's your impression of your own reaction times with the "follow the dot" and the questions?

4. How many times did you feel that the telerobot collided?

5. Can you draw a rough sketch of the maze layout from your memory?

6. Here is the maze where you were driving through. Please make a mark where the person was standing and where he/she was giving you instructions.

7. How did you feel about the VR glasses? (VR sickness, headache, calibration problems….) (only fort the last round)

8. Overall how confident did you feel while driving the robot, on a scale from 1 to 5? And can you explain why you chose that confidence?

9. Do you have any questions or comments? (only for the last round)

A.9 Experiment protocl

Experiment Protocol (Gaze - controlled Telepresence Training)

Outside Design Lab

- Before participant shows up, make sure that equipment is set up and all relevant forms are available.

Hardware equipment

- PC, connected with telerobot inside Design Lab;
- Laptop
- Keyboard
- Mouse
- *Fove* head mounted display (HMD)

Forms

- consent form
- Protocol
- Task Instruction
- Questionnaires (paper, *limesurvey*)
- Post-Test interview

Software

- Start up *Unity and video* recorder software
- Video recorder
- limesurvey, GOT connection on laptop

After the participant shows up,

- Welcome and greet participant and have them sit in front of computer monitor.
- Give participant consent form and ask participant to read and sign form if they agree to give consent in order to participate.
- Ask the participant to fill in questionnaire with laptop;
- Refer to Verbal Protocol for instructions that are to be read to participant

Running of Experiment

- Explain the task based on instruction
- Refer to counterbalancing scheme for order of conditions (ID, maze type, training type)
- Check if the headset, GOT, and the telerobot has been connected or not
- Adjust HMD accordingly
- Place HMD on participant's head
- Click calibration button, ask the participant to follow instruction
- Click start button of screen shooting software to start recording
- Click on "setting" in drop-down menu and input ID and maze information
- Click "start running" button in *Unity*

- Click “H” to center the view
- Tell the test person that they are facing to the south now.

During First and Last trial

- Press

Time	Hotkey
Saccade Test A	A
When approaching the person	Q (for First Trial) W (for Last Trial)
After talking with the person	
Close to the end	C (both trials)
Saccade Test B	B

During Training Session

- Refer to counterbalancing scheme for training condition (reality/VR) and start the first training
- Click start button of screen shooting software to start recording
- Click start button in Unity
- Obeserve and record the telerobot’s collision times
- Stop the training session when the telerobot has two collision times
- Click stop button in Unity and screen shooting software

After first and last trial

- Click stop button in Unity
- Click Stop button in video recorder
- Ask the participant to take off HMD
- Ask the participant to fill in the questionnaire with the laptop

After participant is finished with all trials, thank them and invite them to have a post-test interview.

Inside Design Lab

- Before participant shows up, make sure that equipment are set up and the person is already inside.
- Refer to counterbalancing scheme, make sure that the maze layout has been changed

<u>Hardware equipment and tool</u>

- Telerobot connected with the PC outside Design Lab
- 5 Sensors (*GamesOnTrack*)
- Ricoh theta V Camera

<u>Before first and last trial</u>

- Refer to counterbalancing scheme for order of conditions
- Put telerobot to the starting point of the maze
- Check GOT's connection
- Start Ricoh theta V camera to record (First trial and last trial only)
- Person who will walk in the maze is ready

<u>After first and last trial</u>

- Stop video recording
- Check the maze and change to the corresponding design

<u>Before each training (reality)</u>

- Put the telerobot to the starting point of the maze

<u>After each training (reality)</u>

- Check the maze and change to the corresponding design

A.10 Experiment checklist

Check list experiment:	**Done**
Before participant arrives:	
Set up trial maze accordint to scheduele	
Test if the robot works and have a live stream	
Put out consent form	
Set-up survey on computer	
Write name, age, condition etc. in unity for data collection	
Note what condition there are going to train in (VR or real world)	
Make sure all questions are active for test.	
Put tape on the floor to indicate goals	
When participant arrive:	
Hand them consent form.	
Read them instructions	
Let them fill in demographic survey and SAM	
Put on headset, adjust position	
Calibrate them	
During 1. Trial:	
Turn on camera	
Inform them again that they are facing south	
Let them drive a meter or so: press A.	
Let them drive a little more: press C: (make schema with each participant answer to the question) in this case what direction they are facing.	
Let them drive a little more, right before reaching examiner: press R, answer question regarding there progeress	
Participant get instruction	
Let them turn around: press B	
Let them swing left towards the door: press W: the instructers information (note!! Does this make sense??)	
turn off camera	
Ask them to fill in SAM and NASA survey	
Finalize data collect from unity	
Take out SD-card to save Games on track data	
Training:	
Instruct them on the 2-error rule	
Calibrate them.	
For each training session note the number of mistakes (Make schema for each participant)	
For each training session note the coverage percentage of necessary area	
After all training sessions: let participant fill in SAM and NASA Survey.	
Exam:	
Set maze up according to layout plan	
Calibrate them	
Turn on camera	
Inform them again that they are facing south	
Let them drive a meter or so: press A.	
Let them drive a little more: press C: (make schema with each participant answer to the question) in this case what direction they are facing.	
Let them drive a little more, right before reaching examiner: press R, answer question regarding there progeress	
Participant get instruction	
Let them turn around: press B	
Let them swing left towards the door: press W: the instructers information (note!! Does this make sense??)	
turn off camera	
Finalize data collect from unity	
Take out SD-card to save Games on track data	
NASA and SAM	
Post-trial survey –interview form conversation (take notes)	
Hand them gift card and say thank you for their participation	

A.11 Counterbalancing scheme

Counterbalancing Scheme

Conditions

1. VR training + same layout
2. VR training + different layout
3. Reality training + same layout
4. Reality training + different layout

Balanced latin square

Participants performed the experiment in a single session lasting 8-10 minutes. The order in which participants were exposed to each condition was counterbalanced by a Latin square.

Subject ID	First Trial	Training Type	Last Trial
1	Maze 1	VR training + same layout	Maze 1
2	Maze 1	Reality training + same layout	Maze 1
3	Maze 2	VR training + different layout	Maze 2
4	Maze 2	Reality training + different layout	Maze 2
5	Maze 1	VR training + same layout	Maze 1
6	Maze 1	Reality training + same layout	Maze 1
7	Maze 2	VR training + different layout	Maze 2
8	Maze 2	Reality training + different layout	Maze 2
9	Maze 1	VR training + same layout	Maze 1
10	Maze 1	Reality training + same layout	Maze 1
11	Maze 2	VR training + different layout	Maze 2
12	Maze 2	Reality training + different layout	Maze 2
13	Maze 1	VR training + same layout	Maze 1
14	Maze 1	Reality training + same layout	Maze 1
15	Maze 2	VR training + different layout	Maze 2
16	Maze 2	Reality training + different layout	Maze 2
17	Maze 1	VR training + same layout	Maze 1
18	Maze 1	Reality training + same layout	Maze 1
19	Maze 2	VR training + different layout	Maze 2
20	Maze 2	Reality training + different layout	Maze 2
21	Maze 1	VR training + same layout	Maze 1
22	Maze 1	Reality training + same layout	Maze 1
23	Maze 2	VR training + different layout	Maze 2
24	Maze 2	Reality training + different layout	Maze 2
25	Maze 1	VR training + same layout	Maze 1
26	Maze 1	Reality training + same layout	Maze 1
27	Maze 2	VR training + different layout	Maze 2
28	Maze 2	Reality training + different layout	Maze 2
29	Maze 1	VR training + same layout	Maze 1
30	Maze 1	Reality training + same layout	Maze 1
31	Maze 2	VR training + different layout	Maze 2
32	Maze 2	Reality training + different layout	Maze 2

B Appendix - Pro-saccade test code

B.1 Pro-saccade test code

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Runtime.InteropServices;
using UnityEngine;
using UnityEngine.UI;
using System.IO;
using System.Linq;
using ProBuilder2.Common;
using UnityEngine.Serialization;


/* Author : Sebastian Hedegaard Hansen and Oliver Behrens
 *
 * description:
 * Saccade test implementation
 * .
 */


public class SaccadeManager : MonoBehaviour
{
    [System.Serializable]
    public class ListWrapper
    {
        public KeyCode Hotkey;
        public bool ActiveForTest;
        public List<Vector2> coordinates;
    }

    //Inspector variables - UNITY ONLY
    [Header("Experiment Parameters")] [Tooltip("This key must correspond to
     one of the questions' hotkeys.")] [SerializeField] private KeyCode
     FinalSaccadeHotkey;

    [Tooltip("1 = Real World, 2= Simulated VR inviroment.")]
     [SerializeField] private int BLOCK;
    [SerializeField] private double Resolution_X;
    [SerializeField] private double Resolution_y;
    [SerializeField] private int SUBJECT_AGE;
    [Tooltip("f=female, m=male.")] [SerializeField] private String
     SUBJECT_GENDER;
    [SerializeField] private string SUBJECT_NAME;
   // public static string SUBJECT_NAME;
    [SerializeField] private string SUBJECT_GROUP;
    [SerializeField] private int CONDITION;

    [SerializeField] private List<ListWrapper> saccadexperiments;
    [Header("JSON formatting ")] [SerializeField] private string filePath;
    [Tooltip("Makes the JSON file valid by removing special characters used
     for appending information.")] [SerializeField] private bool
     FinalizeJsonFile = false;
    [Tooltip("Resets the JSON file to an initial preset format.")]
     [SerializeField] private bool ClearJsonFile = false;
```

```
[Header("Saccade Manager options ")] [SerializeField] private bool
 SaccadeIsActive;
[Tooltip("Resets the SaccadeManager.")] [SerializeField] private bool
 ResetTest;
[SerializeField] private RobotControlTrackPad EyeControlPanel;


//Variables
public float saccadeTestStart;

private Dictionary<KeyCode, int> CurrentActiveKeys;
private Dictionary<KeyCode, int> KeysToIndexMap;
private Dictionary<KeyCode, int> InitActiveKeys; // this is pretty ugly,
 try to manage this in a better way
private int indexSaccade;
private float fixationTimer;
private float fixation = 2.0f;
private float tagetTimer;
private float tagetTime = 1.0f;
private float lastTimer;
private bool shownFix = false;
private bool in_saccade;
public bool showingSaccade;
private int coorIndex = 0;
private int[] SaccadeListCounters;
private string sample_message;
private KeyCode pressedKey;
private bool delay;
private int? FIXATION_INDEX;
private int TRIAL_INDEX;
private int prevCoorindex;
private bool shownLastSaccade = false;
//Saccade gameobjects
private GameObject Border;

private GameObject SaccadePanel;
private GameObject background;
private GameObject Dot;
public static SaccadeManager Instance { get; private set; }
private FileStream _file;
private StreamWriter _writer;
private bool _streamClosed = false;
private int inSaccade;
private int in_fixation;
private List<string> Sample_message = new List<string>();
private List<DateTime> Timestamp = new List<DateTime>();
private List<float> trial_index = new List<float>();
private List<Vector2> Taget = new List<Vector2>();
private List<Vector2> CurrentSaccade = new List<Vector2>();
private List<Vector2> PrevSaccade = new List<Vector2>();
private float PrevSaccade_x;
private float PrevSaccade_y;
private int inSaccadePrev;
private int fixationTæller = 0;
```

```
//close the file
    private void CloseWriter()
    {
        if (!_streamClosed)
        {
            _writer.Flush();
            _file.Flush();
            _writer.Close();
            _file.Close();
            _streamClosed = true;
        }
    }

    void Start()
    {

        //catch gameobjects - the saccade test.
        SaccadePanel = transform.GetChild(2).gameObject;
        if (SaccadePanel)
        {
            background = SaccadePanel.transform.GetChild(0).gameObject;
            Border = SaccadePanel.transform.GetChild(1).gameObject;
            Dot = SaccadePanel.transform.GetChild(2).gameObject;
        }
        else
        {
            Debug.Log("fangede ikke pannel");
        }
        SaccadePanel.SetActive(false);

        //controlResult = new Vector2(-2, -2);

       ResetManager();
//create the file
        try
        {
            _file = File.Create(@"C:
             \Users\VirtualReality\Desktop\OliverOgSebastian\SaccadeData\" +
             DateTime.Now.ToString("HHmmss") + SUBJECT_NAME + "SaccadeData"
             + CONDITION + ".txt");
        }
        catch (System.Exception ex)
        {
            Console.WriteLine(ex.ToString());
        }
        //write geader
        _writer = new StreamWriter(_file);
        _writer.WriteLine("GAZE_X" + "," + "GAZE_Y" + "," + "IN_SACCADE" +
         "," + "FIXATION_INDEX" + "," + "SAMPLE_MESSAGE" + "," + "TIMESTAMP"
         + "," + "TRIAL_INDEX" + "," + "BLOCK" + "," + "RESOLUTION_X" + ","
         + "RESOLUTION_Y" + "," + "IN_BLINK" + "," + "SECOND_DISPLAY" + ","
         + "TARGET_X" + "," + "TARGET_Y" + "," + "SUBJECT_GENDER" + "," +
         "SUBJECT_AGE" + "," + "SUBJECT_NAME" + "," + "SUBJECT_GROUP" + ","
         + "IN_FIXATION" + "," + "CONDITION");
        //
```

```
    //DisableSaccadeInterface();
    showingSaccade = false;

}

void Update()
{
    if (!SaccadeIsActive)
        return;

    //Tester pushes one of the hotkeys that will activate
    if (!showingSaccade)
    {
        foreach (KeyCode key in CurrentActiveKeys.Keys)
        {
            if (Input.GetKeyDown(key))
            {
                pressedKey = key;

                saccadeTestStart = Time.deltaTime;
                indexSaccade = KeysToIndexMap[key];
                CurrentSaccade =
                 saccadexperiments[indexSaccade].coordinates;
                PauseManagers(false);
                SaccadePopUp();
                showingSaccade = true;
                sample_message = "FIXATION_TARGET_ONSET";
                Dot.transform.localPosition = new Vector2(0, 0);
                in_saccade = false;

            }
        }



    }
    //check if a certain amout of time has passed - change coordinates
     of dot. See variables for the duration
    else if (showingSaccade)
    {
        fixationTimer += Time.deltaTime;
        tagetTimer += Time.deltaTime;
        lastTimer += Time.deltaTime;

        //fixation onset
        if (fixationTimer >= fixation && !shownFix && !shownLastSaccade)
        {
            sample_message = "FIXATION_TARGET_OFFSET";
            SaveData();
            sample_message = "TARGET_ONSET";

            Dot.transform.localPosition = CurrentSaccade[coorIndex];
            Debug.Log(CurrentSaccade[coorIndex]);
```

```
    if (saccadexperiments[indexSaccade].coordinates.Count ==
     coorIndex+1)
    {
        shownLastSaccade = true;
        lastTimer = 0;
    }
     tagetTimer = 0;
    fixationTimer = 0;
    shownFix = true;
    in_saccade = true;
}
//target on set
else if (tagetTimer >= tagetTime && shownFix && !
 shownLastSaccade)
{
    sample_message = "TARGET_OFFSET";
    SaveData();
    sample_message = "FIXATION_TARGET_ONSET";
    if (!shownLastSaccade)
    {
        FIXATION_INDEX = 1;
        fixationTæller = 1;
        coorIndex += 1;
    }

    Dot.transform.localPosition = new Vector2(0, 0);
    in_saccade = false;
    fixationTimer = 0;
    shownFix = false;
    tagetTimer = 0;

}
//If the last saccade
if (shownLastSaccade && lastTimer>=tagetTime)
{
    sample_message = "TARGET_OFFSET";
    if (pressedKey == KeyCode.B)
    {
        SaveData();
    }

    Debug.Log(coorIndex);
    showingSaccade = false;
    shownFix = false;
    tagetTimer = 0;
    fixationTimer = 0;
    coorIndex = 0;
    shownLastSaccade = false;
    in_saccade = false;
    fixationTæller = 0;
    if (pressedKey == KeyCode.B)
    {
        CloseWriter();

    }
```

```
                SaccadeClosePopUp();
                PauseManagers(true);
            }


//save all variables and prec saccade coordinates for detection algoritm
            SaveData();
            PrevSaccade_x = (-VRController.SacCord().x * 10 + 10) * 51;
            PrevSaccade_y = (VRController.SacCord().y * 10 + 10) * 38;

            inSaccadePrev = inSaccade;
            sample_message = null;
        }

    }

    private void SaccadePopUp()
    {
        SaccadePanel.SetActive(true);
    }

    private void SaccadeClosePopUp()
    {
        SaccadePanel.SetActive(false);
    }

    //Stop the robot from driving
    public void PauseManagers(bool enable)
    {
        //pause/disable managers
        if (!enable)
        {
            EyeControlPanel.SetExternallyDisabled(true);
            GazeTrackingDataManager.Instance.RecordingData(false);

            if (StreamController.Instance.VirtualEnvironment)
            {
                VirtualUnityController.Instance.Disconnect();
            }
            else
            {
                RobotInterface.Instance.EnableRobotCommands(false);
            }


        }
        else
        {
            EyeControlPanel.SetExternallyDisabled(false);
            GazeTrackingDataManager.Instance.RecordingData(true);

            if (StreamController.Instance.VirtualEnvironment)
            {
                VirtualUnityController.Instance.Connect();
            }
```

```
            else
            {
                RobotInterface.Instance.EnableRobotCommands(true);
            }
        }


    }

    public void EnableSaccadeInterface()
    {

        Border.SetActive(true);
        Dot.SetActive(true);

    }

    public void DisableSaccadeInterface()
    {
        Border.SetActive(false);
        background.SetActive(false);
        Dot.SetActive(false);
    }
//collects all the relevant datapoints
    public void SaveData()
    {
        if (Math.Abs((PrevSaccade_x) - (-VRController.SacCord().x * 10 + 10)
         * 51) >= 30f || Math.Abs((PrevSaccade_y) -
         (VRController.SacCord().y * 10 + 10) * 38) >= 30f)
        {
            inSaccade = 1;
            in_fixation = 0;
            FIXATION_INDEX = null;

        }
        else
        {
            inSaccade = 0;
            in_fixation = 1;
           //FIXATION_INDEX = 1 + coorIndex;

        }
        if (inSaccadePrev ==1 && inSaccade == 0)
        {
            fixationTæller += 1;
            FIXATION_INDEX = fixationTæller;
        }
        /*if (in_saccade)
        {
            //inSaccade = 1;
            //in_fixation = 0;
            //FIXATION_INDEX = 0;
        }
        else
        {
```

```
            inSaccade = 0;
            in_fixation = 1;
            FIXATION_INDEX = 1+coorIndex;
        }*/
        string unicTimestamp =
         DateTime.UtcNow.ToString("yyyyMMddHHmmssFFF");
        if (pressedKey == KeyCode.A)
        {
            TRIAL_INDEX = coorIndex+1;
        }
        else if (pressedKey == KeyCode.B)
        {
            TRIAL_INDEX = coorIndex + 6;
        }
        if (prevCoorindex == 5 && TRIAL_INDEX == 1)
        {
            TRIAL_INDEX = 5;
        }
        int IN_BLINK = 0;
        int SECOND_DISPLAY = 0;

        //write a new line to file. Notice we added a scaling on the
         coordinates. This was done due to issues in unity.
        _writer.WriteLine((VRController.SacCord().x*15+15)*42.5 + "," +
         (VRController.SacCord().y*9+9)*40 + "," + inSaccade + ","
         +FIXATION_INDEX + "," + sample_message + "," + unicTimestamp + ","
         + TRIAL_INDEX + "," + BLOCK + "," + Resolution_X + "," +
         Resolution_y + "," + IN_BLINK + "," + SECOND_DISPLAY + "," +
         (15+CurrentSaccade[coorIndex].x)*42.5 + "," +
         (9+CurrentSaccade[coorIndex].y)*40 + "," +SUBJECT_GENDER +","+
         SUBJECT_AGE + "," + SUBJECT_NAME + "," + SUBJECT_GROUP + "," +
         in_fixation + "," + CONDITION);
        prevCoorindex = TRIAL_INDEX;
    }


    private void ResetManager()
    {

        //QueryDelay = 0;
        indexSaccade = 0;
        //CurrentQListIndex = 0;
        showingSaccade = false;
        // ResetTimers();
        SaccadeIsActive = true;
        //Initialize data structures from scratch
        KeysToIndexMap = new Dictionary<KeyCode, int>();
        CurrentActiveKeys = new Dictionary<KeyCode, int>();
        InitActiveKeys = new Dictionary<KeyCode, int>();
        SaccadeListCounters = new int[saccadexperiments.Count];
        //RSList = new List<List<ResponseTimes>>();
        for (int i = 0; i < saccadexperiments.Count; i++)
        {
            SaccadeListCounters[i] = 0;
            KeysToIndexMap.Add(saccadexperiments[i].Hotkey, i);
```

```
            //if the question type is active then enable the hotkey for the
             test
            if (saccadexperiments[i].ActiveForTest)
            {
                CurrentActiveKeys.Add(saccadexperiments[i].Hotkey, i);
                InitActiveKeys.Add(saccadexperiments[i].Hotkey, i);
            }
        }

    }



}
```

C Appendix - Data analysis results

C.1 Data analysis results

Dependent variable	Independet variable	Statictical value	P-value	Effect	(mean/SD)/(median/mean rank)	Test type
Performance						
Number of collisions	Training enivornement	F(1,60)=6.69	p<0.05	eta^2=0.076	Reality= 2.25(SD=2.67)VR= 3.46(SD=3.30)	3-way ANOVA
	Trial order	F(1,60)=20.95	p<0.001	eta^2=0.239	Pretrial= 4.50(SD=3.20)Final trial= 1.13(SD=1.56)	3-way ANOVA
	Maze layout	No significance			Different = 3.06(SD=3.39)Same= 2.56(SD=2.63)	3-way ANOVA
Task completion time	Trial order	F(1,60)=34.91	p<0.001	eta^2=0.366	Pretrial= 3.59(SD=0.50)Final trial= 2.58(SD=0.65)	3-way ANOVA
	Training enivornement	No significance			Reality= 3.10(SD=0.87)VR= 3.07(SD=0.63)	3-way ANOVA
	Maze layout	No significance			Different = 3.11(SD=0.73)Same= 3.06(SD=0.81)	3-way ANOVA
Total drive time	Trial order	F(1,60)=36.76	p<0.001	eta^2=0.377	Pretrial= 4.58(SD=0.65)Final trial= 3.59(SD=0.58)	3-way ANOVA
	Training enivornement	No significance			Reality= 4.11(SD= 0.87)VR= 4.06(SD=0.69)	3-way ANOVA
	Maze layout	No significance			Different = 4.14(SD=0.73)Same= 4.04(SD=0.85)	3-way ANOVA
Training session						
Training session 1 collisions	Maze layout	No significance			VR= 3.62(SD=3.42)= 1.87(SD=2.68)	2-way ANOVA
	Training enivornement	F(1,29) = 11	p<0.01	Eta^2 = 0,2536	Final trial= 1.18(SD=1.75)= 4.31(SD=3.49)	2-way ANOVA
Training session 2 collisions	Maze layout	No significance			Same= 2.87(SD=2.21)=	2-way ANOVA
	Training enivornement	Fails homogenity			Final trial= 1.43(SD=1.20)= 2.81(SD=2.31)	2-way ANOVA
Training session 3 collisions	Maze layout	No significance			VR= 2.06(SD=1.84)= 0.75(SD=1.18)	2-way ANOVA
	Training enivornement	F(1,29)=16.69	p<0.01	Eta^2 = 0,3065	Same= 0.5(SD=0.81)= 2.31(SD=1.81)	2-way ANOVA
Training session 4 collisions	Maze layout	No significance			Final trial= 1.37(SD=1.62)=	2-way ANOVA
	Training enivornement	No significance			VR= 0.68(SD=1.07)= 2.12(SD=2.36)	2-way ANOVA
Training session 5 collisions	Maze layout	No significance			Same= 1.62(SD=2.06)=	2-way ANOVA
	Training enivornement	No significance			= 0.5(SD=0.81)= 2.0(SD=1.93)	2-way ANOVA
Training session 1 task completion time	Maze layout	No significance			Different = 1.73(SD=1.21)Same=	2-way ANOVA
	Training enivornement	F(1,29) = 17.37	p<0.001	Eta^2 = 0,371	Reality= 1.09(SD=0.77)VR= 2.08(SD=0.93)	2-way ANOVA
Training session task completion time	Maze layout	No significance			Different = 1.05(SD=0.47)Same=	2-way ANOVA
	Training enivornement	f(1,29) = 12.51	p<0.01	Eta^2 = 0,3	Reality= 0.81(SD=0.47)VR= 1.33(SD=0.34)	2-way ANOVA
Training session 3 task completion time	Maze layout	No significance			Different = 0.98(SD=0.50)Same=	2-way ANOVA
	Training enivornement	F(1,29)=13.67	p<0.001	Eta^2 = 0,318	Reality= 0.68(SD=0.36)VR= 1.1(SD=0.39)	2-way ANOVA
Training session 4 task completion time	Maze layout	No significance			Different = 0.97(SD=0.55)Same=	2-way ANOVA

	Training enivornement	F(1,29)=7.73	p<0.01	Eta^2 = 0,2089	Reality= 0.69(SD= 0.34)VR= 1.16(SD=0.57)	2-way ANOVA
Training session 5 task completion time	Maze layout	No significance			Different = 0.95(SD=0.48)Same=	2-way ANOVA
	Training enivornement	F(1,29)=36.78	p<0.001	Eta^2 = 0,553	Reality= 0.58(SD=0.25)VR= 1.22(SD=0.35)	2-way ANOVA
Workload (NASA-TLX)						
Mental demand	Trial order	No significance			Pretrial= 12.23(SD=5.48)Final trial= 9.93(SD=4.93)	3-way ANOVA
	Training enivornement	No significance			Reality= 11.09(SD=5.36)VR= 11.07(SD=5.32)	3-way ANOVA
	Maze layout	No significance			Different = 11.96(SD=5.59)Same = 10.20(SD=4.92)	3-way ANOVA
Physical demand	Trial order	No significance			Pretrial= 7.36(SD=4.75)Final trial= 7.03(SD=4.97)	3-way ANOVA
	Training enivornement	No significance			Reality= 6.65(SD=4.53)VR= 7.82(SD=5.15)	3-way ANOVA
	Maze layout	No significance			Different = 8.03(SD=4.83)Same= 6.36(SD=4.75)	3-way ANOVA
Temporal demand	Trial order	No significance			Pretrial= 10.43(SD=4.09)Final trial= 9.06(SD=4.46)	3-way ANOVA
	Training enivornement	No significance			Reality= 9.93(SD=4.53)VR= 9.53(SD=4.09)	3-way ANOVA
	Maze layout	No significance			Different = 9.8(SD=4.5)Same= 9.7(SD=4.12)	3-way ANOVA
Overall performance	Trial order	F(1,60)=7.88	p<0.01	eta^2=0.108	Pretrial= 9.20(SD=4.66)Final trial= 6.03(SD=3.59)	3-way ANOVA
	Training enivornement	No significance			Reality= 7.18(SD=3.83)VR= 8.10(SD=5.05)	3-way ANOVA
	Maze layout	F(1,60)=4.16	p<0.05	eta^2=0.056	Different = 6.86(SD=4.08)Same= 8.36(SD=4.69)	3-way ANOVA
Effort	Trial order	F(1,60)=8.87	p<0.01	eta^2=0.128	Pretrial= 12.13(SD=4.01)Final trial= 9.00(SD=4.63)	3-way ANOVA
	Training enivornement	No significance			Reality= 10.34(SD=5.02)VR= 10.82(SD=4.09)	3-way ANOVA
	Maze layout	No significance			Different = 10.13(SD=4.82)Same = 11.00(SD=4.36)	3-way ANOVA
Frustration	Trial order	F(1,60)=8.22	p<0.01	eta^2=0.118	Pretrial= 9.63(SD=5.16)Final trial= 6.2(SD=4.23)	3-way ANOVA
	Training enivornement	No significance			Reality= 8.40(SD=4.77)VR= 7.35(SD=5.27)	3-way ANOVA
	Maze layout	No significance			Different = 8.10(SD=5.05)Same= 7.73(SD=5.01)	3-way ANOVA
Situational awareness (SPAM)						
Perception response time	Trial order	F=(1,50)=5.19	p<0.05	eta^2=0.091	Pretrial= 5.31(SD=3.07)Final trial= 9.29(SD=6.33)	3-way ANOVA
	Training enivornement	No significance			Reality= 8.15(SD=6.25)VR= 6.81(SD=4.57)	3-way ANOVA
	Maze layout	No significance			Different = 8.16(SD=5.69)Same= 6.80(SD=5.20)	3-way ANOVA
Comprehension response time	Trial order	No significance			Pretrial= 6.60(SD=3.77)Final trial= 7.80(SD=2.97)	3-way ANOVA

	Training enivornement	No significance			Reality= 6.83(SD= 2.25)VR= 7.64(SD=4.18)	3-way ANOVA
	Maze layout	No significance			Different = 7.37(SD=3.17)Same= 7.14(SD=3.63)	3-way ANOVA
Projection response time	Trial order	No significance			Pretrial= 10.67(SD=4.00)Final trial= 9.01(SD=4.21)	3-way ANOVA
	Training enivornement	No significance			Reality= 10.12(SD=4.01)VR= 9.46(SD=4.34)	3-way ANOVA
	Maze layout	No significance			Different = 10.94(SD=3.84)Same = 8.70(SD=4.22)	3-way ANOVA
Average response time of preliminary question	Trial order	F(1,50)=19.94	p<0.001	eta^2=0.215	Pretrial= 3.37(SD=1.62)Final trial= 2.34(SD=1.01)	3-way ANOVA
	Training enivornement	no significance			Reality= 2.70(SD=0.97)VR= 2.92(SD=1.74)	3-way ANOVA
	Maze layout	no significance			Different = 2.78(SD=0.89)Same= 2.85(SD=1.78)	3-way ANOVA
Answer perception question	Trial order	No significance				Chi-square test
	Training enivornement	No significance				Chi-square test
	Maze layout	No significance				Chi-square test
Answer comprehension question	Trial order	No significance				Chi-square test
	Training enivornement	No significance				Chi-square test
	Maze layout	No significance				Chi-square test
Answer projection question	Trial order	No significance				Chi-square test
	Training enivornement	No significance				Chi-square test
	Maze layout	No significance				Chi-square test
Recollection						
|Duration - estimated duration|	Trial order	No significance			Pretrial= 2.05(SD=1.62)Final trial= 1.50(SD=2.06)	3-way ANOVA
	Training enivornement	No significance			Reality= 1.58(SD=1.03)VR= 1.99(SD=2.5)	3-way ANOVA
	Maze layout	No significance			Different = 1.67(SD=1.65)Same= 1.88(SD=2.07)	3-way ANOVA
|Collisions- estimated collisions|	Trial order	F(1,58)=11.996	p<0.01	eta^2=0.165	Pretrial= 0.63(SD=0.64)Final trial= 0.72(SD=1.05)	3-way ANOVA
	Training enivornement	No significance			Reality= 0.67(SD=0.67)VR= 0.69(SD=1.06)	3-way ANOVA
	Maze layout	No significance			Different = 0.75(SD=0.94)Same= 0.61(SD=0.79)	3-way ANOVA
Estimated confidence	Trial order	F(1,57) = 18,67	p<0.001	eta^2=0.249	Pretrial= 2.90(SD=0.71)Final trial= 3.80(SD=0.89)	
	Training enivornement	No significance			Reality= 3.33(SD=0.88)VR= 3.37(SD=0.97)	
	Maze layout	No significance			Different = 3.39(SD=0.89)Same= 3.31(SD=0.96)	
Saccade test results						
Number of successful trials	Trial order	No significance			Pretrial= 73.0(SD=14.83)Final trial= 66.16(SD=17.50)	3-way ANOVA
	Training enivornement	No significance			Reality= 71.40(SD=16.02)VR= 67.50(SD=16.9)	3-way ANOVA
	Maze layout	No significance			Different = 69.83(SD=13.80)Same= 69.33(SD=18.97)	3-way ANOVA
Number of correct trials	Trial order	No significance			Pretrial= 69.00(SD=15.99)Final trial= 62.83(SD=18.36)	3-way ANOVA

	Training enivornement	No significance			Reality= 67.65(SD=16.89)VR= 63.92(SD=17.96)	3-way ANOVA
	Maze layout	No significance			Different = 65.50(SD=15.33)Same= 66.33(SD=19.42)	3-way ANOVA
Latency of first saccade - interaction effect	Trial order	F(1,56)=5,271	p<0,05	eta ^2= 0.073	Pretrial= 300.15(SD=33.53)Final trial= 277.36(SD=43.67)	3-way ANOVA
	Maze layout				Reality= 291.0(SD=46.72)VR= 286.19(SD=32.04)	
	Training enivornement	No significance				3-way ANOVA
Latency of first saccade pretrial	Maze layout	F(1,29)=6.33	p<0.05	eta ^2= 0.179		2-way ANOVA
	Training enivornement	No significance				2-way ANOVA
Latency of first correct saccade	Trial order	No significance			Different = 293.12(SD=36.48)Same= 279.9(SD=42.35)	3-way ANOVA
	Training enivornement	No significance			Reality= 288.17(SD=45.78)VR= 284.61(SD=32.24)	3-way ANOVA
	Maze layout	No significance			Different = 285.80(SD=32.76)Same= 287.21(SD=46.26)	3-way ANOVA
Amplitude any correct saccade	Trial order	No significance			Pretrial= 9.17(SD=1.37)Final trial= 9.54(SD=1.69)	3-way ANOVA
	Training enivornement	No significance			Reality= 9.39(SD=1.68)VR= 9.32(SD=1.38)	3-way ANOVA
	Maze layout	No significance			Different = 9.32(SD=1.62)Same= 9.39(SD=1.47)	3-way ANOVA
SAM						
Self assesed pleasure	Trial order	No significance			Mean rank Pretrial= 27.85 mean rank Final trial= 1,38541666666667/ Median Pretrial=4 Median Final trial =4	Mann-Whitney U test
	Training enivornement	U = 670, Z = 2.2	p<0.05	r = 0.275	Mean rank Reality= 34.40 mean rank VR= 26.03/ Median Reality=4 Median VR =4	Mann-Whitney U test
	Maze layout	No significance			Mean rank Different = 28.66 mean rank Same= 32.33/ Median Different =4 Median Same =4	Mann-Whitney U test
Self assessed dominance	Trial order	U = 300, Z = -3	p<0.01	r = 0.375	Mean rank Pretrial= 24.41 mean rank Final trial= 36.58/ Median Pretrial=3 Median Final trial =4	Mann-Whitney U test
	Training enivornement	No significance			Mean rank Reality= 32.65 mean rank VR= 28.03/ Median Reality=4 Median VR =3	Mann-Whitney U test
	Maze layout	No significance			Mean rank Different = 29.41 mean rank Same= 31.58/ Median Different =3 Median Same =3.5	Mann-Whitney U test
Self assesed arousal	Trial order	No significance			Mean rank Pretrial= 31.4 mean rank Final trial= 29.6/ Median Pretrial=4 Median Final trial =3.5	Mann-Whitney U test

	Training enivornement	No significance			Mean rank Reality= 32.46 mean rank VR= 28.25/ Median Reality=4 Median VR =4	Mann-Whitney U test
	Maze layout	No significance			Mean rank Different	Mann-Whitney U
Sel assessed pleasure after training	Trial order	No significance				Mann-Whitney U test
	Training enivornement	U = 197, Z = 2.17	p<0.01	r= 0.375		Mann-Whitney U test
	Maze layout	No significance				Mann-Whitney U
Self assessed dominance after training	Trial order	No significance				Mann-Whitney U test
	Training enivornement	No significance				Mann-Whitney U test
	Maze layout	U = 66.5, Z = -2.51	p<0.05	r = 0.444		Mann-Whitney U test
8 best vs. 8 worst performers						
Number of correct saccades on pretrial	Performance group	F(1,14) = 3,913	p<0,1	η^2 = 0.2185	Best= 70.00(SD=15.11)Worst= 46.87(SD=29.391)	1-way ANOVA
Amplitude of first wrong saccade on pretrial	Performance group	F(1,7)=12,53	p<0,001	η^2 = 0.641	Best= 3.02(SD=0.67)Worst= 7.39(SD=2.02)	1-way ANOVA
Reaction time 2. projection related question	Performance group	F(1,9) = 6,589	p<0,05	η^2 = 0,351	Best= 7.57(SD=2.28)Worst= 12.05(SD=5.42)	1-way ANOVA
Reactiontime preliminary question 1. projection related question	Performance group	F(1,9) = 6,163	p<0,05	η^2 = 0.406	Best= 1.83(SD=0.45)Worst= 9.84(SD=15.65)	1-way ANOVA
Latency any correct saccade pretrial	Performens group	U= 4, Z = -2,777	p<0,01	r = 1.694	Mean rank Best= 5 mean rank Worst= 12/ Median best=273.84 Median worst =346.00	Mann-Whitney U test
Latency any correct saccade final trial	Performens group	U= 12, Z = -2,1	p<0,05	r = 0.525	Mean rank Best= 6 mean rank Worst= 11/ Median best=266.85 Median worst =323.83	Mann-Whitney U test
Latency first correct saccade pretrial	Performens group	U= 2,5, Z = -2,95	p<0,01	r = 0.73	Mean rank Best= 4 mean rank Worst= 12.18/ Median best=290.41 Median worst =313.33	Mann-Whitney U test
NASA effort pretrial	Performens group	U= 13, Z = -2	p<0,05	r = 0.501	Mean rank Best= 6 mean rank Worst= 10.87/ Median best=9.5 Median worst =14.5	Mann-Whitney U test
NASA Physical after training	Performens group	U= 11,5, Z = -2,16	p<0,05	r = 0.541	Mean rank Best= 5 mean rank Worst= 11.06/ Median best=4.5 Median worst =10.5	Mann-Whitney U test
NASA Physical final trial	Performens group	U= 5,5, Z = -2,78	p<0,01	r = 0.697	Mean rank Best= 5 mean rank Worst= 11.81/ Median best=4.5 Median worst =10	Mann-Whitney U test
Self assessed pleasure after pretrial	Performens group	U= 11,5, Z = -2,161	p<0,05	r = 0.541	Mean rank Best= 1 mean rank Worst= 5.5/ Median best=4.5 Median worst =3.5	Mann-Whitney U test
Self assessed pleasure after final trial	Performens group	U= 56, Z = 2,63	p<0,05	r = 0.659	Mean rank Best= 1 mean rank Worst= 5.5/ Median best=4.5 Median worst =3	Mann-Whitney U test