EXPLORING MIXED REALITY ROBOT COMMUNICATION UNDER DIFFERENT TYPES OF MENTAL WORKLOAD

by

Nhan Tran

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Computer Science).

Golden, Colorado

Date _____

Signed: _____
Nhan Tran

Signed: _____
Dr. Thomas Williams
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Tracy Camp
Professor and Head
Department of Computer Science

ABSTRACT

Research has shown that the use of physical deictic gestures such as pointing and presenting by robots enables more effective and natural human-robot interaction. However, not all robots come equipped with gestural capabilities. Recent advances in augmented reality (AR) and mixed reality (MR) provide powerful new forms of deictic gestures in human-robot communication. My thesis focuses on allocentric mixed reality gestures, in which target referents are picked out in fields of view of human teammates using AR visualizations such as circles and arrows, especially when these gestures are paired with verbal referring expressions and deployed under various types of mental workload of human teammates. We also present a software architecture that enables mixed reality gestural capabilities, and present the results of a human subject experiment that measures user objective performance and their subjective responses. These results demonstrate the trade-offs between different types of mixed reality robotic communication under different levels of user workload. The findings of this study suggest that although humans may not notice differences, the manner of load a user is under and the type of communication style used by a robot they interact with do in fact interact to determine their task time. The data collected from my experiment is a first step towards answering this overarching question: *How can a robot select the most effective communication modality given information regarding its human teammate's level and type of mental workload?*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Human-Robot Interaction	HRI
Brain-Computer Interface	BCI
Functional near-infrared spectroscopy	fNIRS
Electroencephalography	EEG
Head-mounted display	HMD
Brain-machine interface	BMI
NASA Task Load Index	NASA TLX
Bayes factors	Bf

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Thomas Williams, for sparking my research interests in human-robot interaction and mixed reality. I would like to thank my undergraduate research teammates (Kai Mizuno, Morgan Cox, Jared Hamilton, and Nicholas Woodward) at the Mines Interactive Robotics Research Lab for assisting me in this project and running some of the human subject experiments. I would also like to thank Dr. Leanne Hirshfield and collaborators within the System Human-Interaction with NIRS and EEG (SHINE) Lab at the University of Colorado at Boulder's Institute of Cognitive Science for providing valuable perspectives from their experience in cognitive science and neurophysiology. Lastly, I wish to express my sincere appreciation to the Computer Science department at Colorado School of Mines led by Dr. Tracy Camp and the growing Robotics program. Without their support and funding, this project could not have reached its goal.

CHAPTER 1

INTRODUCTION

In recent decades, the demand for human-robot teaming continues to grow in domains such as industrial factories, healthcare, and search-and-rescue operations. Successful human-robot teaming requires robots to facilitate effective and natural communication, while minimizing the need for special training for the human partners who might not be robotics experts. Researchers in the field of Human-Robot Interaction (HRI) have sought to enable robots to engage in natural and human-like dialogue [2–4]. Unlike chat bots and conversational agents, interactive robots need to facilitate natural language communication that is situated [5], sensitive to the situated context [5–7], and designed as a task-based dialogue [8]. For example, if a human asks the robot "Could you bring me that mug of coffee please?", the robot needs to map the word "mug" to its own perception of the world (i.e., recognizing the objects in its camera sensor corresponding to the word) and extract the intent in the utterance (e.g., the human wants me to grab the cup and bring it to them). After producing a plan or policy to grasp the cup, the robot should be sensitive to the uncertain nature of the task, though not explicitly stated, that the human wants it to lift the mug in a way that does not spill the coffee. In addition, natural language and nonverbal gesture are usually combined together in human natural language communication, as gesture complements fluent speech to convey abstract ideas and to draw attention to the area that contains the target referent. Speakers often employ deictic gestures such as pointing to direct the attention of the listeners to the object of interest in their environment. The use of deictic gestures helps to minimize the ambiguity of the speaker's utterance to refer to their target and to minimize the addressee's cognitive strain when processing the speaker's description of the target referent.

Since nonverbal gestures can accompany language to bear the burden of communication and even substitute for speech on its own [9], HRI researchers have sought to make interactive robots capable of understanding and generating human-like, appropriate gestures in situated interactions with humans. While research has been conducted to allow robots to understand gestures using camera and depth sensors [10], generation of deictic gesture remains a challenge due to the complexity of robot perception, action, and the need for expressive end effectors. HRI researchers have used Wizard-of-Oz, a technique in which a puppeteer manually controls the robot behind the scene to study how humans perceive robot-generated deictic gestures beyond pointing such as presenting, exhibiting, touching, grouping and sweeping [11]. Most of the robots currently in use, such as assistive wheelchairs, robot vacuum cleaner, and unmanned aerial vehicles, have neither expressive end effectors nor gesture-generating capabilities. And even for robots that have arms like industrial robot arms and humanoid robots, the generation of deictic gestures is still limited, particularly when the robots need to communicate in an unknown environment about hard-to-describe referents. Tran et al. [12] pose a scenario involving a human teammate and an unmanned aerial vehicle (UAV), collaborating together in a search-and-rescue operation. If the robot has no arms and needs to communicate with a human teammate to search out a victim in a highly ambiguous environment, it can not simply say "I found a victim behind that tree" without having a physical gesture accompanying the language. Moreover, if this robot has arms, a simple pointing gesture would probably not help to guide the human teammate in a labyrinthine environment without using complex language such as "The victim is in the clump of trees to the right of the large boulder near the fourth tree on the left." [12]

Mixed reality (sometimes referred to as augmented reality) technologies that overlay virtual objects in the physical world allow new approaches to tackle the above limitations and allow robots to generate gestural cues in the human's field of view. Consider the earlier example, if the human partner wears a mixed reality head-mounted display (HMD), the UAV could draw visual annotations around the region of interest, such as arrows and circles, and

tell the human "there is a victim behind [circle] that tree." Furthermore, while mounting physical arms on these robots can be mechanically infeasible or cost-intensive, mixed reality visualizations of robot arms can simply and cheaply enable these robots to gesture as they have a physical arm. The use of mixed reality deictic gestures has become a growing topic of interest in the HRI community[13] because such gestures allow better knowledge sharing between people and robots to improve shared mental models, calibrated trust and situational awareness [14]. Williams et al. [1] established a taxonomy of *mixed reality deictic gesture* (see Figure 1.1), including physical gestures, augmented reality (AR) annotations, and combinations thereof [1, 15]:

- Egocentric gestures: Physical gestures performed by the speaker.
- Allocentric gestures: AR gestures annotating the speaker's target referent from the addressee's perspective (e.g., an AR circle or arrow drawn around or pointing to an object).
- Perspective-free gestures: Gestures that change how all observers perceive the world, that are not tied to the perspective of any one agent (e.g., projecting a light on an object).
- Ego-sensitive allocentric gestures: AR gestures indicating the speaker's referent within the addressee's perspective but performed as if generated from the speaker's perspective (e.g., a robot pointing with a simulated AR arm).
- Ego-sensitive perspective free gestures: Gestures that change how all observers perceive the world, but that are performed as if generated from the speaker's perspective (e.g., projecting an arrow from the robot to its referent).

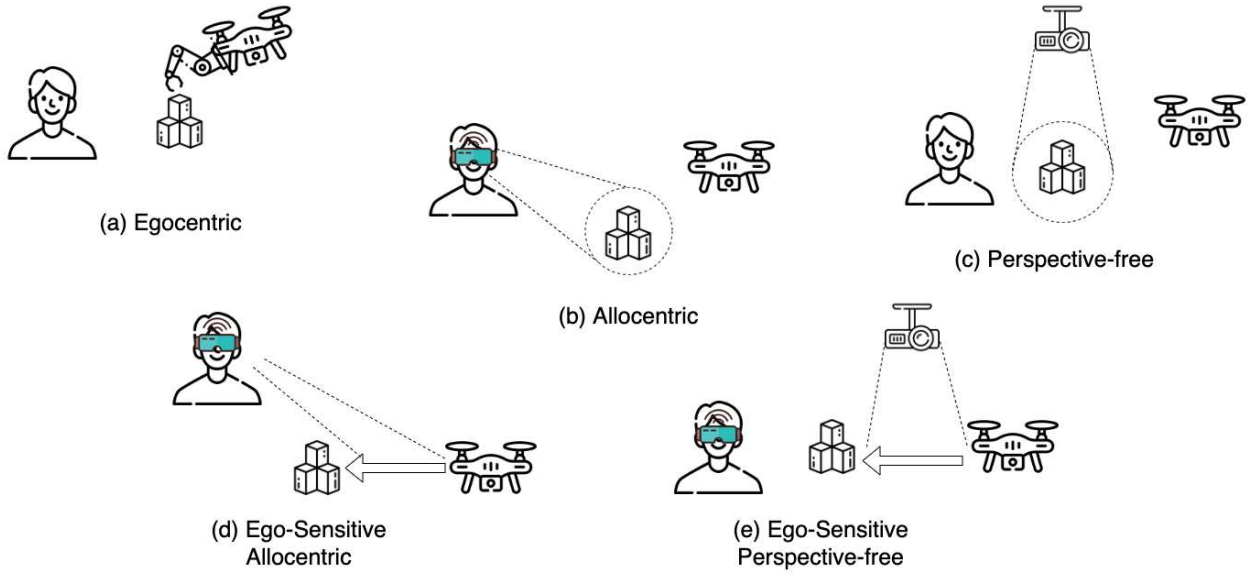

Figure 1.1 Categories of mixed reality deictic gestures proposed by Williams et al. [1]

Our previous work demonstrated the potential of *allocentric gestures* to improve the accuracy and efficiency of non-humanoid robot communication in a simulated video-base experiment [16]. When coupled with complex referring expressions, the gestures are considered more effective and likable than communication with the language alone [17]. However, a drawback of this simulated video-base experiment is the low ecological validity, as participants watching the videos had complete views of the entire experimental environment while users wearing an HMD might have a restricted field of view due to hardware limitations. My thesis presents the first demonstration of mixed reality deictic gesture generated on an actual HMD, the Microsoft HoloLens, in the context of task-based human-robot interaction.

Moreover, Hirshfield et al. [18] suggest several contextual factors that may influence the scenarios in which mixed reality deictic gestures can become helpful to human teammates: teammates' cognitive load may dictate whether they are capable of accepting new information; and their auditory and visual perceptual load may dictate the most effective modality to accompany or replace natural language. In contexts with high visual load, it might not

be advantageous to heavily rely on visual communication, and in context of high auditory or working memory load, relying on spoken language alone might not be effective [18]. Motivated by prior theoretical research in Multiple Resource Theory [19], the Perceptual Load Model [20], and the Dual-Target Search Model [21], these intuitions take into account the complex interplay between human cognitive load and perceptual processing load, which influence how human processes information and optimizes task performance. [19, 22, 23]. My thesis aims to address this question: *How do different types of mental workload impact the effectiveness of different robot communication modalities?*

The remainder of this thesis continues as follows. In chapter 2, we review the related work related to mixed reality/augmented reality for HRI and cognitive load estimation. In chapter 3, we formally define our experimental hypotheses and describe a human-subject experiment carried out with 36 participants to analyze those hypotheses. In chapter 4, we discuss our results, and discuss insights into the trade-offs between different forms of mixed reality communication in contexts with different types of workload in chapter 5. Lastly, in chapter 6, we describe the technical approach to implementing the mixed reality generation system.

CHAPTER 2

RELATED WORK

In this chapter, I review related work that seeks to integrate augmented reality in human-robot interaction and examine techniques to measure mental workload.

2.1 AR for HRI

For at least twenty-five years, there has been progress in integrating augmented reality in human-robot communication. Milgram et al. [24] (1993) first implemented the ARGOS (Augmented Reality through Graphics Overlays on Stereovideo) interface which overlaid a stereoscopic display with virtual information to allow a human operator to teleoperate a robot arm. The operator gathered stereoscopic information from the remote environment and used virtual landmarks to determine what command to send to the remote robot arm [24]. In 1999, Johnson et al. [25] developed an "EgoSphere", a 3D sphere around the robot which displayed various sensory data and events, to enhance the usability of their graphical user interface (GUI). Their enhanced GUI screen consists of landmark map, camera view, and the sensory "EgoSphere" to help users better visualize the robot's present state. AR has been used to enhance the human operator's control over the robot and improve the expressivity of human's view into the robot's internal states.

In the past decade, there has been a surge of research interest in using AR for training users how to operate robots [26], as well as communicating the perspectives, trajectories and intentions of robots. Amor et al. [27] used a projector to project instructions and to highlight task-relevant objects within an environment shared by humans and robots. This project does not use natural language generation, and visualizations are cast as part of the task environment instead of as part of the robot's communication [27]. Rosen et al. [28] developed a mixed reality interface to allow a robot to communicate its motion intent to a user. Likewise, this system only considers visual communication of the robot motion states.

Sibirtseva et al. [29] designed a system that allowed a robot to circle reference candidates in the user's AR head-mounted display as the user described a target referent [29]. Since the visualizations were generated from the robot's perspective to select referential candidates, this project shares some similarities to our research topic. However, we're interested in using AR as an active communication instead of passively responding to the human's communication. Reardon et al. [30] developed a robot that generated the trajectory for a human teammate to follow along and also highlighted the intended targets on the virtual path. These previous works do not focus on language-based robot communication when using the novel mixed reality platform to visually communicate shared goals and enable cooperative behaviors. In contrast, our laboratory has researched how to use AR as a *active* communication strategy, generated as gestures to accompany the natural language communication. [16, 17].

2.2 Mixed Reality Deictic Gestures

Williams et al. [1] suggested that a robot operating within a mixed reality environment can generate visual visualizations that can function as traditional deictic gestures. These visual gestures fall under the category of view-augmenting mixed reality interaction design elements in the Reality-Virtuality Interaction Cube framework of Williams et al. [31]. As mentioned in Chapter 1, these are also called allocentric visualizations or gestures, and they can be shown in the HMD or be projected onto the ground using a perspective-free projector. Recent research has explored the use of circles and arrows as allocentric gestures drawn over the target object. Sibirtseva et al. [29] compared three different modalities a robot could choose to communicate about the object of interest in addition to verbal request: projector, HMD (Microsoft HoloLens), and a side monitor as the control condition. Each modality overlaid the circles around the objects that the robot wanted the participant to pick up. They found that participants perceived the HMD condition to be more engaging, but most preferred the perspective free visualization due to its less intrusiveness. Williams et al. [16] explored the perceived effectiveness of allocentric mixed reality deictic gestures in multi-modal robot communication. Their experiment also used annotated circles to allow

robot to communicate its intent to the human. Their results showed that human perceived effectiveness and the perceived likability increased when mixed reality deictic gesture was used. However, this was a simulated video-based experiment. Our experiment follows up on this study by enabling mixed reality deictic gesture to be generated on an actual HMD.

Although prior research shows human preference for the projector-based AR over HMD [29], our study considers HMD because the human partner has to 1) carry out complicated physical tasks in a search-and-rescue operation such as running and climbing, and 2) keep moving instead of interacting with a video projector in one sit. In future work, we intend to combine the HMD with a lightweight physiology sensor so that the robot teammate can passively monitor the human mental workload and adapt its mixed reality deictic communication style while the human is wearing both devices.

2.3 Cognitive Load Measurement

2.3.1 Cognitive Load

Ideal task performance depends on the limited information processing capabilities of a human brain [22, 23]. Task performance degradation can occur if the task demand exceeds the brain's available processing capacity. [23, 32, 33]. In a classroom, for example, if an instructor presents too much information at once, the students may experience cognitive overload and much of that information may be lost.

2.3.2 Selective Attention and Perceptual Load

Selective attention refers to focusing on a specific aspect of a scene while ignoring other aspects [34]. Simons and Chabris [35] created the famous "the invisible gorilla" experiment to test selective attention. In their study, participants were asked to watch a video of a group of people–some dressed in white, some in black—passing basketballs around. While watching the video, the participants were asked to count the passes between the players dressed in white while ignoring the passes of those in black. Halfway through the video, a gorilla walked through the game, pounded his chest, then fled the scene. When the participants were asked

if they saw the gorilla, more than half of them missed it. This study demonstrated that when people selectively focus on something, they can become blind to the details they don't seek out. This failure of awareness is also known as "inattentional blindness." [35]

Perceptual load refers to the the amount of information involved in processing task-relevant stimuli determines the efficiency of selective attention [36]. If a human is asked to search for a specific water bottle on a table with hundreds of of similar looking bottles, he/she may experience high visual perceptual load. If a human is asked to listen for a particular instruction but there are multiple similar sounds playing in the background, he/she may experience high auditory perceptual load. In this research, we focus on how humans' visual and auditory perceptual loads are affected by or in turn affect the effectiveness of robots' mixed reality deictic gestures. The level of perceptual load in a task can influence selective attention, and a high load can affect the individual's ability to see obvious objects [37].

2.3.3 Multiple Resource Theory

Some context-aware, multi-tasks, and multi-modal systems today gain inspiration from Wickens [22]'s Multiple Resource Theory to design ways of presenting information to the user for effective use of human information processing resources [38]. The Multiple Resource Theory states that people have separate fixed-capacity resource pools for information processing. These resources can be categorized along three dimensions: 1) early vs. late information processing stage, 2) spatial vs. verbal information processing code, and 3) visual vs. auditory modality [19, 22]. Different pools of resources can be tapped simultaneously. Based on the complexity of the task, these resources will process information sequentially if the various tasks need the same pool of resources, or if the task needs different resources, they can be processed in parallel. A decrease in task performance indicates a shortage of these different resources and that the information processing limited capability has been reached. When the individual performs two or more tasks that require a single resource, a supply and demand problem occurs. Task error and slow performance will occur when a task that requires the same resource causes excess workload. Nachreiner [39] studied the com-

plex relationship between workload and job performance. This complex relationship shows that an increase in workload does not always cause a decrease in performance. Performance can be affected by both high and low workload [39]. If the users are under low workload, also known as underload, they might become bored, lose situation awareness, and reduce alertness. The Multiple Resource Theory helps designers to assess: 1) when tasks can be carried out simultaneously, 2) how tasks interfere with each other, and 3) how increasing in one task's difficulty may impact another task's performance. In our research, Wicken's Multiple Resource Theory helped us understand various forms of resources for processing information.

2.3.4 Dual-Tasking

Dual-tasks consist of a primary and a secondary task, with primary tasks often requiring long-term attention or involve a more important goal. Wicken's Multiple Resource Theory demonstrates how performance decreases under dual-task conditions [19, 22]. In a scenario when a human is driving and talking, s/he can easily drive and converse at the same time on a good day. However, in poor conditions, the driver will usually be under high cognitive and perceptive load, thus s/he will need to solely focus on the primary task: driving. Sawyer et al. [40] studied the effect of Google Glass, an AR HMD, on driving performance. They found that displaying virtual information on the HMD can reduce load in multitasking but it did not eliminate all distracting cognitive demands. Woodham et al. [41] demonstrated a climbing dual-task in which participants focused on climbing as the primary task and tried to recall the virtual words shown on HMD as the secondary task. Their results showed that the secondary task deteriorates the primary task's performance and vice versa. Participants climb slower when they have to recall information and their ability to recall information declines as they climb compared to sitting. An interesting observation from this study is that even though participants were not told to prioritize one task over another, climbing took precedence. The drop in performance for the secondary task (word recall) is greater than for the primary task (climbing).

2.3.5 Subjective Measures

The two commonly used subjective scales of cognitive load are the Paas Cognitive Load Scale and the NASA Task Load Index (TLX) [42]. The Paas scale asks the participants to rate their perceived intensity of their mental effort along a nine-point scale (1 = very low mental effort, 9 = very high mental effort) [33]. The NASA TLX requires participants to rate six subscales (range: 0 - 20): mental demand, physical demand, temporal demand, performance, effort, and frustration [43]. The total cognitive load is interpreted as the sum of six subscales (maximum: 120 points). The sum of cognitive load in both Paas Cognitive Load Scale and NASA TLX is intended to measure the three distinct types of cognitive load: intrinsic load (how complex the task is), extraneous load (how the task is presented), and germane load (how the learner processes the task for learning) [44–46]. Naismith et al. [42] compared these two scales and found that the intrinsic cognitive load is synonymous with mental effort in the Paas Scale and mental demand in the NASA TLX. The extraneous and germane cognitive loads, however, were not reflected in these two subjective methods. Our study does not consider the design of the learning materials (extraneous load) and the generation of knowledge structure in the learner's long-term memory (germane load). Because Naismith et al. [42] suggest that Paas Scale and TLX can be used interchangeably as measures of cognitive load, we choose to use TLX in our own research.

2.3.6 Physiological Measures

Advances in physiology allow for novel ways of measuring brain activity in real time through electrodes on the human scalp surface. Electroencephalography (EEG) and Functional near-infrared spectroscopy (fNIRS) are two portable and non-invasive methods to measure brain activity in natural environments. EEG measures electrical activity generated by synchronized activity of thousands of neurons in the cerebral cortex and strength of various oscillation frequencies such as delta, theta, and gamma [47]. It provides low spatial resolution which makes it difficult to determine the origin of the signal in the brain, but

offers excellent temporal resolution that can detect activity in sub-seconds [48].

Unlike EEG, fNIRS relies on the neurovascular coupling principle that measures changes in haemoglobin concentration and tissue oxygenation in the brain caused by neuronal activation [49]. Using the light sources placed on the scalp, fNIRS sends near-infrared light which is mainly absorbed by deoxygenated and oxygenated haemoglobin in the blood. These two provide relevant markers of the changes associated with neural activity in the brain [50]. As a lightweight and non-invasive device, fNIRS is gaining popularity in the Human-Computer Interaction community [51], as it offers several advantages such as greater spatial resolution, higher signal-to-noise ratio, and better practicality for use in normal working conditions [52]. The high spatial resolution of fNIRS enables the localization of different brain regions. Hirshfield et al. [18] designed a multiclass/multilablel fNIRS classifier for classifying the levels (high, medium, absent) of different types of cognitive load (Response Inhibitions, Working Memory, Spatial Attention, Visual Lexical Processing, and Visual Search), perceptual load (visual and auditory), and negative affect (frustration, stress). This has direct implications to HRI and our research project. The research done by our collaborator in capturing these different neurophysiological measures will help our future studies.

fNIRS has been used in various HRI research such as robot-assisted therapy, prosthetic control, engagement in education [53]. It is often used for two main purposes: 1) as an active or passive BMI interface for robot control, and 2) as an evaluation tool for measuring brain activity during an interaction [53]. However, fNIRS does pose some limitations such as interface design, signal processing, analysis and inference, and context-dependent hardware concerns [53].

2.4 AR and Cognitive Load

Several studies suggest the effect of augmented reality on human's cognitive load levels. Hou et al. [54] designed two human subject experiments to assess the participants' performance during a building block assembly task and their cognitive load when following traditional paper-based manual vs. animated AR guidance. The first experiment applied

the memory-base dual-task to study the participant's cognitive performance between AR and a traditional manual. Participants in the AR condition had shorter completion times, lower number of errors, and lower workload determined by the NASA TLX [55] compared to the physical manual condition. Their second experiment compared the learning curves of AR training with assembly manual training. The results showed participants had better task performance when they were trained using AR vs. manual training. Another study conducted by Gavish et al. [56] supported this finding. In their experiments, forty expert technicians were randomly assigned to four training groups in an electronic actuator assembly task: Control-VR (watching a filmed demo twice), VR (training with the VR platform once), Control-AR (training with the real actuator and assisted by the filmed demonstration once), and AR (training with the AR platform once). The study found the use of AR training improved the technicians' performance.

Funk et al. [57] employed an abstract Lego Duplo assembly task to compare the effects of using AR HMD instructions (Epson Moverio BT-200), handheld tablet instructions (non AR), paper instructions, and in-situ projected instructions (AR). They measured the participants' perceived cognitive load using the NASA-TLX questionnaire [55], number of errors, and various task competition times (e.g., time to locate the correct picking position, time to perform the pic, time it takes to understand instruction and place the picked part correctly, and time it takes to perform the assembly). They counterbalanced the order of the conditions according to the Balanced Latin Square to prevent the carryover effects of a previous experimental condition impacting the current experimental condition. Their results suggested that participants experience lower self-reported cognitive load and made few errors using the in-situ projected instructions compared to instructions provided by the AR HMD. Additionally, participants took less time to assemble parts using the projection, and took more time to locate the positions using the HMD. The finding is not surprising as the Epson Moverio BT-200 has a very narrow field of view, and participants' field of views were also blocked by the instructions.

Woodham et al. [41] explored the dual-task costs of users doing a visual communication task using an HMD while climbing an indoor climbing wall. While the participant were climbing and wearing an HMD, they were shown the virtual words on the HMD with and without auditory warnings. In the other two conditions, the participants sat and saw the words presented on the HMD with and without auditory warnings. Motion data was captured throughout the experiment. Their findings demonstrated a performance decrease in both climbing and word recalling tasks as participants slowed down around word presentations on the HMD. Participants in the climbing task had lower word recall performance than those in the seated task, suggesting that complex physical activity, like climbing, hinders memory rehearsal, and cognitive tasks also hinder physical performance. They also found that visual stimuli is more disruptive to the climbing performance than auditory stimuli. The implication of this study is that cognitive HMD tasks that require later recall should be avoided when users are climbing or performing other complex physical tasks [41].

Küçük et al. [58] compared the effects on medical students' academic performance and cognitive load using a mobile AR application against a printed textbook. The students' cognitive load was measured using the nine-point Likert Scale developed by Paas et al. [33]. The academic performance test was measured using a thirty multiple choice quiz. They found that students who used the AR mobile application to study anatomy experienced lower cognitive load and achieved higher academic performance.

2.5 Robotics and Neurophysiology

Over the past two decades, brain-machine interface (BMI) researchers have been using neurophysiological measurements of brain activity to find ways to manipulate robots. Chapin et al. [59] (1999) trained rats to press the lever for a water reward and then implanted multi-electrode recording arrays in them. They then derived mathematical transformations, including neural networks, to convert rats' multineuron signals into real-time signals for robot arm control [59]. Wessberg et al. [60] (2000) implanted microwire arrays in monkeys and successfully converted cortically derived signals of the primates, which involved various

aspects of motor control, to real-time control of robotic device, both locally and online. Most of these invasive BMIs were tested primarily on laboratory animals, and few were tested on human subjects for safety reasons. Patil et al. [61] obtained acute ensemble recordings from subthalamic nucleus and thalamic motor areas from human patients during deep brain stimulator surgery and demonstrated that these signals could be used to predict force-task performance.

Non-invasive BMIs such as EEG certainly have the advantage of not exposing humans to the dangers of invasive surgical procedures but compared to those the implanted electrodes, non-invasive BMIs have lower resolution, slower rate of transfer, and increased noise due to measurements on the scalp [62–64]. Non-invasive BMIs have thus been tested more frequently with human subjects and show their promise in providing simple communication but shortcomings in more complex operations [63]. Millan et al. [64] (2004) demonstrated an EEG-based BMI and machine learning methods capable of controlling a robot's rotation in indoor environments. Galán et al. [65] (2008) presented non-invsive BMIs for brain-actuated wheelchair driving.

The BMI systems which require the users to consciously and directly control their brain activity to control an application are categorized as *active* systems [66]. These *active* systems, however rarely generalize to multiple users, because they often require extensive system and user tuning and training [67, 68]. Zander and Kothe [66] (2011) categorized previous works which do not involve active control but which implicitly monitor human brain activity to augment human-machine interaction as *passive* systems. Szafir and Mutlu [69] (2012) explored how to use neurophysiological information to make robot behaviors adaptive. In their experiment, participants interacted with an adaptive robot which monitored their engagement using EEG signals in real time and adapted its verbal and nonverbal immediacy cues. The study found that the robot's adaptation based on the participant's neurophysiological information improved their recall ability and the overall learning experience. Around the same time, Girouard et al. [70] (2013) designed a passive system called the online fNIRS analysis

and classification (OFAC) capable of analyzing brain signal in real time and using machine learning to classify different affective and workload states. The researchers conducted two experiments. The first experiment compared the offline analysis to the real-time analysis, and the result showed a 12 percent decrease in classification accuracy and a minimum of 12 examples of each class in order to achieve stable accuracy. The second experiment evaluated the ability of OFAC to process cognitive states and to adapt the game interfaces accordingly in real time. Although there was not a statistically significant difference in accuracy between the adaptive environments, their findings indicated that user satisfaction was mostly neutral and positive.

Although EEG has finer temporal resolution in the sub-second scale, our use case does not require a constant data stream. Each of our communication strategies, including mixed reality deictic gestures and language, takes more than a few seconds for the robot to execute. We thus prefer fNIRS's greater spatial resolution to adapt the robot's communication styles based on more load types and Wicken's Multiple Resource Theory. To summarize, the ability of fNIRS to measure various types of mental workload and emotion, as well as being non-invasive and impervious to user movement, made it an ideal tool for our use case of robotic adaptation.

CHAPTER 3

EXPERIMENT

In this chapter, we present the design of a human-subject experiment to study how human teammates perceive augmented (allocentric) mixed reality deictic gestures, and how such gestures interact with the teammates' perceptual or cognitive load (as measured with fNIRS). In particular, we are interested in these effects when allocentric mixed reality deictic gesture is compared to or paired with complex natural language expressions.

3.1 Hypotheses

Specifically, this experiment was designed to test the following hypotheses, which formalize the intuitions of Hirshfield et al. [18].

H1 Users under high *visual perceptual load* will perform quickest when robots rely on complex natural language without the use of mixed reality deictic gestures.

H2 Users under high *auditory perceptual load* will perform quickest when robots rely on mixed reality deictic gestures without the use of complex natural language.

H3 Users under high *working memory load* will perform quickest when robots rely on mixed reality deictic gestures without the use of complex natural language.

H4 Users under *low overall load* will perform quickest when robots rely on mixed reality deictic gestures paired with complex natural language.

3.2 Task Design

To assess these hypotheses, we designed a human-subject experiment in which participants interacted with a language-capable robot while wearing the Microsoft HoloLens, over a series of trials, with the robot's communication style and the user's cognitive load systematically varying between trials.

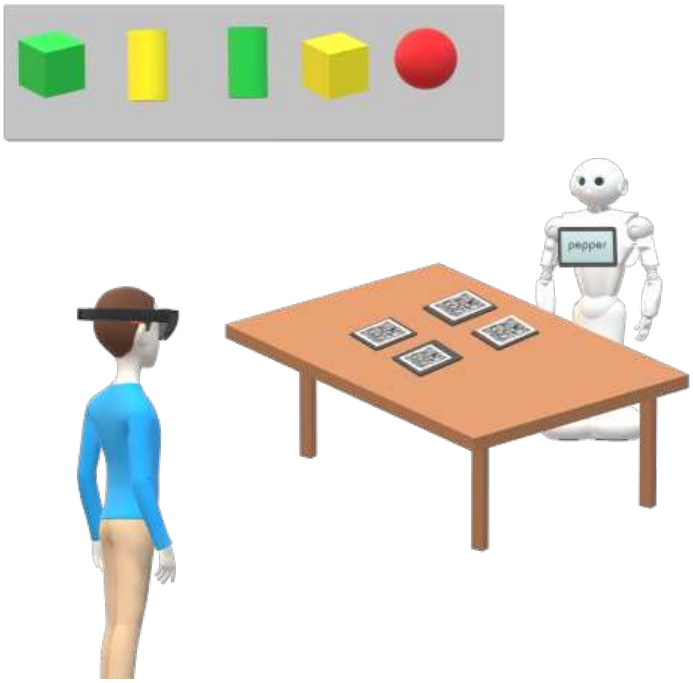

Figure 3.1 Our experimental setup.

The task used for this experiment employed a dual-task paradigm oriented around a tabletop pick-and-place task. Participants view this task through the Microsoft HoloLens, allowing them to see virtual bins overlaid over a set of fiducial markers on the table, as well as a panel of blocks above the table that changes every few seconds (Figure 3.2). As shown in Figure 3.1, the Pepper robot is positioned behind the table, ready to interact with the participant.

3.3 Primary Task

The user's *primary task* is to look out for a particular block in the block panel (selected from among *red cube*, *red sphere*, *red cylinder*, *yellow cube*, *yellow sphere*, *yellow cylinder*, *green cube*, *green sphere*, *green cylinder*[1]). These nine blocks were formed by combining

[1]These block colors were chosen for consistent visual processing, as blue is well known to be processed differently within the eye due to spatial and frequency differences of cones between red/green and blue. This did mean that our task was not accessible to red/green colorblind participants, requiring us to remove from our dataset the data of several colorblind participants.

three colors red, yellow, green with three shapes cube, sphere, cylinder. Whenever they see this target block, their task is to pick-and-place it into any one of a particular set of bins. For example, a user might be told that whenever they see a *red cube* they should place it in bins *two or three*.

Two additional factors increase the complexity of this primary task. First, in order to force participants to remember the full set of candidate bins, rather than just one particular bin from that set, at every point during the task one random bin is marked as unavailable (with the disabled bin changing each time a block is placed in a bin). Second, to allow us to examine auditory load, the user hears a series of syllables playing in the task background (selected from among *bah*, *beh*, *boh*, *tah*, *teh*, *toh*, *kah*, *keh*, *koh*). These nine syllables were formed by combining three consonant sounds *b,t,k* with three vowel sounds *ah,eh,oh*. The user is given a target syllable to look out for, and told that whenever they hear this syllable, the bins that they should consider to place blocks in should be exchanged with those they were previously told to avoid. For example, if the user's target bins from among four bins are bins two and three, and they hear the target syllable, then future blocks will need to be placed instead into bins one and four.

3.4 Secondary Task

Three times per experiment block, the participant encounters a secondary task, in which the Pepper robot interjects and asks the participant to move a particular, currently visible block, to a particular, currently accessible bin.

3.5 Experimental Design

To prevent the carryover effect that carries over from one experimental condition to another, we used a Latin square counterbalanced within-subjects experimental design with two independent variables serving as within-subjects factors: Cognitive Load and Communication Style.

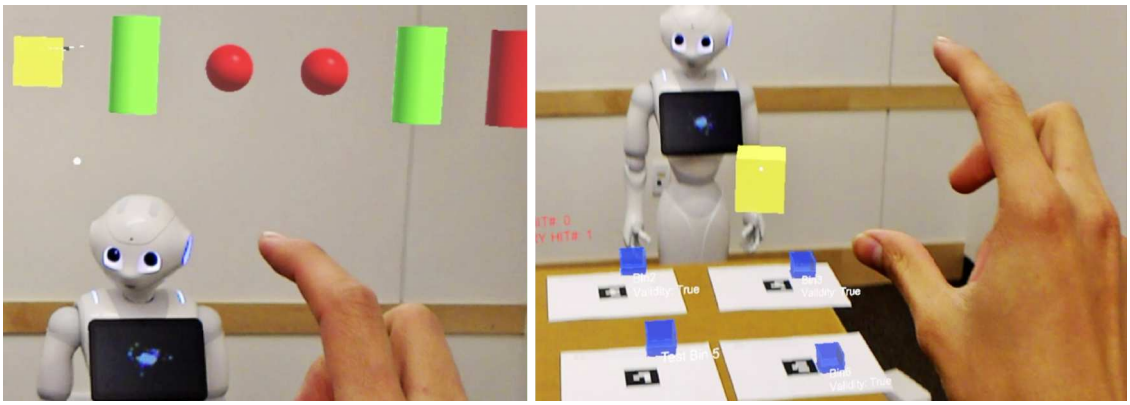

Figure 3.2 Experiment in progress

3.5.1 Cognitive Load

Our first independent variable, cognitive load was manipulated through our primary task. Following Beck and Lavie [71], we manipulated communication style by jointly manipulating memory constraints and target/distractor discriminability (cp. [37]), producing four different load profiles: one in which all load was considered low; one in which only working memory load was considered to be high, one in which only visual perceptual load was considered to be high, and one in which only auditory perceptual load was considered to be high.

Working memory load was manipulated as follows: In the high working memory load condition, participants were required to remember the identities of three target bins out of a total of six visible bins, producing a total memory load of seven items when including the two properties of the target block (shape and color) and the two properties of the target syllable (consonant and vowel). In all other conditions, participants were only required to remember the identities of two target bins out of a total of four visible bins, producing a total memory load of six items.

Visual perceptual load was manipulated as follows: In the high visual perceptual load condition, the target block was always difficult to discriminate from distractors due to sharing of one common property with all distractors. For example, if the target block was a red cube, all distractors would be either red or cubes (but not both). In the low visual perceptual load

condition, the target block was always easy to discriminate from distractors due to sharing no common properties with any distractors. For example, if the target block was a red cube, no distractors would be red or cubes.

Auditory perceptual load was manipulated as follows: In the high auditory perceptual load condition, the target syllable was always difficult to discriminate from distractors due to sharing of one common property with all distractors. For example, if the target syllable was *kah*, all distractors would either start with *k* or end with *ah* (but not both). In the low auditory perceptual load condition, the target syllable was always easy to discriminate from distractors due to sharing no common properties with any distractors. For example, if the target syllable was *kah*, no distractors would either start with *k* or end with *ah*.

3.5.2 Communication Style

Our second independent variable, communication style, was manipulated through our secondary task. Following Williams et al. [16] and Williams et al. [72], we manipulated communication style by having the robot exhibit one of three behaviors:

During experiment blocks associated with the *complex language* communication style condition, the robot with which participants interacted referred to objects using full referring expressions needed to disambiguate those objects.

During experiment blocks associated with the *complex language + AR* communication style condition, the robot with which participants interacted referred to objects using full referring expressions needed to disambiguate those objects (e.g., "the red sphere"), paired with a mixed reality deictic gesture (an arrow drawn over the object to which the robot was referring).

During experiment blocks associated with the *simple language + AR* communication style condition, the robot with which participants interacted referred to objects using minimal referring expressions (e.g., "that block"), paired with a mixed reality deictic gesture (an arrow drawn over the object to which the robot was referring).

Following Williams et al. [16] and Williams et al. [72], we did not examine the use of simple language without AR, as that communication style does not always allow complete referent disambiguation, resulting in the user needing to ask for clarification or guess at random between ambiguous options.

3.6 Measures

We expected performance improvements to manifest in our experiment in four different ways: task accuracy, task reaction time, perceived mental workload, and perceived communicative effectiveness.

These aspects of performance were measured as follows:

Accuracy was measured for both primary and secondary tasks by logging which virtual object participants clicked on, and determining whether or not this was the object intended by the task or by robot.

Reaction time was measured for both primary and secondary tasks by logging time stamps at the moment participants interacted with virtual objects (both blocks and bins). In the primary task, reaction time was measured as the time between placement of the previous primary target block and picking of the next primary target block. In the secondary task, reaction time was measured as the time between the start of Pepper's utterance and the placement of the secondary target block.

Perceived mental workload was measured using a NASA Task Load Index (NASA TLX) survey[43] administered at the end of each experiment block.

Perceived communicative effectiveness was measured using the modified version of the Gesture Perception Scale [11] previously employed by Williams et al. [16, 72], which was delivered along with the NASA TLX Survey at the end of each experiment block.

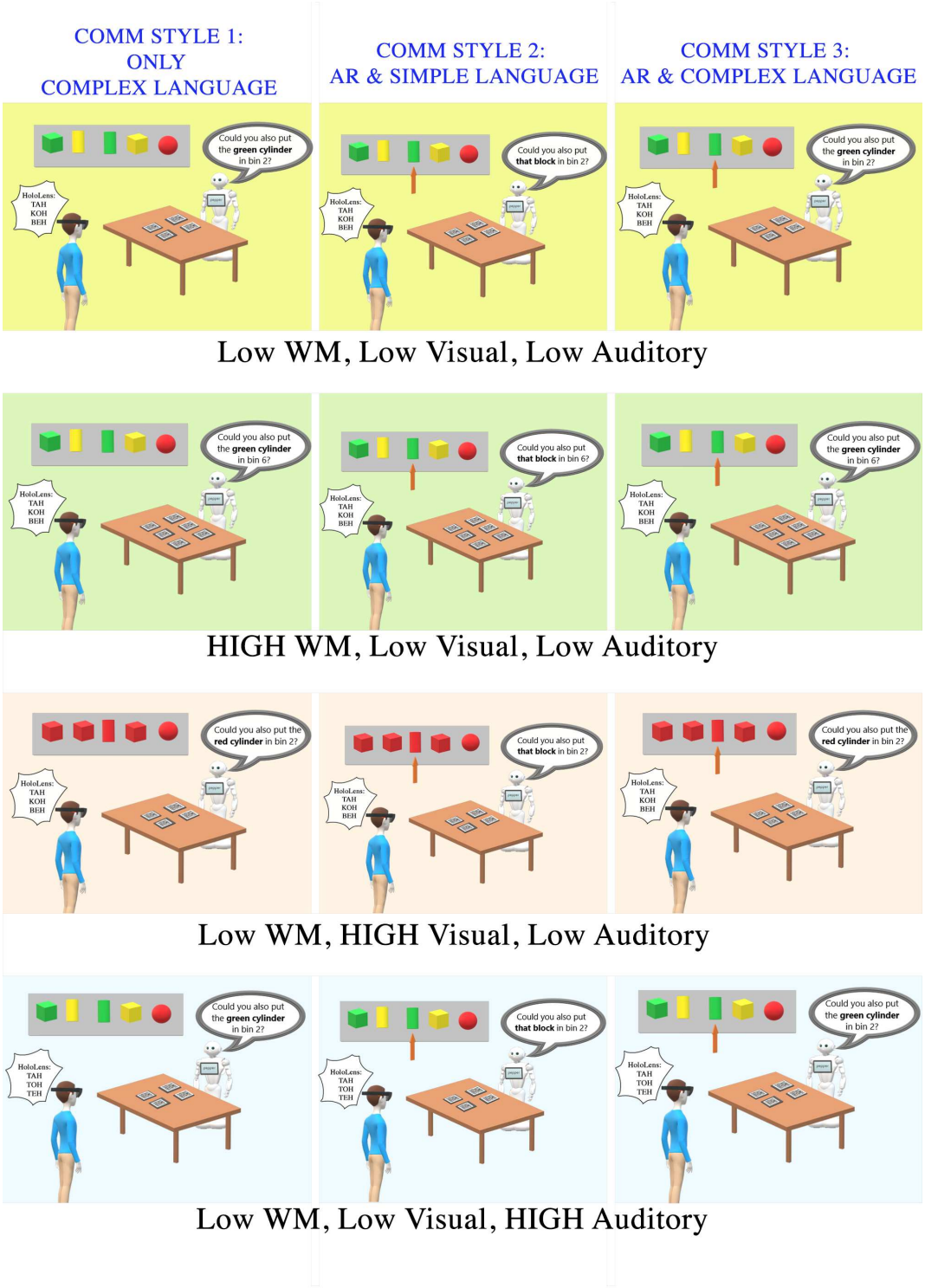

Figure 3.3 Twelve within-subject conditions (4 workload profiles x 3 communication styles)

3.7 Procedure

Upon arriving at the lab, providing informed consent, and completing a demographic and visual capability survey, participants were introduced to the task through both verbal instruction and an interactive tutorial.

Figure 3.4 Participants were asked to go through the Tutorial before starting the experiment

The tutorial scene provides text and visuals that walk the participant through how a round in the experiment will function. When the participant starts the tutorial, they see a panel with text-instructions, a row of blocks, and four bins (Fig. Figure 3.4). Participants are walked through how to use the HoloLens air tap gesture to pick up blocks and put them in bins through descriptive text and an animation showing an example air tap gesture, and informed of task mechanics with respect to both target/non-target bins and temporarily disabled grey bins. Participants then start to hear syllables being played by the HoloLens.

When the target syllable *teh* plays, the target and non-target bins switch. Each bin on screen is labeled as a 'target' or 'non-target', in order to help the participant understand what is happening when the target syllable plays. These labels are only shown in the tutorial and participants are reminded that they will have to memorize which bins are targets for the actual game. At the end of the tutorial the participant has to successfully put a target block in a target bin three times before they can start the experiment.

After completing training, participants engaged in each of the twelve (Latin square counterbalanced) experiment blocks formed by combining the four cognitive load conditions and the three communication style conditions, with surveys administered after each experiment block (see Figure 3.3).

3.8 Participants

36 participants were recruited from Colorado School of Mines (31 M, 5 F), ranging in age from 18 to 32. None had participated in any previous studies from our laboratory. Each participant went through 12 within-subject rounds/conditions (see Figure 3.3) and each round took 90 seconds. After every round, participants were asked to take at least 30 seconds to take a break and fill out our subjective survey. Figure 3.5 shows the timeline of our experiment.

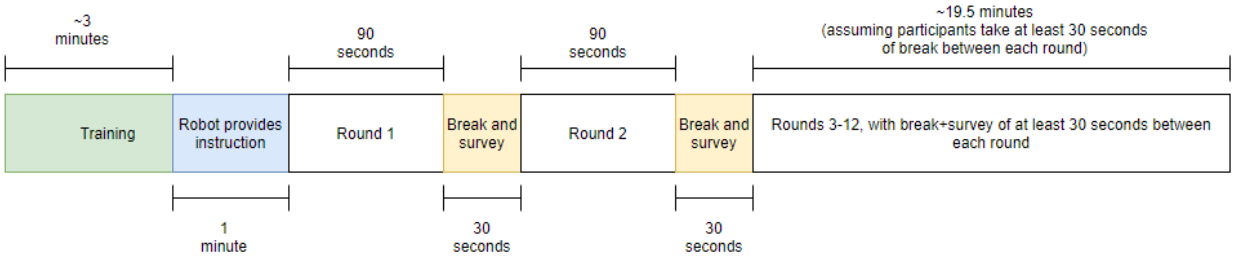

Figure 3.5 Experiment Protocol and Phases

CHAPTER 4

RESULTS

Data analysis was performed within a Bayesian analysis framework using the JASP 0.11.1 [73] software package, using the default settings as justified by Wagenmakers et al. [74]. For each measure, a repeated measures analysis of variance (RM-ANOVA) [75–77] was performed, using communication style and cognitive load as random factors. Baws factors [78] were then computed for each candidate main effect and interaction, indicating (in the form of a Bayes Factor) for that effect the evidence weight of all candidate models including that effect compared to the evidence weight of all candidate models not including that effect. When sufficient evidence was found in favor of a main effect, the results were further analyzed using a post-hoc Bayesian t-test [79, 80] with a default Cauchy prior (center=0, r=$\frac{\sqrt{2}}{2}$=0.707). When sufficient evidence was found in favor of an interaction effect, the results were further evaluated using a series of post-hoc paired-samples t-tests each category of cognitive load.

4.1 Reaction Time

4.2 Secondary Task

Our results provided extreme evidence[2] in favor of effects of both communication style (Bf 3.109e29) and cognitive load (Bf 9.881e9) on secondary task reaction time, as shown in Figure 4.1 and Figure 4.2, as well as an interaction between communication style and cognitive load (Bf. 1.160e12) on reaction time, as shown in Figure 4.3.

Post-hoc analysis of the main effect of communication style on secondary task reaction time revealed significant differences specifically between the use of complex language alone ($\mu = 8.116sec$, $\sigma = 0.543sec$) and both complex language + AR ($\mu = 7.399sec$, $\sigma = 0.610sec$, Bf 2.955e21) and simple language + AR ($\mu = 7.501sec$, $\sigma = 0.545sec$, Bf 9.396e15),

[2]Bayes Factors (Bf) above 100 indicate extreme evidence in favor of a hypothesis [81, 82]. Here, for example, our Baws Factor Bf of 7.024e25 suggests that our data were 7.024e25 times more likely to be generated under models in which communication style is included than under those in which it is not.

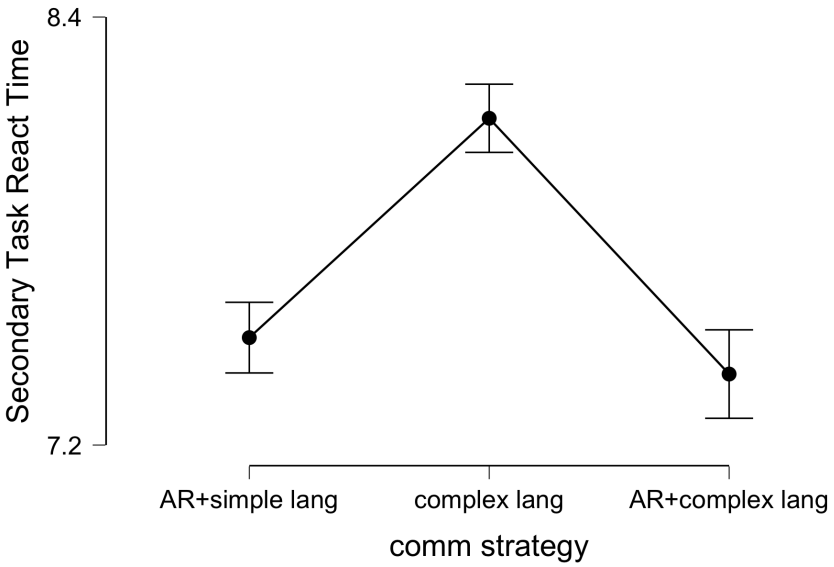

Figure 4.1 Effect of communication strategy (complex language + AR vs. complex language vs. simple language + AR) on secondary task reaction time.

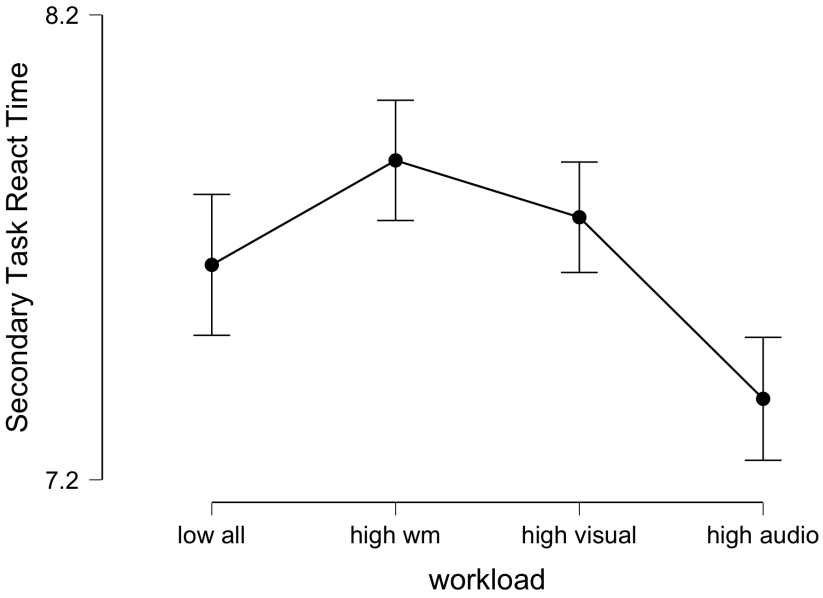

Figure 4.2 Effect of workload (Low All) vs. (High Visual) vs. (High Auditory) vs. (High Working Memory) on participant's secondary task reaction time.

with anecdotal evidence against a difference between complex language + AR and complex language alone (Bf = .46 in favor of an effect; 1/.46 = Bf 2.14 against an effect)

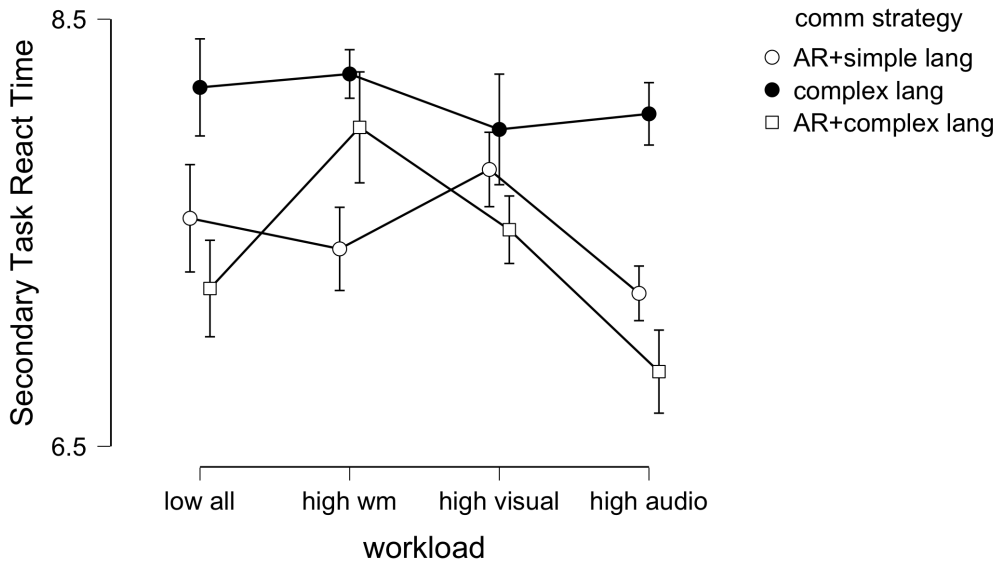

Figure 4.3 Effect of both workload and communication strategy on participant's secondary task reaction time.

This yields a preference ordering where complex language < (simple language + AR = complex language + AR) when cognitive load is not considered.

Post-hoc analysis of the main effect of cognitive load on secondary task reaction time revealed significant differences specifically between conditions with high auditory perceptual load ($\mu = 7.374sec$, $\sigma = 0.454sec$) and all other conditions, i.e., low overall load ($\mu = 7.662sec$, $\sigma = 0.684sec$, Bf 2931.437), high visual perceptual load ($\mu = 7.765sec$, $\sigma = 0.574sec$, Bf 283407.874), and high working memory load ($\mu = 7.887sec$, $\sigma = 0.551sec$, Bf 1.343e9), as well as between conditions with high working memory load and those with low overall load (Bf 13.381).

This yields a preference ordering where high auditory perceptual load < ((low overall load < high working memory load) = high visual perceptual load) when communication style is not considered.

Post-hoc analysis of the interaction effect between communication style and cognitive load on secondary task reaction revealed the following additional findings:

Low Overall Load:

Extreme evidence was found under low overall load between each pair of communication strategies: simple language + AR ($\mu = 7.568sec$, $\sigma = 0.732sec$) vs complex language alone ($\mu = 8.195sec$, $\sigma = 0.685sec$, Bf 8.995e6); simple language + AR vs complex language + AR ($\mu = 7.253sec$, $\sigma = 0.654sec$, Bf 703110.101); complex language alone vs complex language + AR Bf 1.281e13.

This yields a preference ordering where complex language alone < simple language + AR < complex language + AR in the low overall load condition.

High Working Memory Load:

Extreme evidence was found under high working memory load between simple language + AR ($\mu = 7.439sec$, $\sigma = 0.565sec$) and both complex language alone ($\mu = 8.240sec$, $\sigma = 0.327sec$,Bf 1.080e7) and complex language + AR ($\mu = 7.988sec$, $\sigma = 0.746sec$, Bf 2076.594).

This yields a preference ordering where (complex language alone = complex language + AR) < simple language + AR in the high working memory load condition.

High Visual Perceptual Load

Moderate to extreme evidence was found under high visual perceptual load between complex language + AR ($\mu = 7.506sec$, $\sigma = 0.456sec$) and both complex language alone ($\mu = 7.997$, $\sigma = 0.747sec$, Bf 1449.784) and simple language + AR ($\mu = 7.781sec$, $\sigma = 0.508sec$, Bf 5.336).

This yields a preference ordering where (simple language + AR = complex language alone) < complex language + AR in the high visual perceptual load condition.

High Auditory Perceptual Load

Extreme evidence was found under high auditory perceptual load between each pair of communication strategies (simple language + AR ($\mu = 7.219sec$, $\sigma = 0.367sec$) vs complex language alone ($\mu = 8.050sec$, $\sigma = 0.421$, Bf 7.374e6); simple language + AR vs complex language + AR ($\mu = 6.859sec$, $\sigma = 0.560sec$, Bf 35.760); complex language alone vs complex language + AR (Bf 1.126e13).

This yields a preference ordering where complex language alone < simple language + AR < complex language + AR in the high auditory perceptual load condition.

4.3 Primary Task

Strong evidence was found *against* any effects of communication style or cognitive load on primary task reaction time (All Bfs > 20 against an effect). When we plotted the description plot to observe the accuracy mean of the primary task (Figure 4.4), we noticed that participants performed the primary task well when they were under low workload. Their accuracy decreased when other types of mental workload were manipulated. However, the Bayes Factors determined that these primary task's results were not statistically significant.

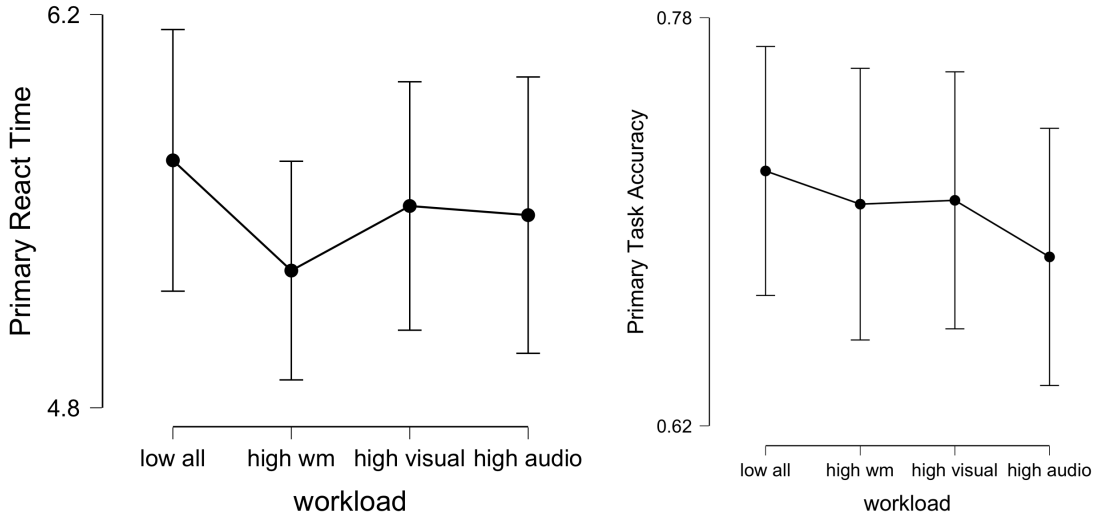

Figure 4.4 Effect of workload on participant's primary task's accuracy and reaction time. Results are not statistically significant

4.4 Accuracy

Strong evidence was found *against* any effects of communication style or cognitive load on primary or secondary task accuracy (All Bfs > 27 against an effect).

4.5 Perceived Mental Workload

Anecdotal to strong evidence was found *against* any effects of communication style or cognitive load on perceived mental workload (Bfs between 22.43 and 40.91 against an effect). Analysis (Figure 4.5) also showed no significant difference between means.

4.6 Perceived Communicative Effectiveness

Anecdotal to strong evidence was found *against* any effects of communication style or cognitive load on perceived communicative effectiveness (Bfs between 2.23 and 83.33 against an effect on all questions). Analysis (Figure 4.6) also showed no significant difference between means.

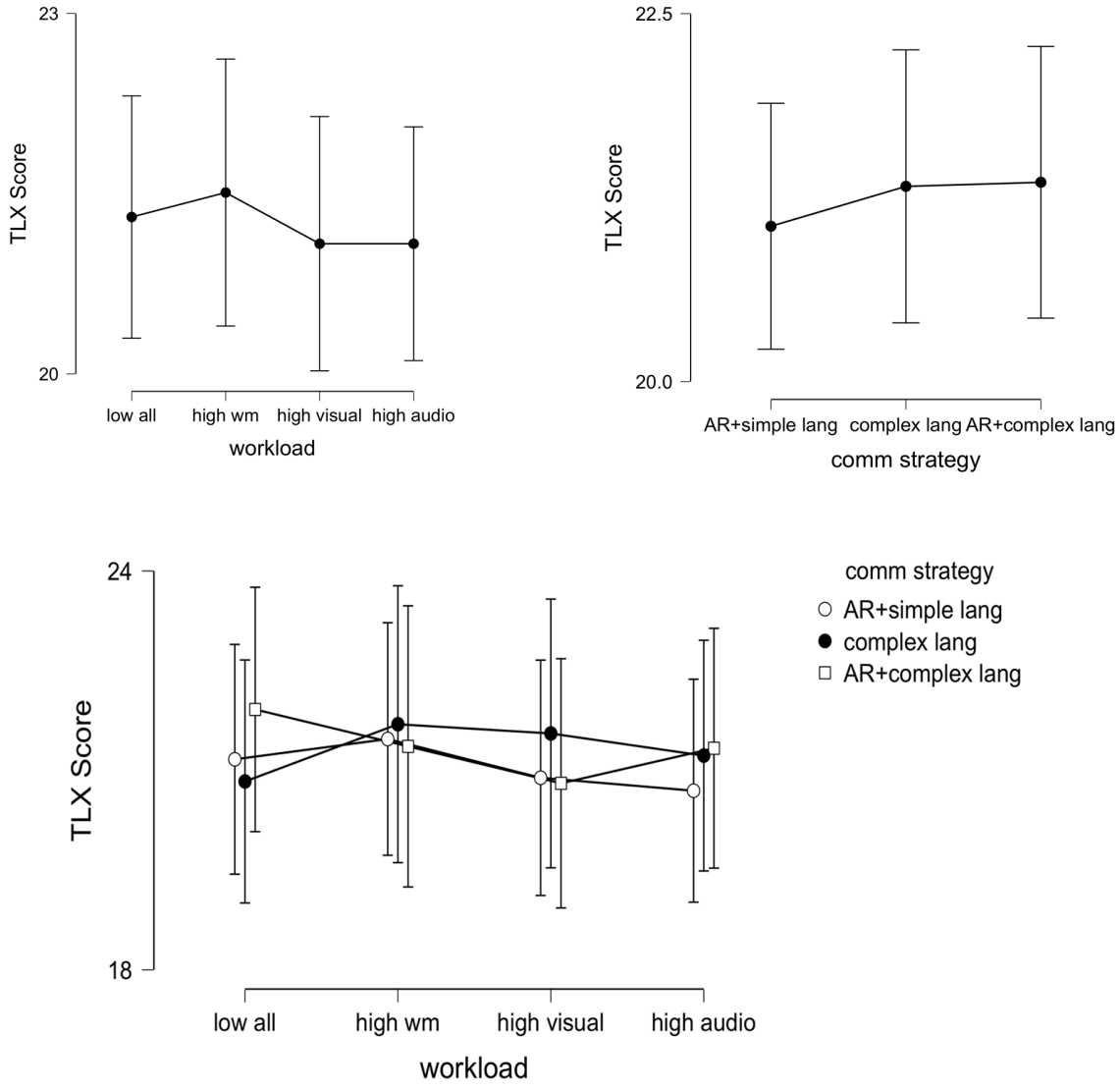

Figure 4.5 Effect of both workload and communication strategy on participant's perceived mental workload

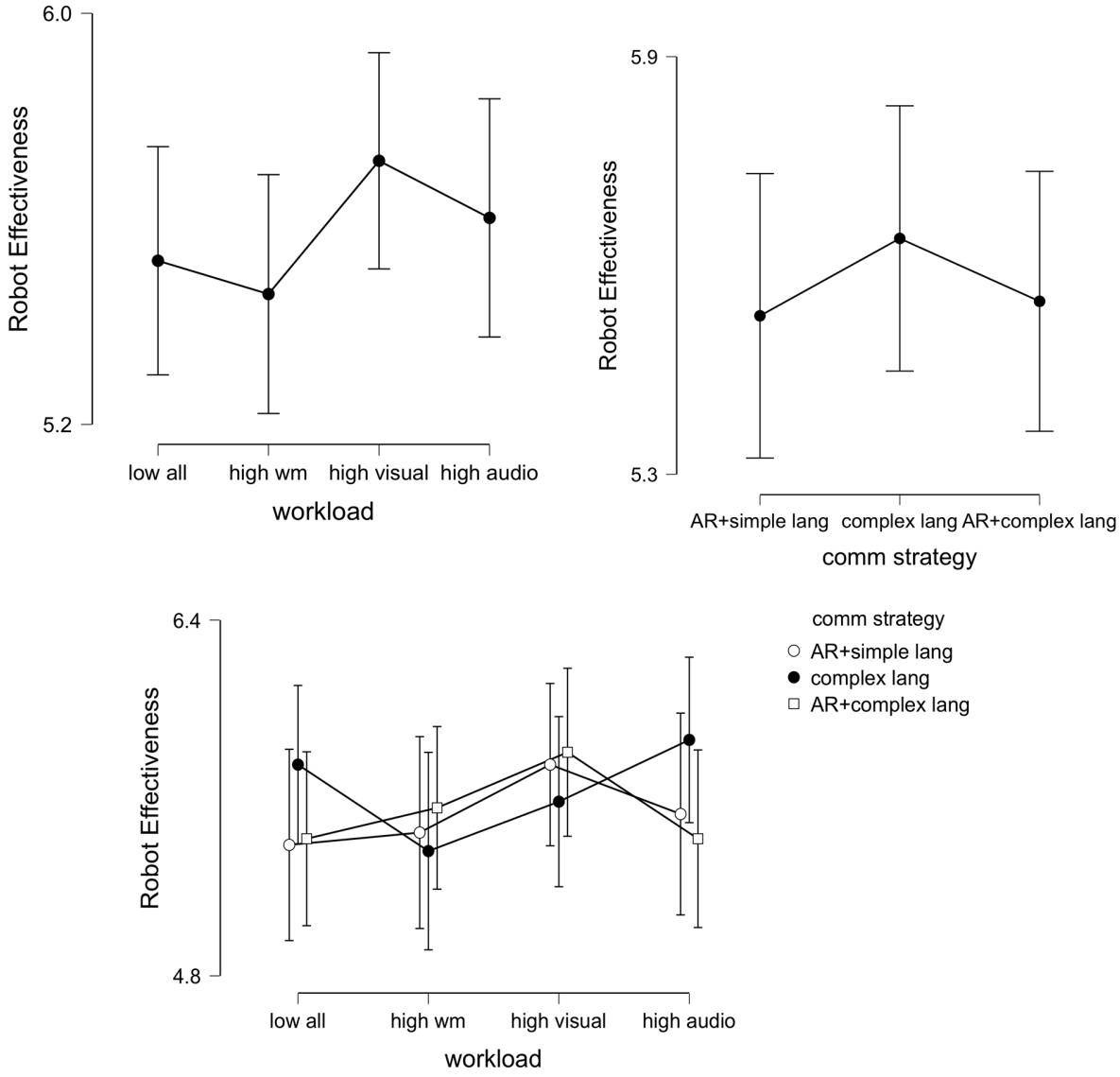

Figure 4.6 Effect of both workload and communication strategy on participant's perceived robot's communication effectiveness

CHAPTER 5
DISCUSSION AND CONCLUSION

Our results suggest that, although humans may not be aware of differences in their task time or mental workload when different mixed reality robotic communication styles are used, or when they are under different types of cognitive load, both of these factors do in fact influence the speed at which they are able to accomplish tasks.

First, our results suggest that different *types* of mental workload do, unsurprisingly, impact task time, with participants under low overall load reacting more quickly than participants under high working memory load. What *is* surprising is that participants under high auditory load clearly demonstrated the fastest reaction times overall. It is not yet clear how to interpret this result, but it is possible that this effect is due to individuals generally responding faster to auditory stimuli that visual [83].

Second, our results suggest, unsurprisingly, that different communication strategies impact task time. In fact, our results exactly match what we observed in previous experiments [72]: participants demonstrate slower reaction times when complex language alone is used, with no clear differences between simple and complex language when it is augmented with a mixed reality deictic gesture.

Finally, our results suggest a complex interplay between communication style and cognitive load. Specifically, our results suggest that while using complex language + AR resulted in the best task time in most workload conditions (an encouraging result given that our previous work has shown that participants find robots most *likeable* when they use this communication style [16]), this does not hold true when users are under high working memory load. Rather, when users are under high working memory load, it is best to use simple language + AR, to avoid overloading participants.

Overall, these results support hypotheses H3 and H4, but fail to support hypotheses H1 and H2. Our original expectation was that the differences between communication styles under different cognitive load profiles would primarily be grounded in whether communication style was overall visual or overall auditory. On the contrary, what we observed is that visual augmentations are *always* helpful, and differences in effectiveness between communication styles depend entirely on whether or not the user is under high cognitive load.

While we observed clear impacts of workload profiles on task time, participants did not demonstrate any differences in perceived workload or perceived effectiveness. It could be the case that the differences in reaction time simply were not large enough for participants to notice: the observed differences were on the order of one second of reaction time when overall reaction time was around 7.5 seconds. Participants may simply not have noticed a 15% speed increase in certain conditions, or may not have attributed it to the robot.

This could also be the case due to overall task difficulty. Although participants' TLX scores had a mean value of approximately 21 out of 42 points in all conditions (i.e., the data was nearly perfectly centered around "medium" load), analysis of individual performance trajectories demonstrates that the task was sufficiently difficult that many participants experienced catastrophic primary task shedding, often immediately after a primary task (likely due to missing an auditory cue while dealing with a secondary task). As illustrated in Figure 5.1 - Figure 5.12, task time and task accuracy varied significantly between participants. All twelve condition plots show similar results to what we observed here. In these figures, the dark black X markers represent the time the robot started uttering secondary task requests. The blue X markers represent the time the human successfully placed a secondary target cube in a secondary target bin. The pink X markers represent the unsuccessful secondary tasks. As can be seen, most participants performed well on the primary task (resulting in many green dots) up until immediately after the first or second secondary task request. As can also be seen, when participants made a mistake, except in cases where the error fell between secondary task initiation and completion, they often failed to recover from the fail-

ure. In each of the following figures, we calculated the average and median time participants took until the first incorrect placement. Most people made mistakes right after the first 22.5 seconds. In every round, we additionally highlighted the top 3-5 highest performers.

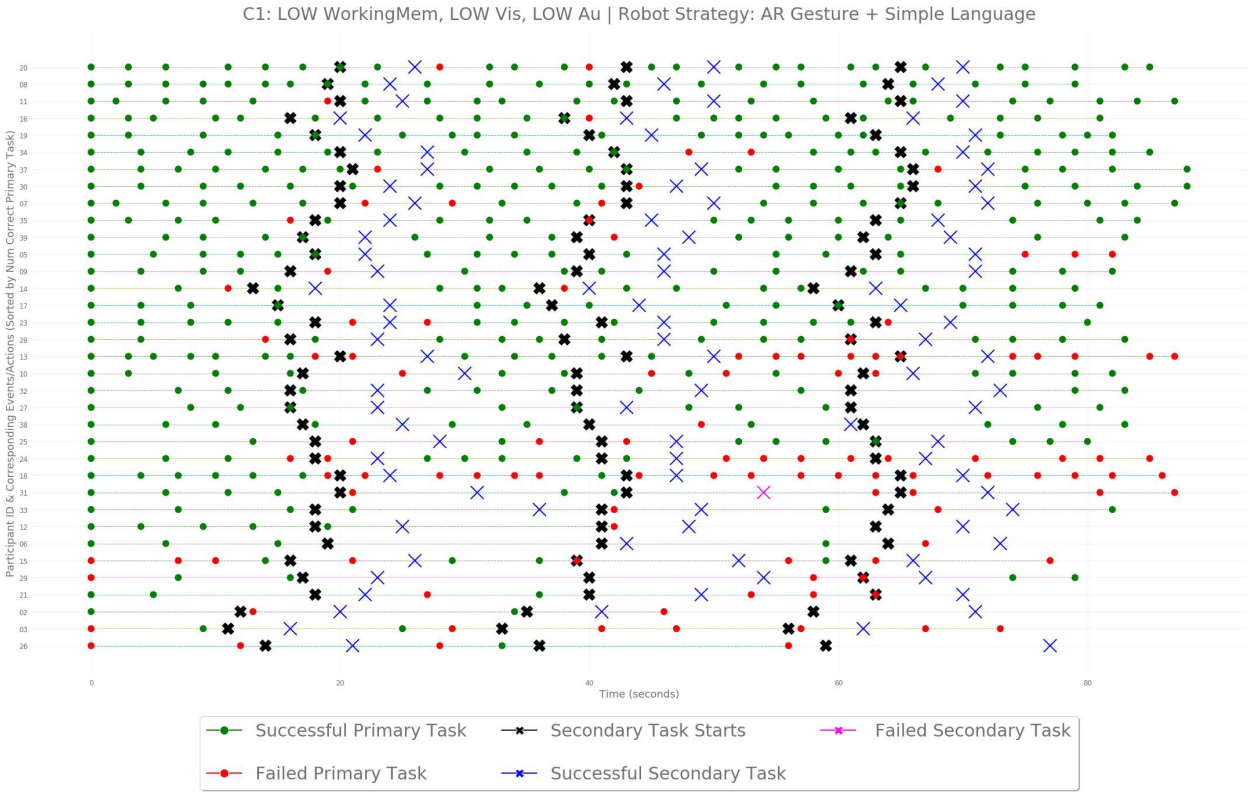

Figure 5.1 Visualization of participant performance in the *AR + Simple Language / Low All*

In condition 1 where participants were under low overall workload and the robot used AR + Simple language, the following facts were observed:

- 5 participants completed the primary tasks without any errors: [8, 17, 27, 32, 19].
- The average time to the first failed primary task: 26.067 seconds.
- The median time to the first failed primary task: 21.000 seconds.
- The standard deviation to first failed primary task: 18.701 seconds.

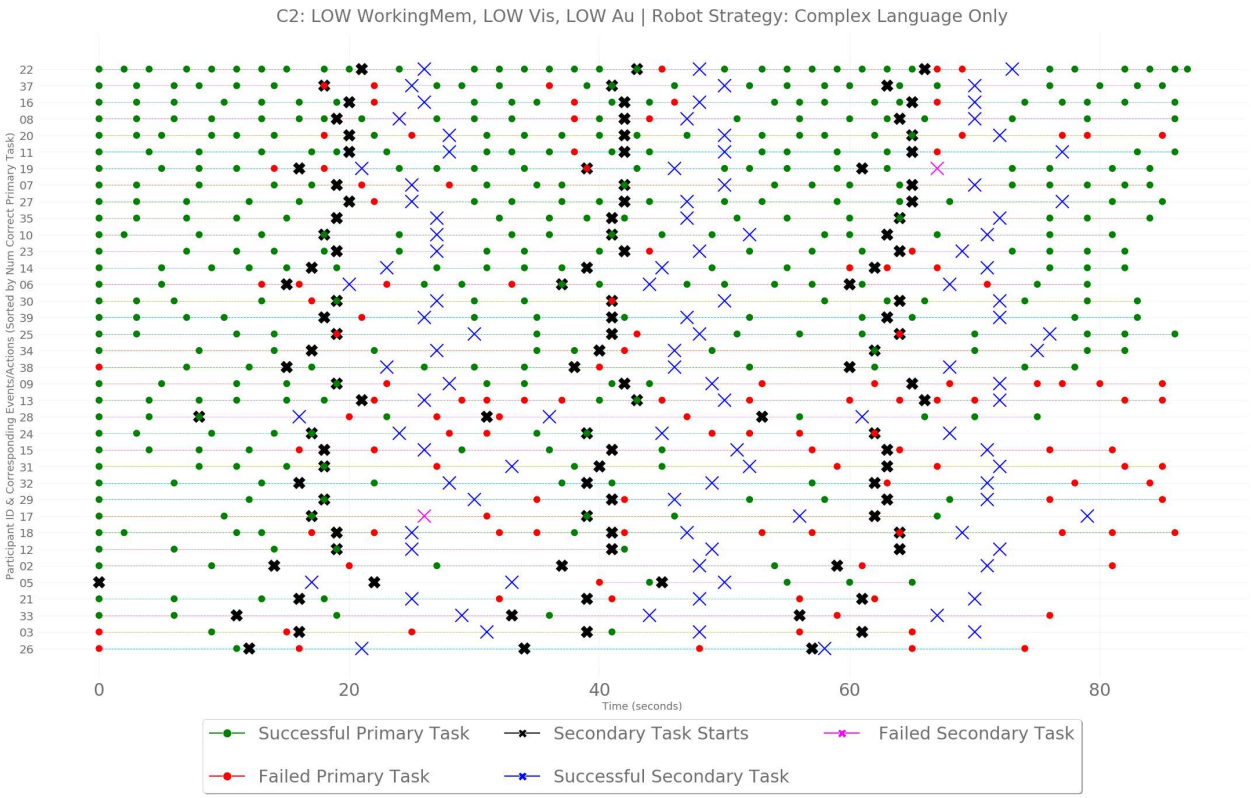

Figure 5.2 Visualization of participant performance in the *Complex Language Only / Low All*

In condition 2 where participants were under low overall workload and the robot used only complex language, the following facts were observed:

- 3 participants completed the primary tasks without any errors: [12, 10, 35].
- The average time to the first failed primary task: 25.606 seconds.
- The median time to the first failed primary task: 22.000 seconds.
- The standard deviation to first failed primary task: 16.413 seconds.

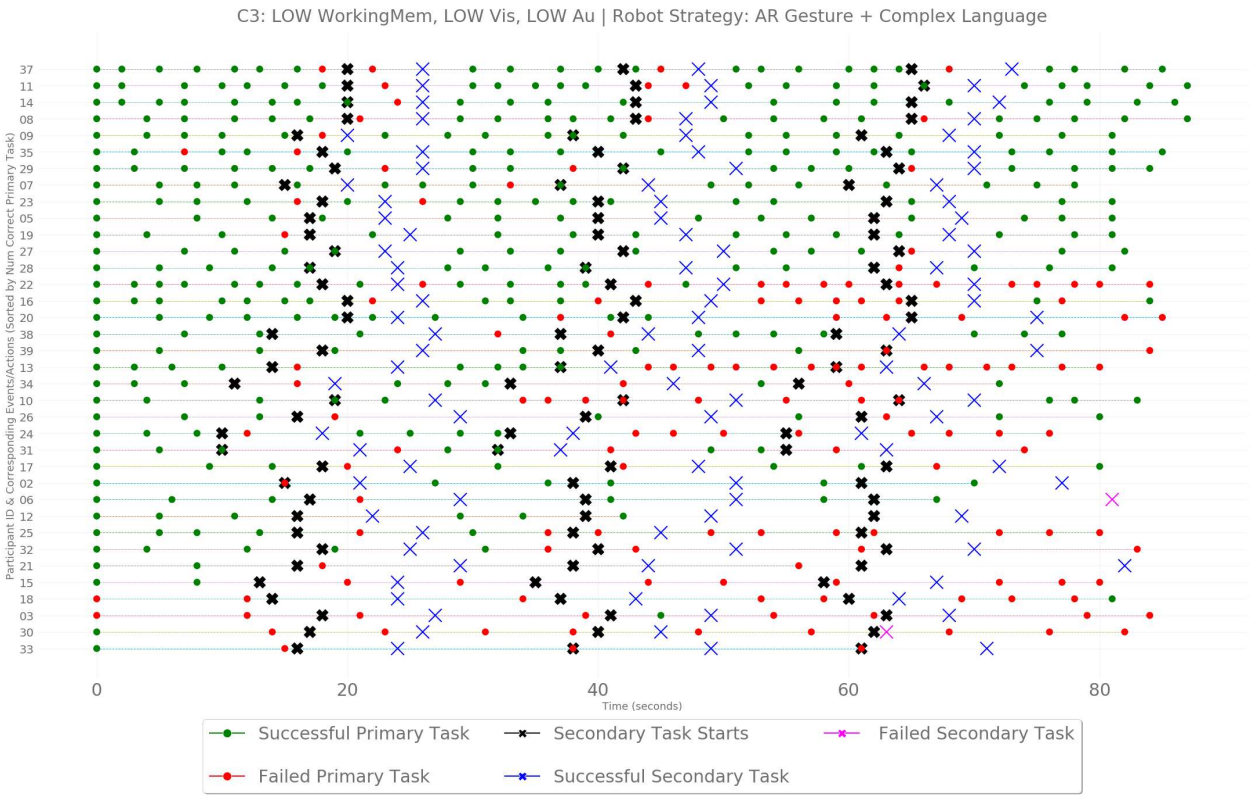

Figure 5.3 Visualization of participant performance in the *AR + Complex Language / Low All*

In condition 3 where participants were under low overall workload and the robot used AR + Complex language, the following facts were observed:

- 2 participants completed the primary tasks without any errors: [12, 5].
- The average time to the first failed primary task: 23.765 seconds.
- The median time to the first failed primary task: 20.500 seconds.
- The standard deviation to first failed primary task: 15.256 seconds.

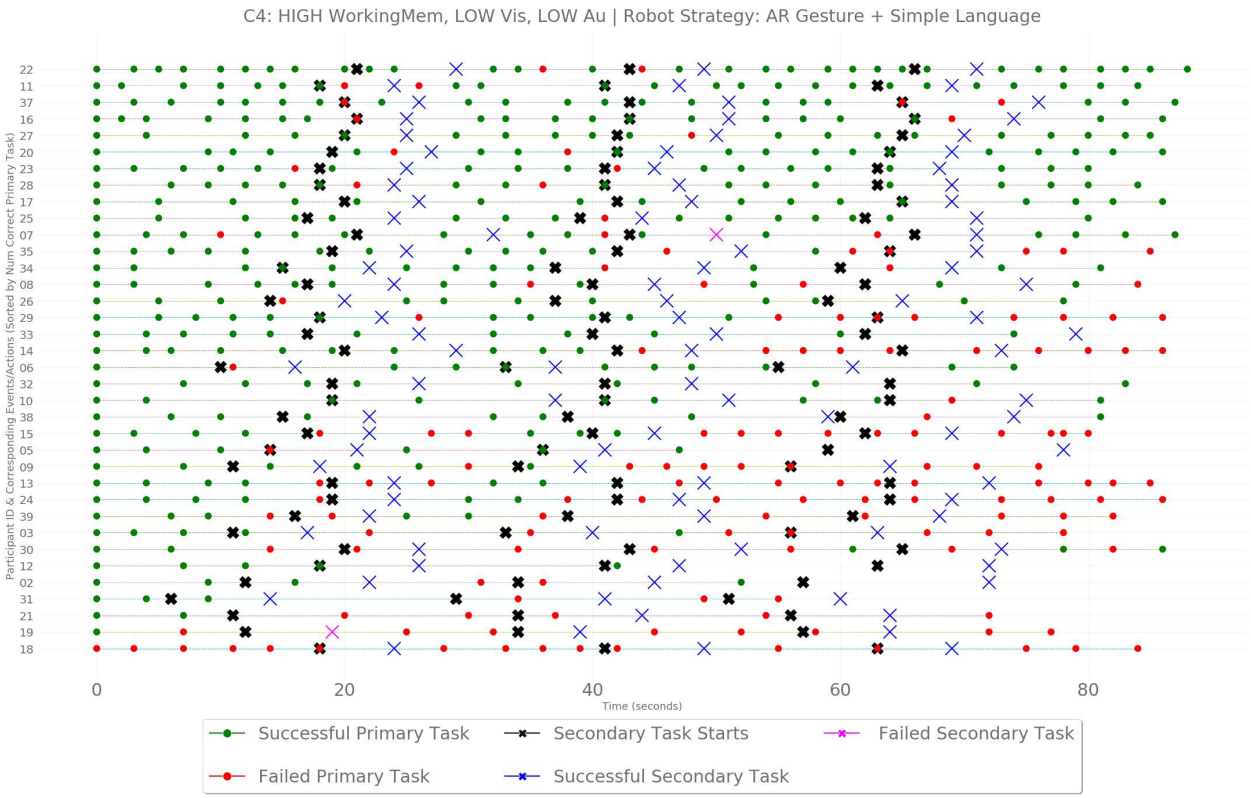

Figure 5.4 Visualization of participant performance in the *AR + Simple Language / High Working Memory Condition*

In condition 4 where participants were under high working memory load and the robot used AR + Simple language, the following facts were observed:

- 4 participants completed the primary tasks without any errors: [17, 12, 33, 32].
- The average time to the first failed primary task: 26.594 seconds.
- The median time to the first failed primary task: 21.000 seconds.
- The standard deviation to first failed primary task: 16.068 seconds.

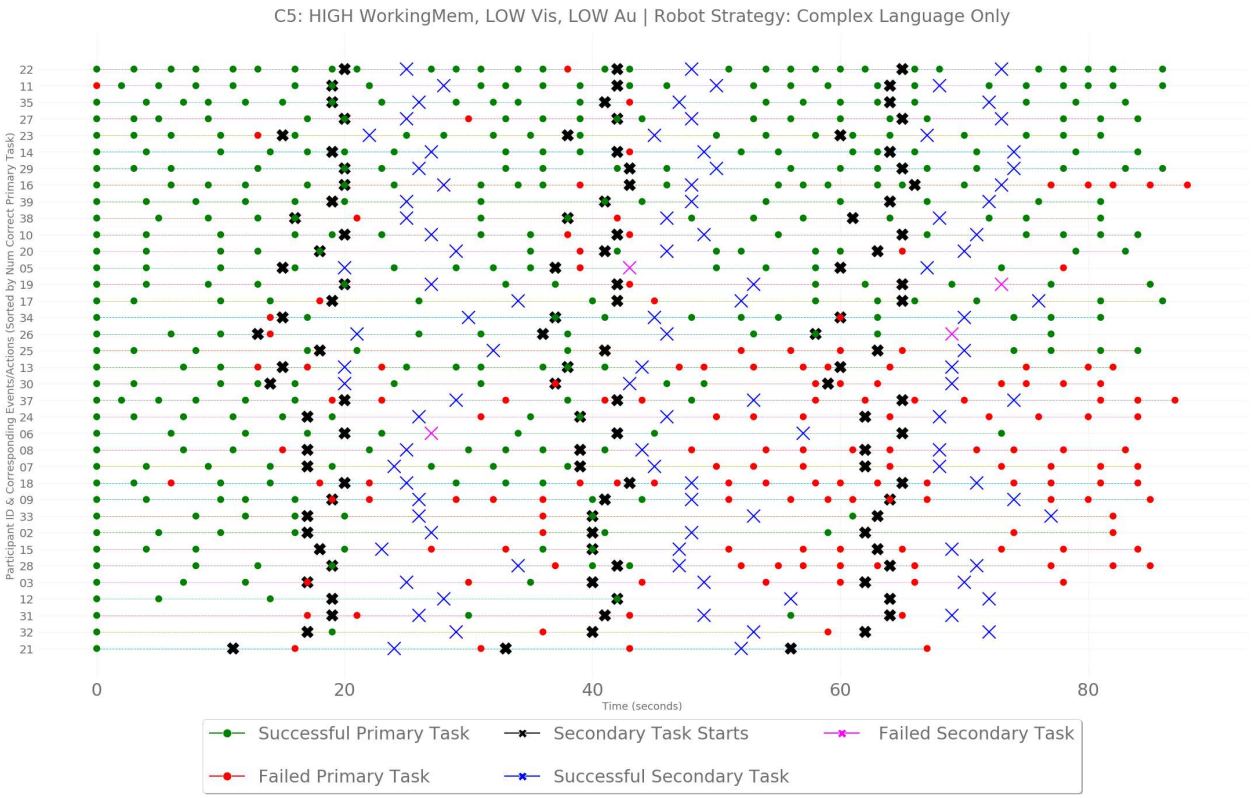

Figure 5.5 Visualization of participant performance in the *Complex Language Only / High Working Memory Condition*

In condition 5 where participants were under high working memory load and the robot used complex language only, the following facts were observed:

- 4 participants completed the primary tasks without any errors: [12, 29, 6, 39].
- The average time to the first failed primary task: 28.000 seconds.
- The median time to the first failed primary task: 30.500 seconds.
- The standard deviation to first failed primary task: 13.491 seconds.

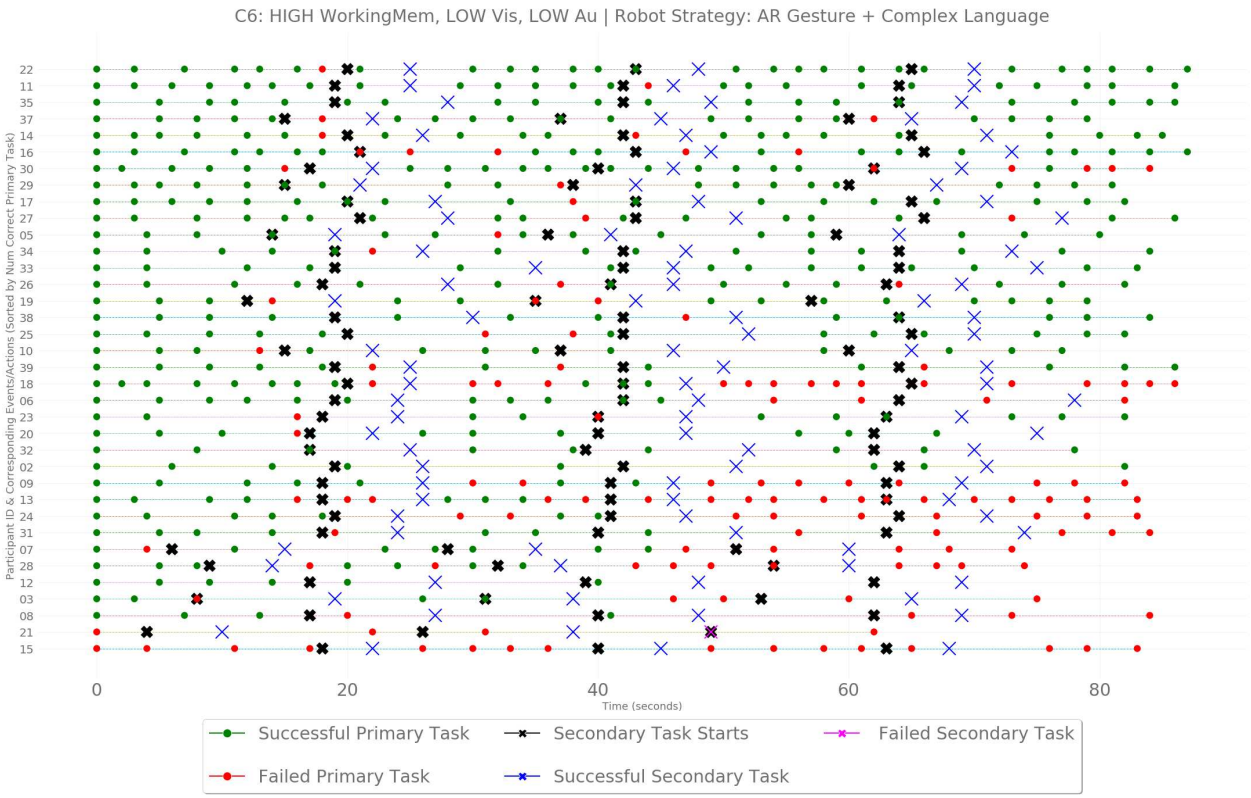

Figure 5.6 Visualization of participant performance in the *AR + Complex Language / High Working Memory Condition*

In condition 6 where participants were under high working memory load and the robot used complex language only, the following facts were observed:

- 5 participants completed the primary tasks without any errors: [12, 33, 32, 35, 2].
- The average time to the first failed primary task: 23.129 seconds.
- The median time to the first failed primary task: 20.000 seconds.
- The standard deviation to first failed primary task: 13.241 seconds.

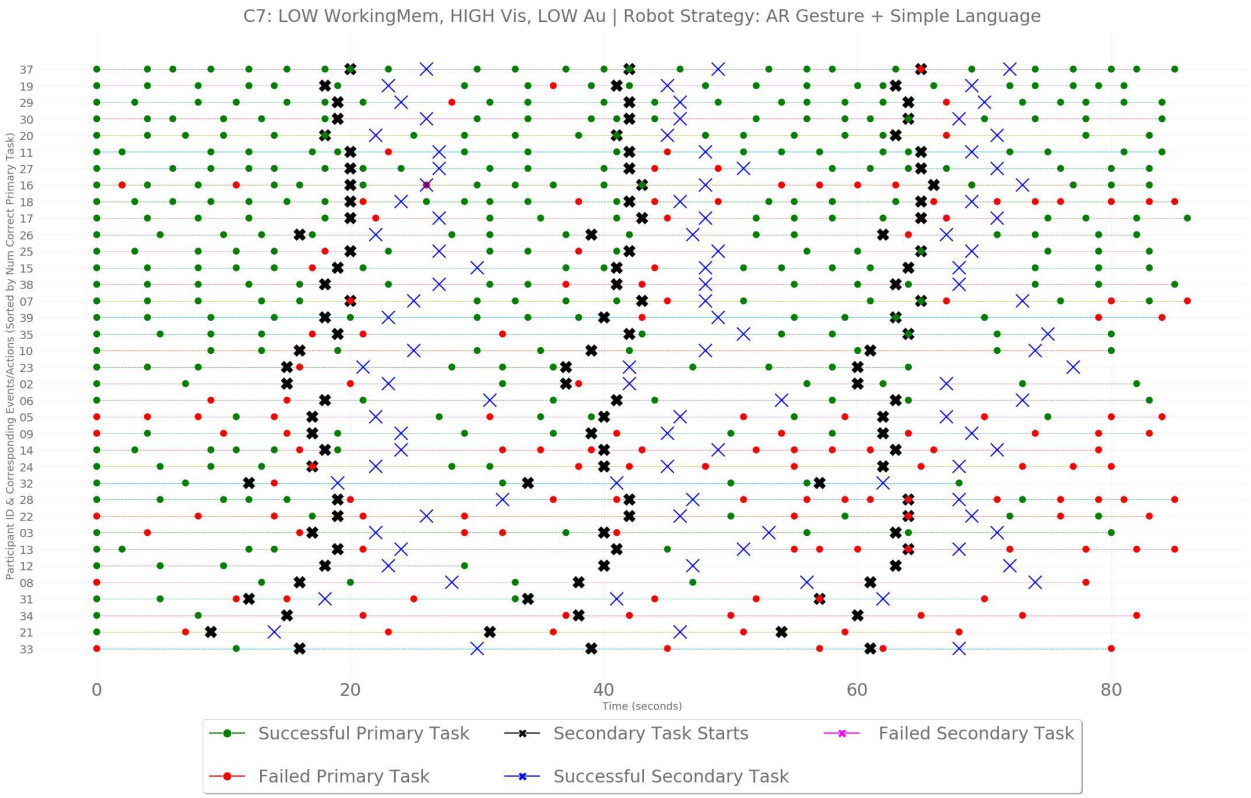

Figure 5.7 Visualization of participant performance in the *AR + Simple Language / High Visual Load Condition*

In condition 7 where participants were under high visual load and the robot used AR + simple language, the following facts were observed:

- 3 participants completed the primary tasks without any errors: [12, 10, 30].
- The average time to the first failed primary task: 21.212 seconds.
- The median time to the first failed primary task: 18.000 seconds.
- The standard deviation to first failed primary task: 18.457 seconds.

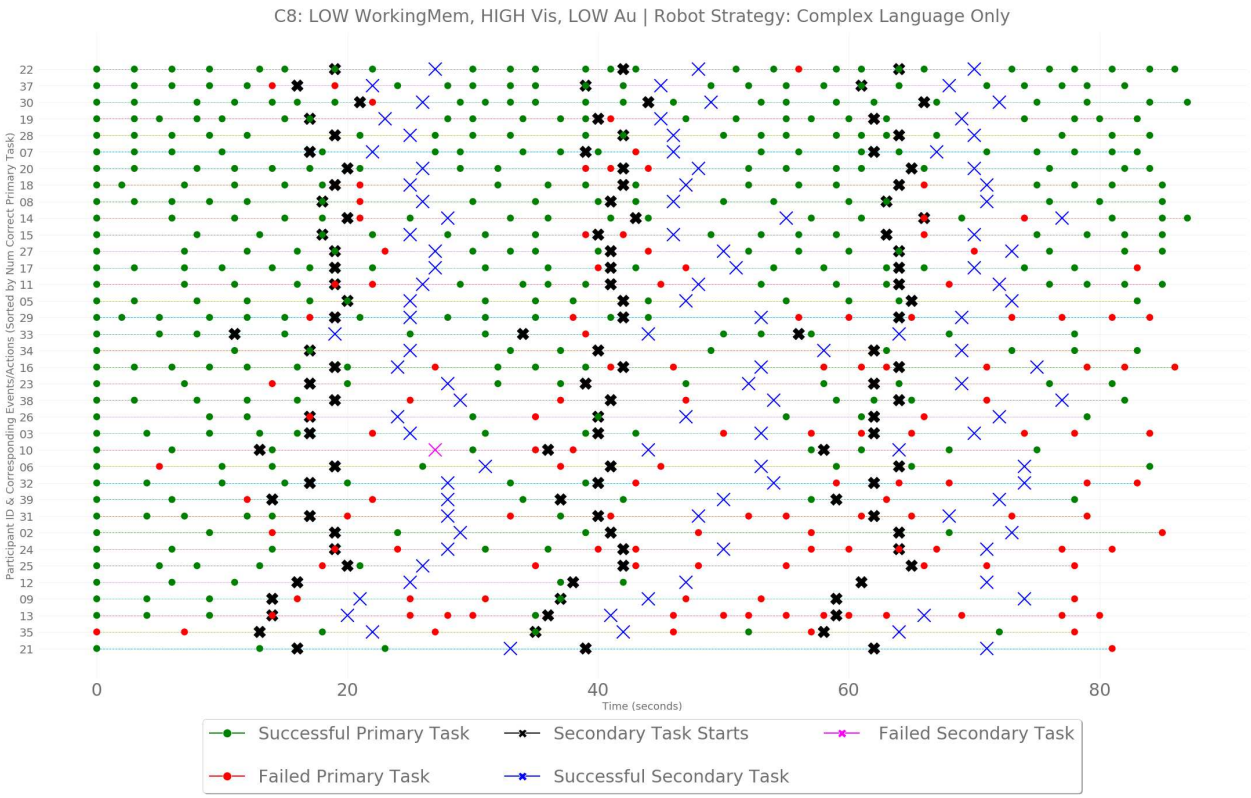

Figure 5.8 Visualization of participant performance in the *Complex Language Only / High Visual Load Condition*

In condition 8 where participants were under high visual load and the robot used complex language only, the following facts were observed:

- 4 participants completed the primary tasks without any errors: [28, 12, 34, 5].
- The average time to the first failed primary task: 26.156 seconds.
- The median time to the first failed primary task: 21.000 seconds.
- The standard deviation to first failed primary task: 16.043 seconds.

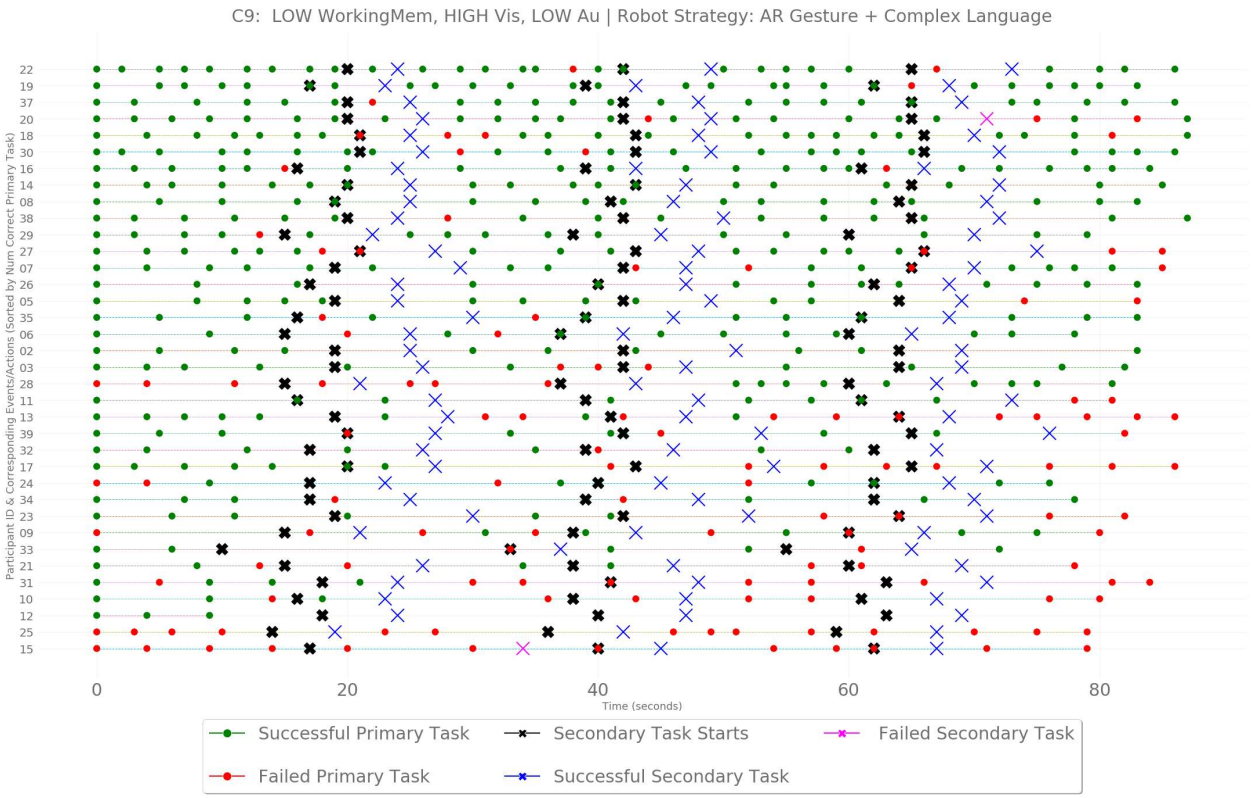

Figure 5.9 Visualization of participant performance in the *AR + Complex Language / High Visual Load Condition*

In condition 9 where participants were under high visual load and the robot used AR + complex language, the following facts were observed:

- 5 participants completed the primary tasks without any errors: [8, 12, 14, 2, 26].
- The average time to the first failed primary task: 27.000 seconds.
- The median time to the first failed primary task: 21.000 seconds.
- The standard deviation to first failed primary task: 21.309 seconds.

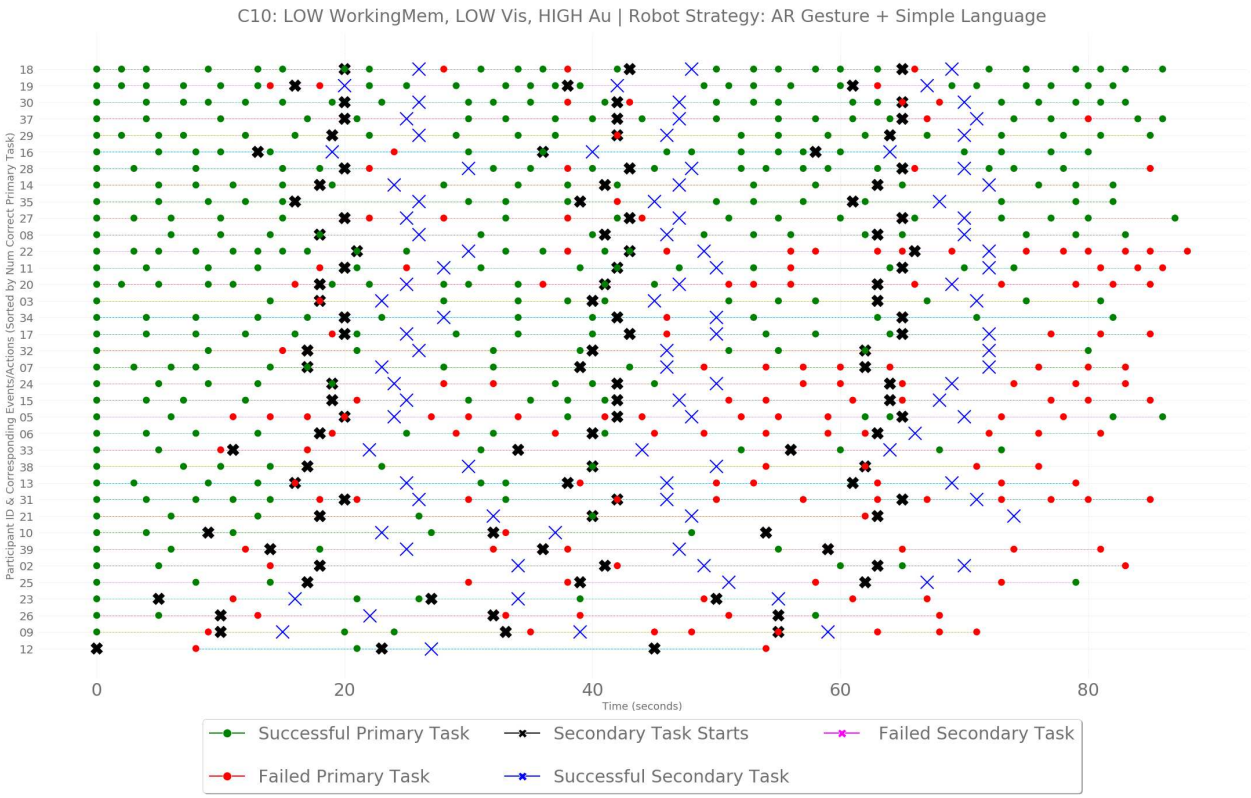

Figure 5.10 Visualization of participant performance in the *AR + Simple Language / High Auditory Load Condition*

In condition 10 where participants were under high auditory load and the robot used AR + simple language, the following facts were observed:

- 2 participants completed the primary tasks without any errors: [8, 14].
- The average time to the first failed primary task: 25.853 seconds.
- The median time to the first failed primary task: 20.000 seconds.
- The standard deviation to first failed primary task: 16.104 seconds.

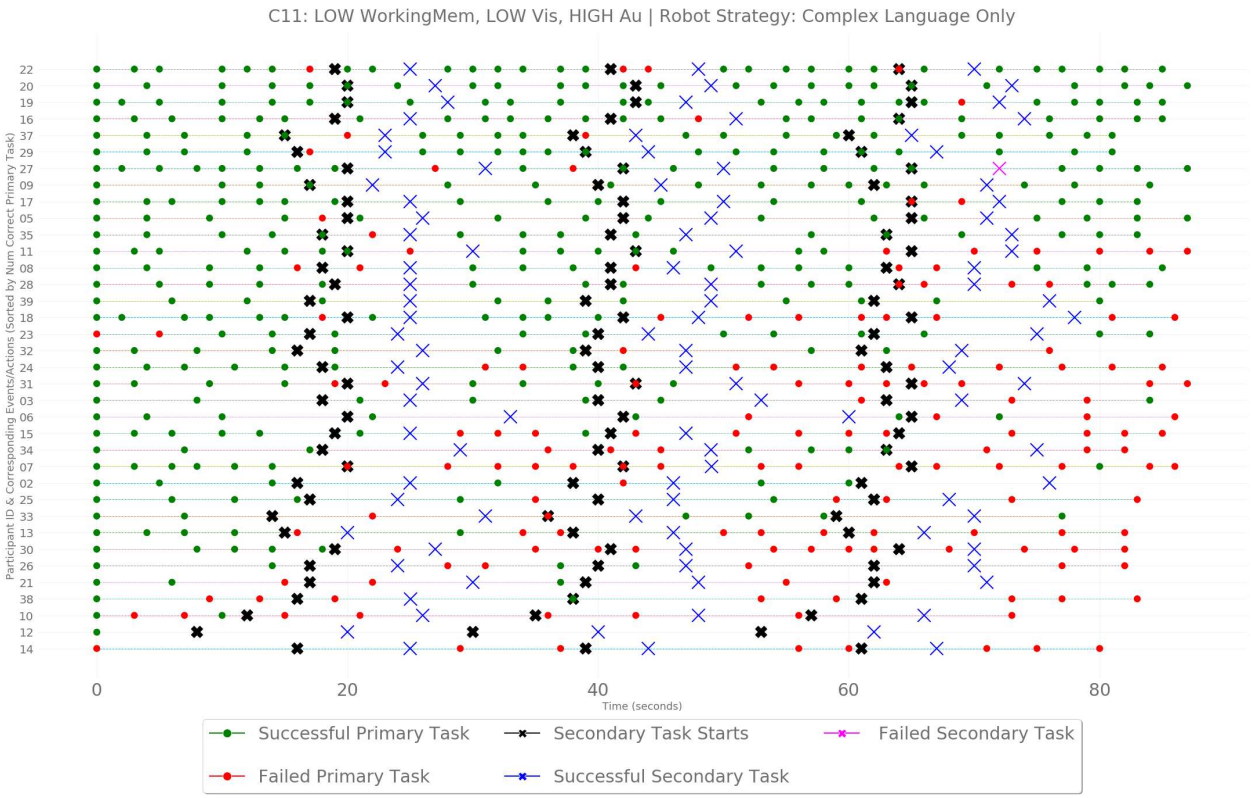

Figure 5.11 Visualization of participant performance in the *Complex Language Only / High Auditory Load Condition*

In condition 11 where participants were under high auditory load and the robot used complex language only, the following facts were observed:

- 4 participants completed the primary tasks without any errors: [12, 9, 39, 20].
- The average time to the first failed primary task: 28.438 seconds.
- The median time to the first failed primary task: 23.000 seconds.
- The standard deviation to first failed primary task: 18.561 seconds.

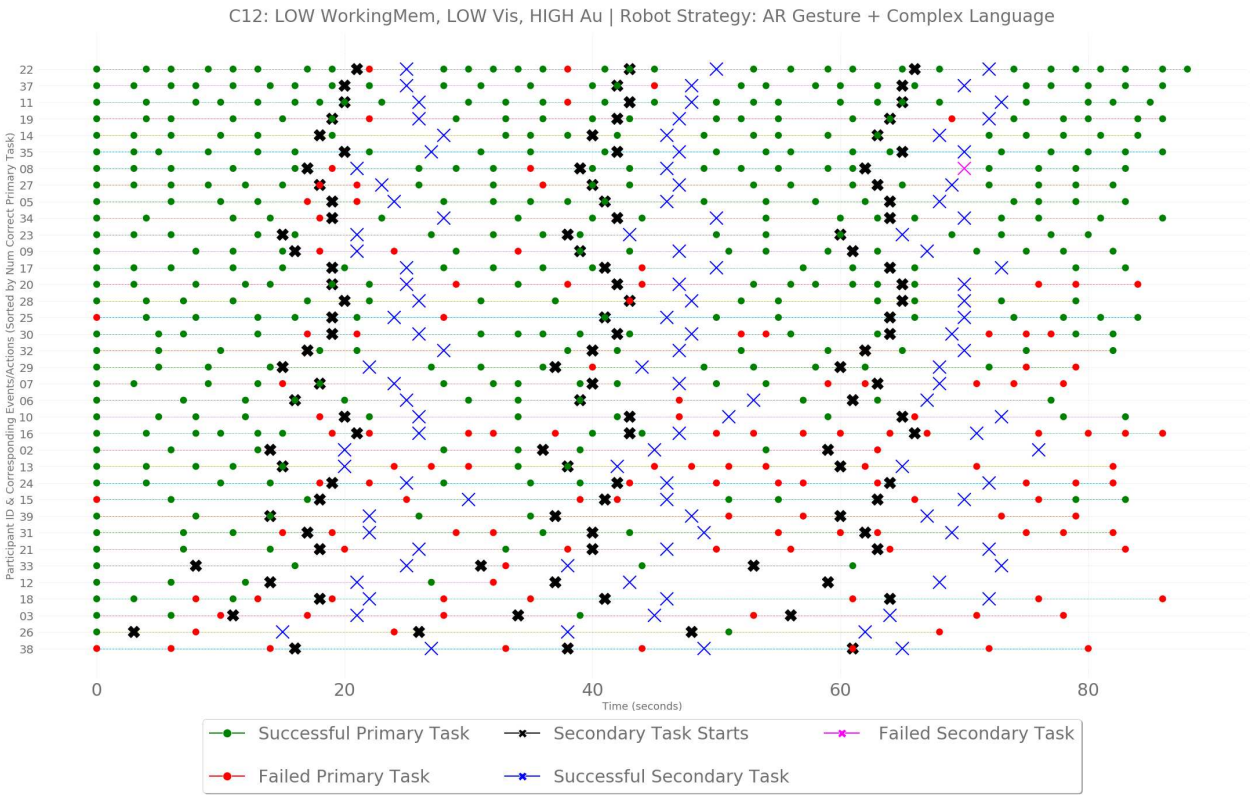

Figure 5.12 Visualization of participant performance in the *AR + Complex Language / High Auditory Load Condition*

In condition 12 where participants were under high auditory load and the robot used AR + complex language, the following facts were observed:

- 4 participants completed the primary tasks without any errors: [23, 32, 35, 14].
- The average time to the first failed primary task: 24.094 seconds.
- The median time to the first failed primary task: 19.000 seconds.
- The standard deviation to first failed primary task: 15.609 seconds.

The ultimate goal of our research is to enable adaptive mixed reality communication for human-robot interaction. We presented the first experimental steps towards achieving this goal. Our results provide critical insights for the future design of our proposed adaptive system. A limitation with this current study was that participants had to wait until the round ended in order to use the NASA TLX survey to self-report their workload. This led to insignificant statistical results. In future work, we plan to complete our integration of the fNIRS neurophysiological sensor with the current mixed reality robotic architecture, in order to accurately measure changes in mental workload *within* experimental conditions, as well as in task contexts that do not have tightly controlled levels of workload. We further plan to integrate all three components together with the Distributed Integrated Affect Reflection and Cognition (DIARC) architecture to leverage its rich natural language understanding and generation capabilities [84, 85].

Moreover, the relationship between workload and task performance is complex: it is not always the case that as workload increases performance will decrease [39]. We hope to conduct deeper analyses of these trajectories, specifically examining factors such as differences in task completion speeds and rates of catastrophic primary task shedding across conditions. As we analyze these metrics and explore other types of visualizations and their properties, we hope to better understand what kind of tasks and visualizations can be combined to yield the least drop in task efficiency.

Finally, in future work, we also plan to consider how robots can tailor gestural cues to be easily discriminable from both background visual stimuli and other task targets without placing the human teammate at risk of inattentional blindness. Instead of building a passive system, we plan to build an active robotic system that can be sensitive to both the current context and the predicted effect of potential choices of communication modality. By designing such an adaptive system for communication modalities selection using probabilistic modeling techniques, we strive to give robots the ability to not only capture the human selective attention but also to tap into the human's unengaged cognitive resources.

CHAPTER 6

SOFTWARE ARCHITECTURE

In this chapter, we present a detailed architecture description of our mixed reality robotic communication system. We built a mixed reality application using the Unity game engine. Within the Unity application, there are several sub-components that talk to each other using Unity events and delegates. We developed a robust communication pipeline, as shown in Figure 6.3, that enabled duplex data transmission between the mixed reality headset Microsoft HoloLens and the Pepper robot from SoftBank Robotics. Setup involved starting the WebSocket server on a centralized computer and connecting with the WebSocket client on the HoloLens and robot sides. After all clients connect to the same WebSocket server, they are capable of publishing and subscribing to real-time messages to each other via bidirectional connection.

6.1 Microsoft HoloLens 1

Figure 6.1 The Microsoft HoloLens version 1 [86]

Released in 2016, the Microsoft HoloLens (see Figure 6.1) was the first commercial AR HMD to enter the market. Unlike other HMDs, the HoloLens does not require an external tethered device. It features an Intel Atom x5-Z8100 1.04 GHz with four logical processors, a Holographic Processing Unit (HPU), 2 GB RAM, 64 GB flash, four environment-processing

camera, one RGB camera, one depth camera, and 2-3 hours of battery life. It projects light through the holographic lenses using two high definition light engines to generate spatial 3D content. The HoloLens 1 comes in the box with gaze tracking, gesture input, spatial sound, and the Cortana virtual assistant [87].

6.2 Unity

Figure 6.2 A Unity Scene of our application

A cross-platform game engine, Unity can be used to quickly prototype and create 2D and 3D games and simulations. A HoloLens app must be built using the Universal Windows Platform (UWP). After designing the application in Unity, the rendering platform must be switched to UWP so that the app can be deployed on the actual headset. To speed up the development process, Microsoft provides the HoloToolkit, a repository of samples, scripts, and components for Unity [88]. The first important component that HoloToolkit supports is Input which manages how users can interact with mixed reality objects using simple hand gestures, eye gaze, and voice. In our experimental application [88], we leveraged the Input system so that users can select virtual menu options using an air tap gesture and move

the virtual blocks into virtual bins using the pinch gesture. Second, the Spatial Mapping is supported so that the Hololens can keep track of the 3D mesh of the surrounding space [88]. We anchored the virtual bins into the real world using Spatial Mapping. The third component is Spatial Sound, which gives the illusion that the sound coming from a virtual object is positioned in 3D space [88]. To improve the user experience, we added auditory effects to all virtual objects. With the support of virtual sounds, the participants can easily localize the source, know when they place something into a bin, experience the auditory load modulation designed in the primary task, and feel like they interact with real objects.

Because the early purpose of Unity was game development, a lot of crucial elements in Unity are game oriented [86]. For example, the user of the application is called the player. The environment of the game is called *scene*. A simple game or application needs only one scene. However, a more complex system such as our experimental app requires multiple scenes which logically divide up several parts of the game/application. Our app consists of four main scenes: the start menu, the main experiment, the break scene, and the tutorial scene.

Everything that lives in a scene is called a GameObject [86]. A GameObject can be a 2D UI canvas, a 3D block, or an empty object without any physical appearance. To make these GameObjects alive and interactive, we controlled their behaviors using the scripts written in the programming language C# (pronounced *C Sharp*) [86]. For example, in order to allow users to pick up a virtual block, hear an auditory cue, or drop a block into a virtual bin, we wrote corresponding C# scripts to handle these behaviors. When developing our app, we wrote most custom scripts based on the samples provided by the HoloToolkit, which tapped into low-level Unity components such as raycasting to detect which object is being gazed at and events system to allow different GameObjects to talk to each other.

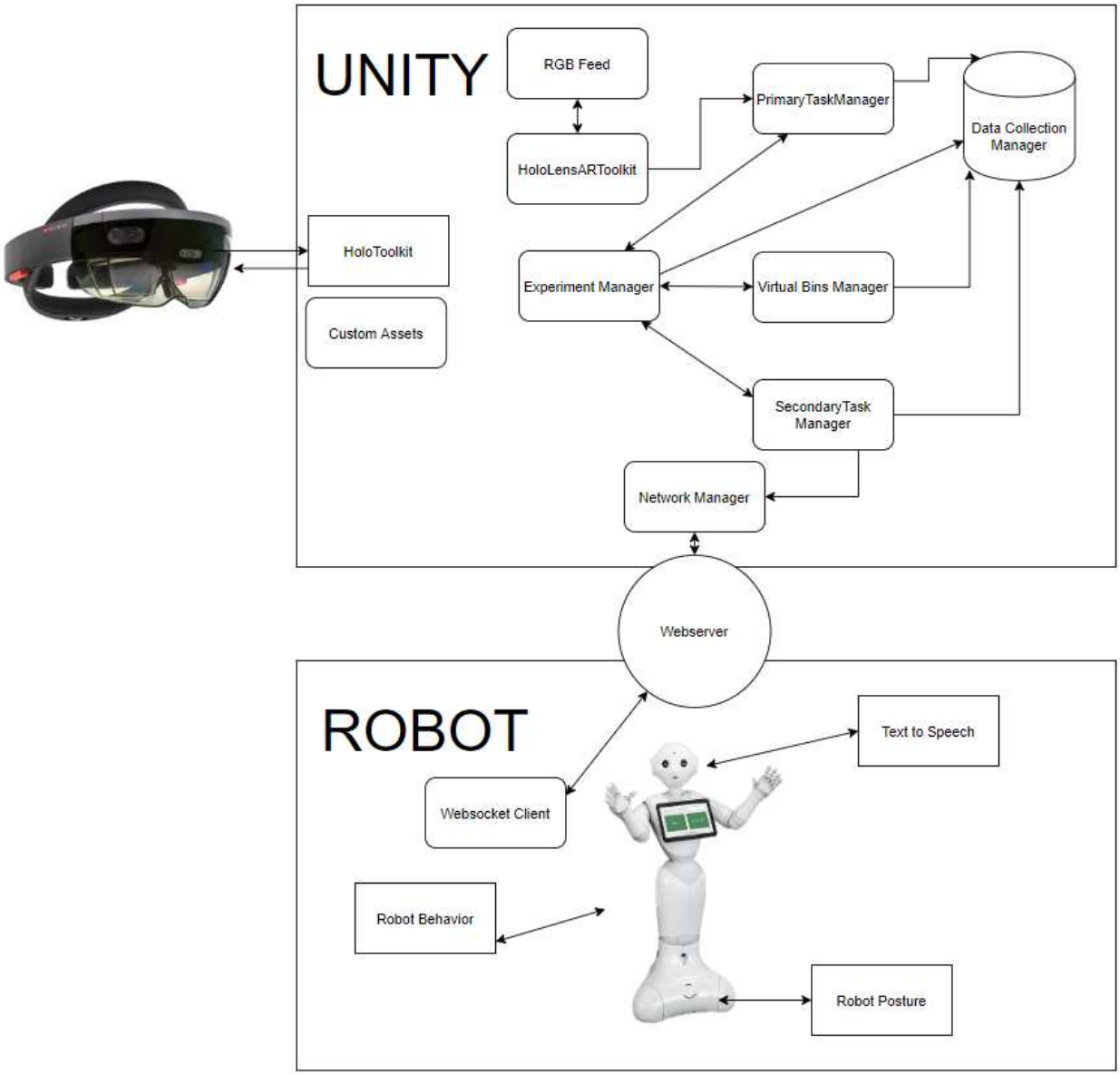

Figure 6.3 Overview of our robot mixed reality system architecture

As shown in Figure 6.3, our main experiment scene consists of eight big components, also known as managers, to manage eight different important tasks happening concurrently.

6.2.1 Primary Task Manager

The Primary Task Manager allows the experimenter to configure several settings within Unity's Inspector panel, including:

- *highVisualLoad (boolean)*: determines whether the visibly virtual blocks should share a color or a shape.
- *blockUpdateDuration (float)*: determines how often the panel of blocks should be updated.

The backend dynamically maintains two *List* data structures: *blockInventory* and *blocksOnScreen*. Initially, *blockInventory* holds all the possible combinations of block's shapes and block's colors. If the experimenter specifies three colors (red, green, yellow) and their shapes (cube, sphere, cylinder), *blockInventory* will contain a Cartesian product of nine blocks. Then, one of these blocks is randomly selected to be the first primary target block. Eight other non-target blocks are also randomly selected from the inventory. Duplicates are allowed as long as there is only one instance of the target block displayed to the user. All nine blocks are then stored in the *blockOnScreen* list. After every n second (*blockUpdateDuration*), a non-target block is randomly removed from *blockOnScreen* and replaced by a different block randomly selected from *blockInventory*. The primary target block remains persistent throughout the entire 90 seconds round.

6.2.2 Secondary Task Manager

The Secondary Task Manager works closely with the Primary Task Manager to coordinate the selection of the secondary target block. The experimenter manually sets up the *durationToTriggerTask (float)* setting in Unity's Inspector panel in order to indicate how often the secondary task should be generated. As shown in Figure 6.4, the robot in our experiment asks the participants every 22.5 seconds within a 90 seconds round. The backend randomly selects among the *blockOnScreen* a secondary target block that is different from

the primary target block. In addition, it randomly selects a secondary target bin among the visibly accessible bins. Depending the round condition the participant is in, it then determines whether to show the AR annotation in the participant's field of view and sends either a simple or complex sentence to the robot via a WebSocket network pipeline. When the robot receives the message over Websocket, it then asks the users to place the secondary target block into a new bin. When the robot requests the participant to pick up a new secondary target block and place it into a new target bin, the Secondary Task Manager informs the Primary Task Manager to skip replacing the secondary task target block with a new random block. While the robot is speaking, the Secondary Task Manager also asks the Sound Manager to pause reading the syllables aloud so that the participant hears only one stream of audio.

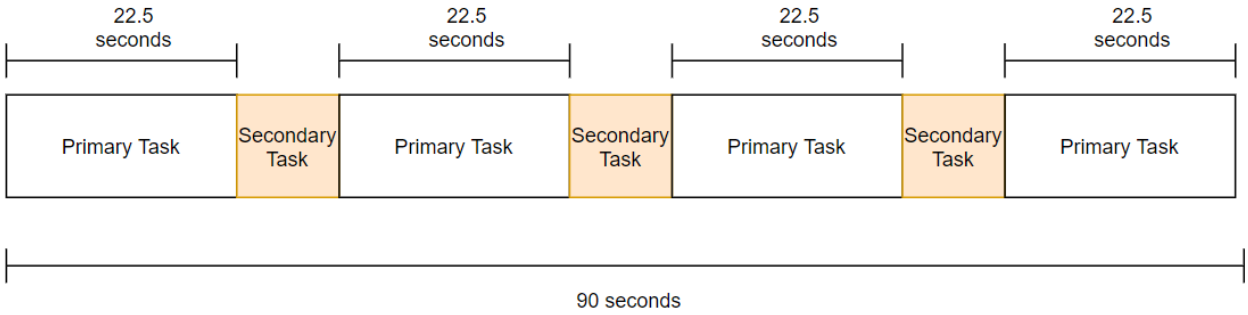

Figure 6.4 Primary and Secondary Tasks in One Game Round

6.2.3 Experiment Manager

The Experiment Manager oversees all other submanagers and acts as a liaison. Figure 6.2 shows the Unity's Inspector view of the Experiment Manager. With direct references to the Primary Task Manager, the Secondary Task Manager, the Bin Manager, the Network Manager, and the Sound Manager, the Experiment Manager ensures the game can only start when all resources are loaded. The experimenter can set up global settings such as Debug Mode, Timer Visibility, Data Logging, Game Quitting/Pausing, and Condition Setup. Since

the participant's primary task is made up of a set of 12 within-subject rounds/conditions, with their order counterbalanced using a Latin square design, the Experiment Manager passes on the conditional settings from the main menu to all other managers.

6.2.4 Bin Manager

The Bin Manager manages all logic related to virtual bins. Experimenters can toggle the *High Working Memory boolean* to change the number of bins displayed to the participants. It holds a *List* data structure of custom BinObjects. Each binObject holds metadata such as *binValid (boolean)* to denote target bins, *isGreyedOut (boolean)* to denote bin accessibility, and *secondaryTaskBin (boolean)* to denote a secondary target bin. The Bin Manager serves three main functions:

1. Randomly flipping the validity of the bins

   As mentioned in Chapter 3.5, the HoloLens plays the sound of syllables at every l second(s) to manipulate the auditory load. Whenever the target syllable $\{consonant \in \{b, t, k\}, vowels \in \{ah, eh, oh\}\}$ is played, all of the acceptable bins flip to the unacceptable status and vice versa. When the Sound Manager plays a target syllable, it will call the Bin Manager to turn all target bins into valid and invalid bins into the new targets.

2. Randomly selecting bins for the secondary task

   When the Secondary Task Manager needs to generate a new request, it will call the Bin Manager to randomly determine a secondary target bin which is 1) accessible (not greyed out), and 2) not the same as the previous secondary target bin.

3. Randomly greying out bins

   For every n second(s) as set up by the experimenter, a bin is randomly greyed out to make the dual-tasks more challenging. This function is leveraged by both the Primary Task Manager and the Secondary Manager. Whenever it is called, it will select a bin

that is 1) not the currently greyed out bin ($isGreyedOut = False$), and 2) not reserved as the secondary target bin ($secondaryTaskBin == False$).

6.2.5 Network Manager

WebSocket connection acts like a byte-stream-bidirectional TCP link between client and server established in an HTTP Send-Receive cycle [89]. Windows UWP provides low-level socket abstraction called *Windows.Networking.Sockets*.

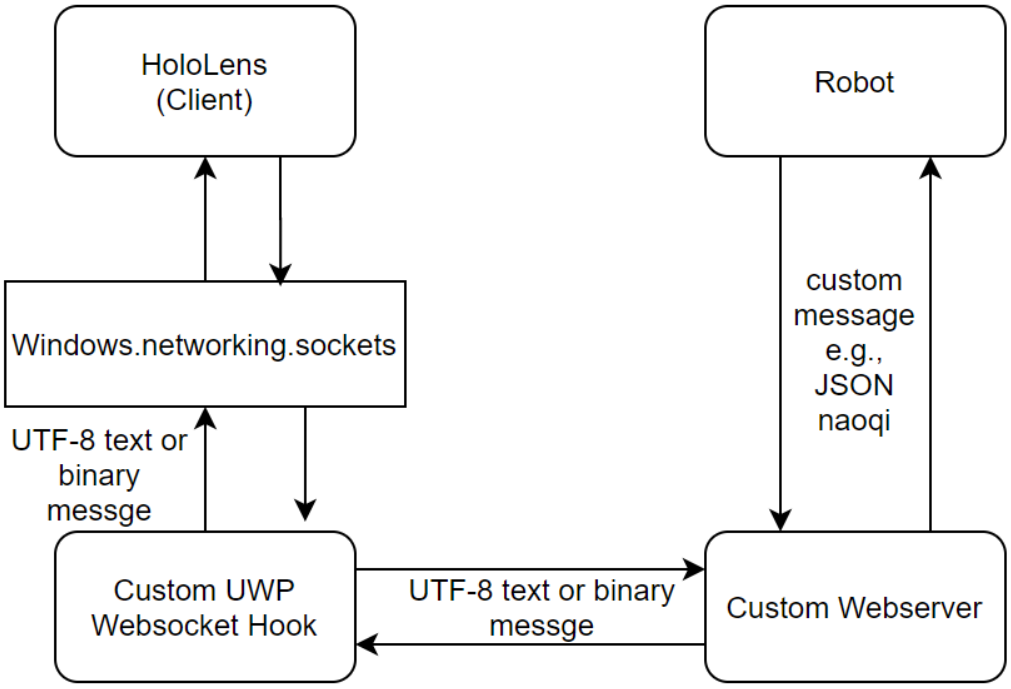

Figure 6.5 Websocket Communication

Building a custom class on top of it, we implemented custom WebSocket interfaces, including *OnStart* (what the app should first do when a connection is established), *OnMessage* (how the app should process message sent by the other node), *OnApplicationQuit* (how the app should properly save all data before closing the socket on the device), *SendMsgToSubscriber* (how the receivers should receive the message). This custom class is the backbone of the Network Manager, and is also modularized to allow other submanagers to easily reference

it. The Network Manager allows our app to connect to any computer that uses the same WebSocket protocol and has the *ws:/xxx.xxx.xxx:8000/* format. Users can set up the new address they want the HoloLens to connect to within the start menu of the device as shown in Figure 6.6. Figure 6.5 shows our app's ability to send and communicate with a Webserver (this can be written in any programming languages). The WebServer then processes the messages and forwards the appropriate commands to the robot.

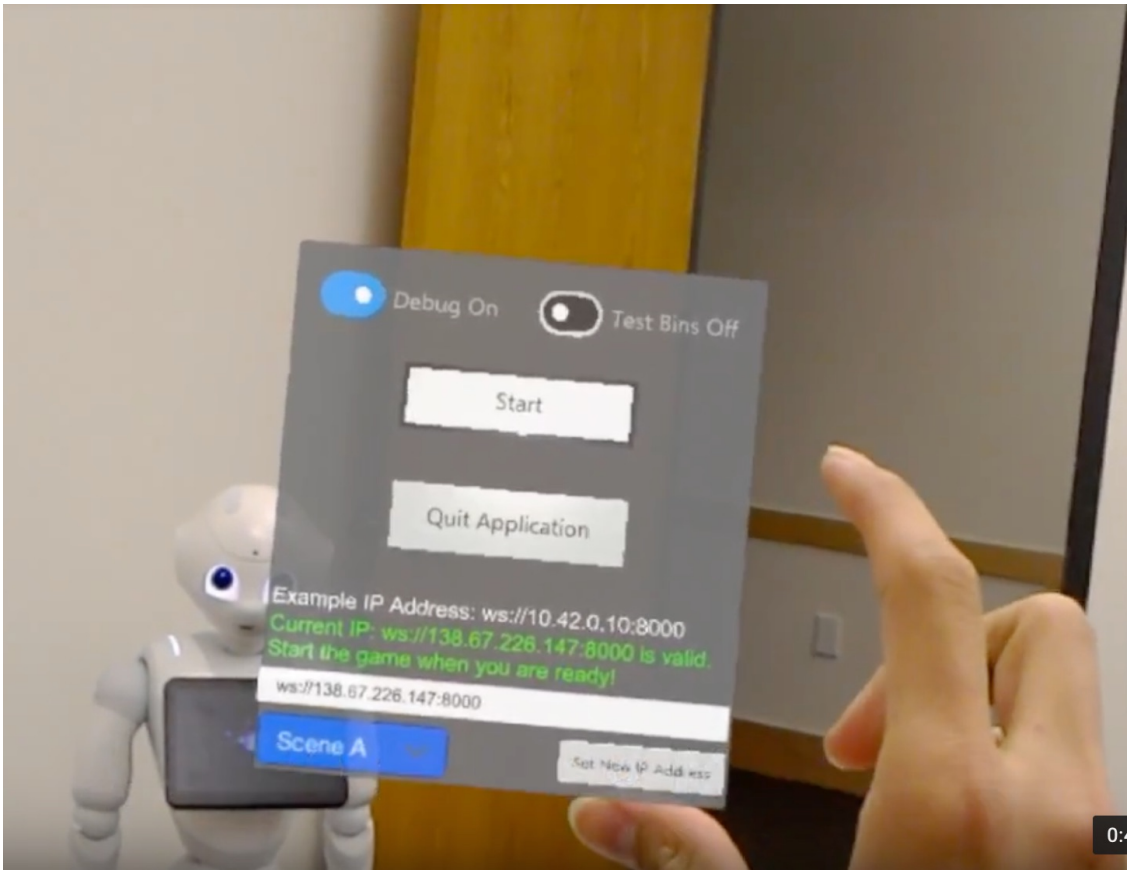

Figure 6.6 A setting scene to set up the WebSocket connection

6.2.6 Data Collection Manager

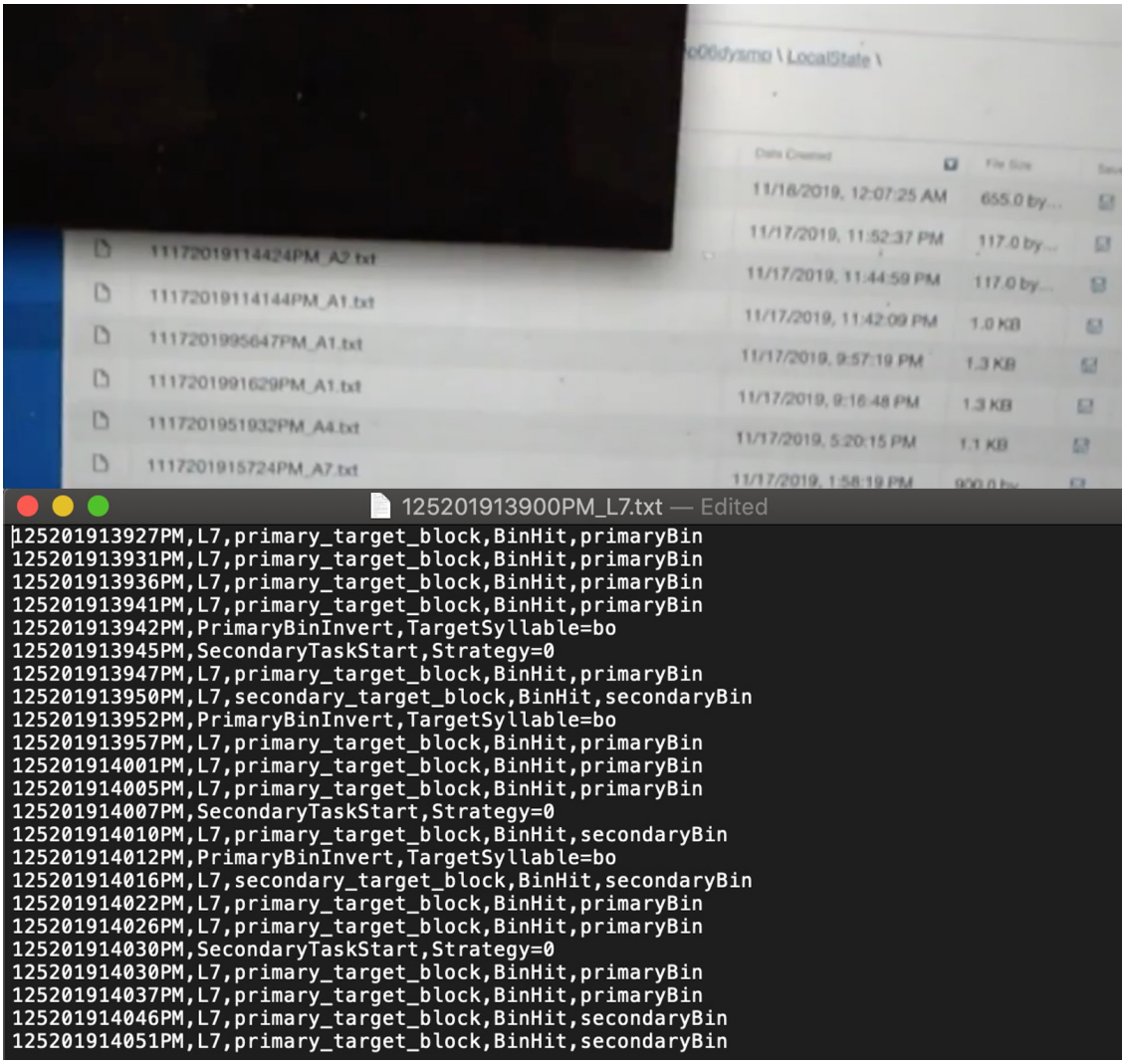

Figure 6.7 Data collected after an experiment

The Data Collection Manager manages the event logging during an experiment. When the app initializes a new round, the Data Collection Manager produces a text file on the HoloLens with the format *datetime_LatinSquareCondition.txt* (see Figure 6.7).

As the participant plays our game, the Data Collection Manager logs:

1. Primary Task Event
2. Primary Task's Bins switch-up
3. Secondary Task Event
4. Latin Square Condition

Since a participant has to go through all twelve within-subject rounds, twelve log files are recorded per participant.

6.2.7 HololensARToolkit

Azimi et al. [90] interfaced the open-sourced, 10 year-old, native ARToolkit library to the Universal Windows Platform, enabling fast marker tracking in HoloLens. They called it *HoloLensARToolKit v0.2*. The library supports rendering at 45-60 frames per second, video capture at 30 frames per second, and tracking at 25-30 frames per second performance [90]. After a short pilot of the experiment, we found that enabling continuous tracking significantly reduced the application performance and frame rate to 20 fps. Since our study tried to measure the human's task time as well as their perception of the task, we looked into various ways to improve the user experience. We then modified the HoloLensARToolKit source code to stop the tracking after successfully recognizing the correct number of markers. After finding the real-world coordinates of four markers (for low working memory load) and six markers (for high working memory load), the app leverages the HoloToolkit's component WorldAnchorStore to save the markers. Spatial Anchors on the first run. The players can only continue playing the game if they have looked around the environment and saved all anchors. Then in the following rounds, the app simply reloads the markers' positions and re-renders the virtual bins on top of those markers without relying on the HoloLensARToolKit. As shown in Figure 6.8, six bins were recognized. The Bin Manager was also informed to

randomly select the primary task' target bins and randomly grey out a bin. This approach allowed us to consistently hit 60 fps during the twelve rounds of the experiment.

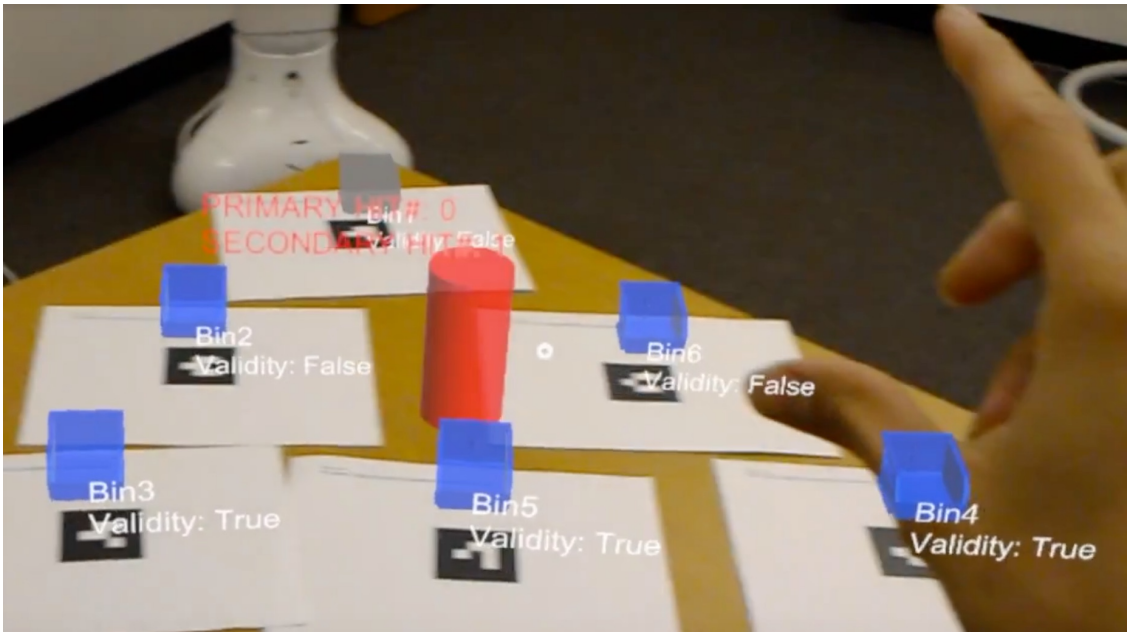

Figure 6.8 HoloARToolkit

6.2.8 WebServer

Our WebSocket server is a TCP application which listens to any server port. For fast prototyping we first implemented the server in Python. We later implemented a WebSocket server in Java using the Java Web Sockets open source JSR-356 API and Glassfish to improve compatibility with our lab's Distributed Integrated Affect Reflection and Cognition (DIARC) architecture. As the project progresses, we will leverage DIARC's affect processing and deep natural language processing features [84].

6.2.9 Robot Integration

The WebServer was built as a multithreaded application. As shown in Figure 6.9, there are three key components to make robot integration successful:

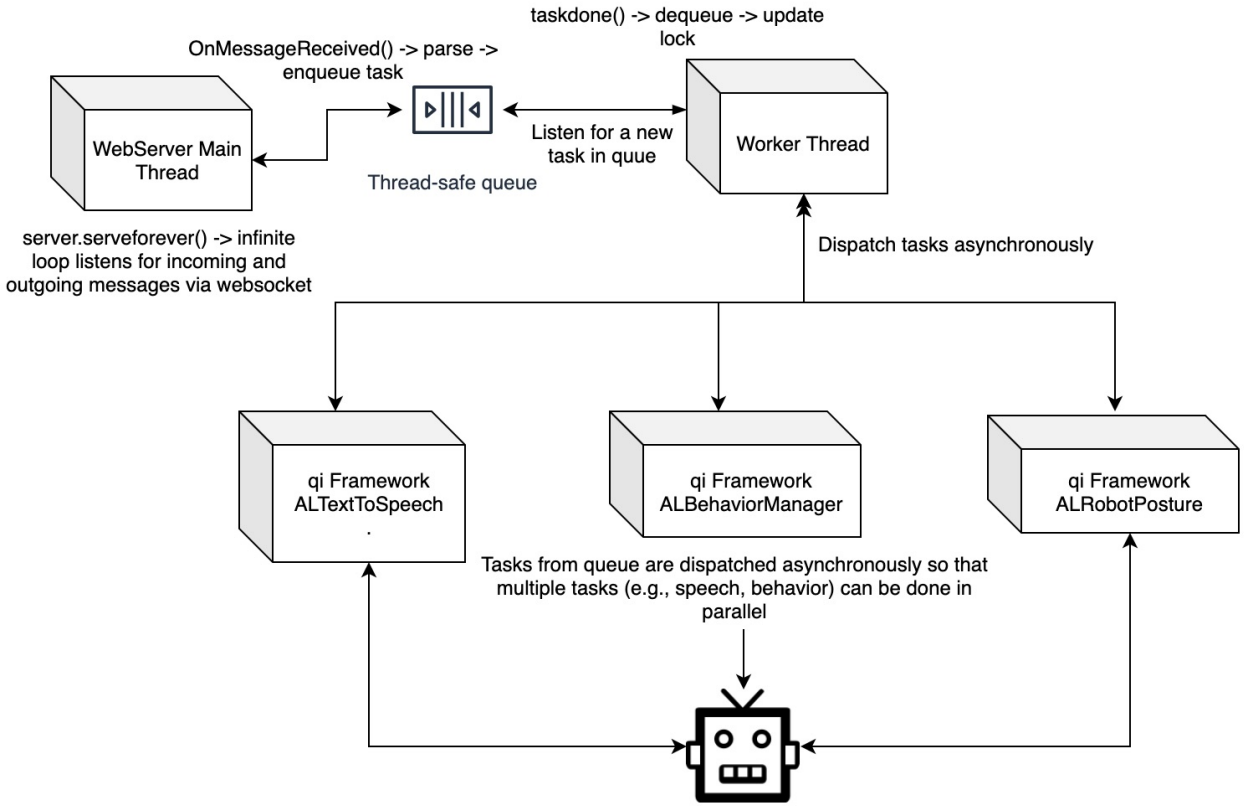

Figure 6.9 WebServer and Naoqi Robot Integration

1. The Main Thread

   The main thread runs in an infinite loop and looks for new messages that come from the Websocket port. All incoming messages are then parsed and categorized into appropriate tasks (for instance, making the robot talk, making the robot move its arm, etc.). Then, these processed tasks are enqueued into a thread-safe queue.

2. The Worker Thread

   The Worker Thread also runs in an infinite loop and has shared access to the thread-safe queue of the successfully processed tasks. For every iteration, the Worker Thread calls *queue.get()* to fetch a new task from the thread-safe queue. It blocks until there is something in that queue. After dequeueing a task, the Worker Thread determines the appropriate robot API that can handle it and then dispatches the task to that robot component. After a finished task, the Worker Thread signals to the queue that the task is done.

3. The Naoqi Components

   The tasks from the shared queue are dispatched asynchronously to the appropriate Naoqi API (e.g., ALTextToSpeech, ALBehaviorManager, ALRobotPosture) such that multiple tasks can be performed in parallel, such as text-to-speech and the robot's limb control. If such activities are carried out synchronously, the robot must execute them sequentially (e.g., stand up and then say "Hello World") which is an unnatural behavior. In addition, we implemented a simple algorithm to prevent waiting tasks from stacking up in the system memory. For example, if the amount of requests happens too quickly (e.g. more than ten requests per second) but each request takes 2-3 seconds for the robot to complete, the system will crash due to a large number of threads consuming the resources. This safety mechanism ensures long-term, non-disruptive interaction and adaptation.

6.2.10 Potential Integration with FNIRS

As these mental load profiles may dynamically change within or between tasks, we argue that an adaptive system is needed, which can be sensitive to both the current context and the predicted effect of potential choices of communication modality. Therefore, we are in the process of designing another human-subject experiment in which we can measure the participants' physiology in real time instead of relying on the subjective TLX survey. The fNIRS

component, developed by our collaborator, Dr. Leanne Hirshfield and team, at the University of Colorado at Boulder, handles raw data from sensor and outputs a multilabel vector consisting of four labels (workload, negative affect, auditory perceptual load, and visual perceptual load) from a multilabel long short-term memory (LSTM) classifier every second. Figure 6.10 demonstrate a potential integration. After our collaborator finishes the tedious experiment and model training, we will then connect our server with their sensor to finish the pipeline. This is still a work in progress. Then we plan on developing such an adaptive model for communication modality selection using probabilistic modeling techniques.

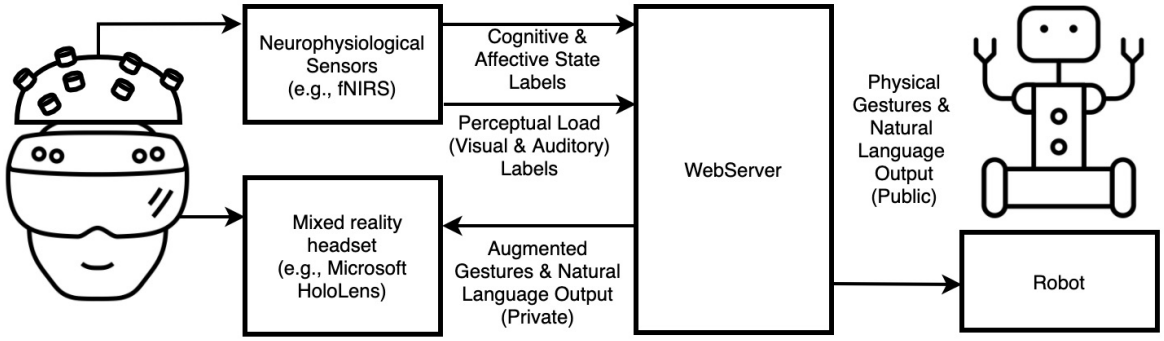

Figure 6.10 Potential integration between fNIRS and our system

REFERENCES CITED

[1] Tom Williams, Nhan Tran, Josh Rands, and Neil T Dantam. Augmented, mixed, and virtual reality enabling of robot deixis. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 257–275. Springer, 2018.

[2] Matthias Scheutz, Paul Schermerhorn, James Kramer, and David Anderson. First steps toward natural human-like hri. *Autonomous Robots*, 22(4):411–423, 2007.

[3] Nikolaos Mavridis. A review of verbal and non-verbal human–robot interactive communication. *Robotics and Autonomous Systems*, 63:22–35, 2015.

[4] Rehj Cantrell, Paul Schermerhorn, and Matthias Scheutz. Learning actions from human-robot dialogues. In *2011 RO-MAN*, pages 125–130. IEEE, 2011.

[5] Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 2020.

[6] Séverin Lemaignan, Raquel Ros, E Akin Sisbot, Rachid Alami, and Michael Beetz. Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *International Journal of Social Robotics*, 4(2):181–199, 2012.

[7] Geert-Jan M Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, Hendrik Zender, Ivana Kruijff-Korbayová, and Nick Hawes. Situated dialogue processing for human-robot interaction. In *Cognitive systems*, pages 311–364. Springer, 2010.

[8] Matthias Scheutz, Rehj Cantrell, and Paul Schermerhorn. Toward humanlike task-based dialogue processing for human robot interaction. *Ai Magazine*, 32(4):77–84, 2011.

[9] Susan Goldin-Meadow. The role of gesture in communication and thinking. *Trends in cognitive sciences*, 3(11):419–429, 1999.

[10] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. Learning from unscripted deictic gesture and language for human-robot interactions. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[11] Allison Sauppé and Bilge Mutlu. Robot deictics: How gesture and context shape referential communication. In *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 342–349. IEEE, 2014.

[12] Nhan Tran, Kai Mizuno, Trevor Grant, Thao Phung, Leanne Hirshfield, and Thomas Williams. Exploring mixed reality robot communication under different types of mental workload.

[13] Tom Williams, Daniel Szafir, Tathagata Chakraborti, and Heni Ben Amor. Virtual, augmented, and mixed reality for human-robot interaction. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 403–404. ACM, 2018.

[14] Daniel Szafir. Mediating human-robot interactions with virtual, augmented, and mixed reality. In *International Conference on Human-Computer Interaction*, pages 124–149. Springer, 2019.

[15] Tom Williams. A framework for robot-generated mixed-reality deixis. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.

[16] Tom Williams, Matthew Bussing, Sebastian Cabrol, Elizabeth Boyle, and Nhan Tran. Mixed reality deictic gesture for multi-modal robot communication. In *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction*, 2019.

[17] Tom Williams, Matthew Bussing, Sebastian Cabrol, Ian Lau, Elizabeth Boyle, and Nhan Tran. Investigating the potential effectiveness of allocentric mixed reality deictic gesture. In *International Conference on Human-Computer Interaction*, pages 178–198. Springer, 2019.

[18] Leanne Hirshfield, Tom Williams, Natalie Sommer, Trevor Grant, and Senem Velipasalar Gursoy. Workload-driven modulation of mixed-reality robot-human communication. In *Proceedings of the Workshop on Modeling Cognitive Processes from Multimodal Data*, page 3. ACM, 2018.

[19] Christopher D Wickens. Multiple resources and performance prediction. *Theoretical issues in ergonomics science*, 3(2):159–177, 2002.

[20] Nilli Lavie. The role of perceptual load in visual awareness. *Brain research*, 1080(1): 91–100, 2006.

[21] Alex Muhl-Richardson, Katherine Cornes, Hayward J Godwin, Matthew Garner, Julie A Hadwin, Simon P Liversedge, and Nick Donnelly. Searching for two categories of target in dynamic visual displays impairs monitoring ability. *Applied cognitive psychology*, 32 (4):440–449, 2018.

[22] Christopher D Wickens. Multiple resources and mental workload. *Human factors*, 50 (3):449–455, 2008.

[23] Ryan McKendrick, Raja Parasuraman, Rabia Murtza, Alice Formwalt, Wendy Baccus, Martin Paczynski, and Hasan Ayaz. Into the wild: neuroergonomic differentiation of hand-held and augmented reality wearable displays during outdoor navigation with functional near infrared spectroscopy. *Frontiers in human neuroscience*, 10:216, 2016.

[24] Paul Milgram, Shumin Zhai, David Drascic, and Julius Grodski. Applications of augmented reality for human-robot communication. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, volume 3, pages 1467–1472. IEEE, 1993.

[25] Carlotta Johnson, A Bugra Koku, Kazuhiko Kawamura, and R Alan Peters. Enhancing a human-robot interface using sensory egosphere. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 4132–4137. IEEE, 2002.

[26] Daniele Sportillo, Alexis Paljic, Luciano Ojeda, Giacomo Partipilo, Philippe Fuchs, and Vincent Roussarie. Learn how to operate semi-autonomous vehicles with extended reality. 2018.

[27] Heni Ben Amor, Ramsundar Kalpagam Ganesan, Yash Rathore, and Heather Ross. Intention projection for human-robot collaboration with mixed reality cues. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.

[28] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. Communicating robot arm motion intent through mixed reality head-mounted displays. In *Robotics Research*, pages 301–316. Springer, 2020.

[29] Elena Sibirtseva, Dimosthenis Kontogiorgos, Olov Nykvist, Hakan Karaoguz, Iolanda Leite, Joakim Gustafson, and Danica Kragic. A comparison of visualisation methods for disambiguating verbal requests in human-robot interaction. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 43–50. IEEE, 2018.

[30] Christopher Reardon, Kevin Lee, and Jonathan Fink. Come see this! augmented reality to enable human-robot cooperative search. In *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–7. IEEE, 2018.

[31] Tom Williams, Daniel Szafir, and Tathagata Chakraborti. The reality-virtuality interaction cube. In *Proceedings of the 2nd International Workshop on Virtual, Augmented, and Mixed Reality for HRI*, 2019.

[32] James P Bliss, John W Harden, and H Charles Dischinger Jr. Task shedding and control performance as a function of perceived automation reliability and time pressure. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 57, pages 635–639. SAGE Publications Sage CA: Los Angeles, CA, 2013.

[33] Fred GWC Paas, Jeroen JG Van Merriënboer, and Jos J Adam. Measurement of cognitive load in instructional research. *Perceptual and motor skills*, 79(1):419–430, 1994.

[34] JEREMY WOLFE. 16 theoretical and behavioral aspects of selective attention. *The cognitive neurosciences*, page 167, 2014.

[35] Daniel J Simons and Christopher F Chabris. Gorillas in our midst: Sustained inattentional blindness for dynamic events. *perception*, 28(9):1059–1074, 1999.

[36] James SP Macdonald and Nilli Lavie. Visual perceptual load induces inattentional deafness. *Attention, Perception, & Psychophysics*, 73(6):1780–1789, 2011.

[37] Nilli Lavie. Perceptual load as a necessary condition for selective attention. *Journal of Experimental Psychology: Human perception and performance*, 21(3):451, 1995.

[38] Pooja P Bovard, Kelly A Sprehn, Meredith G Cunha, Jaemin Chun, SeungJun Kim, Jana L Schwartz, Sara K Garver, and Anind K Dey. Multi-modal interruptions on primary task performance. In *International Conference on Augmented Cognition*, pages 3–14. Springer, 2018.

[39] Friedhelm Nachreiner. Standards for ergonomics principles relating to the design of work systems and to mental workload. *Applied Ergonomics*, 26(4):259–263, 1995.

[40] Ben D Sawyer, Victor S Finomore, Andres A Calvo, and Peter A Hancock. Google glass: A driver distraction cause or cure? *Human factors*, 56(7):1307–1321, 2014.

[41] Alexander Woodham, Mark Billinghurst, and William S Helton. Climbing with a head-mounted display: dual-task costs. *Human factors*, 58(3):452–461, 2016.

[42] Laura M Naismith, Jeffrey JH Cheung, Charlotte Ringsted, and Rodrigo B Cavalcanti. Limitations of subjective cognitive load measures in simulation-based procedural training. *Medical education*, 49(8):805–814, 2015.

[43] S.G. Hart and L.E. Staveland. *Development of NASA-TLX (Task Load Index): Results of empirical and theorical research*, pages pp 139 – 183. Amsterdam, 1988.

[44] John Sweller. Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4):295–312, 1994.

[45] John Sweller and Paul Chandler. Why some material is difficult to learn. *Cognition and instruction*, 12(3):185–233, 1994.

[46] John Sweller, Jeroen JG Van Merrienboer, and Fred GWC Paas. Cognitive architecture and instructional design. *Educational psychology review*, 10(3):251–296, 1998.

[47] Kristel Knaepen, Andreas Mierau, Eva Swinnen, Helio Fernandez Tellez, Marc Michielsen, Eric Kerckhofs, Dirk Lefeber, and Romain Meeusen. Human-robot interaction: does robotic guidance force affect gait-related brain dynamics during robot-assisted treadmill walking? *PloS one*, 10(10), 2015.

[48] Amanda K Robinson, Praveen Venkatesh, Matthew J Boring, Michael J Tarr, Pulkit Grover, and Marlene Behrmann. Very high density eeg elucidates spatiotemporal aspects of early visual processing. *Scientific reports*, 7(1):1–11, 2017.

[49] Marco Ferrari, Leonardo Mottola, and Valentina Quaresima. Principles, techniques, and limitations of near infrared spectroscopy. *Canadian journal of applied physiology*, 29(4):463–487, 2004.

[50] Kurtulus Izzetoglu, Scott Bunce, Banu Onaral, Kambiz Pourrezaei, and Britton Chance. Functional optical brain imaging using near-infrared during cognitive tasks. *International Journal of human-computer interaction*, 17(2):211–227, 2004.

[51] Erin Treacy Solovey, Audrey Girouard, Krysta Chauncey, Leanne M Hirshfield, Angelo Sassaroli, Feng Zheng, Sergio Fantini, and Robert JK Jacob. Using fnirs brain sensing in realistic hci settings: experiments and guidelines. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 157–166. ACM, 2009.

[52] Abdul Serwadda, Vir V Phoha, Sujit Poudel, Leanne M Hirshfield, Danushka Bandara, Sarah E Bratt, and Mark R Costa. fnirs: A new modality for brain activity-based biometric authentication. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2015.

[53] Cody Canning and Matthias Scheutz. Functional near-infrared spectroscopy in human-robot interaction. *Journal of Human-Robot Interaction*, 2(3):62–84, 2013.

[54] Lei Hou, Xiangyu Wang, Leonhard Bernold, and Peter ED Love. Using animated augmented reality to cognitively guide assembly. *Journal of Computing in Civil Engineering*, 27(5):439–451, 2013.

[55] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[56] Nirit Gavish, Teresa Gutiérrez, Sabine Webel, Jorge Rodríguez, Matteo Peveri, Uli Bockholt, and Franco Tecchia. Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, 23 (6):778–798, 2015.

[57] Markus Funk, Thomas Kosch, and Albrecht Schmidt. Interactive worker assistance: comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 934–939, 2016.

[58] Sevda Küçük, Samet Kapakin, and Yüksel Göktaş. Learning anatomy via mobile augmented reality: effects on achievement and cognitive load. *Anatomical sciences education*, 9(5):411–421, 2016.

[59] John K Chapin, Karen A Moxon, Ronald S Markowitz, and Miguel AL Nicolelis. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature neuroscience*, 2(7):664, 1999.

[60] Johan Wessberg, Christopher R Stambaugh, Jerald D Kralik, Pamela D Beck, Mark Laubach, John K Chapin, Jung Kim, S James Biggs, Mandayam A Srinivasan, and Miguel AL Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408(6810):361–365, 2000.

[61] Parag G Patil, Jose M Carmena, Miguel AL Nicolelis, and Dennis A Turner. Ensemble recordings of human subcortical neurons as a source of motor control signals for a brain-machine interface. *Neurosurgery*, 55(1):27–38, 2004.

[62] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[63] Mikhail A Lebedev, Roy E Crist, and Miguel AL Nicolelis. 11 building brain–machine interfaces to restore neurological functions. *Methods for Neural Ensemble Recordings*, page 219, 2007.

[64] Jd R Millan, Frederic Renkens, Josep Mourino, and Wulfram Gerstner. Noninvasive brain-actuated control of a mobile robot by human eeg. *IEEE Transactions on biomedical Engineering*, 51(6):1026–1033, 2004.

[65] Ferran Galán, Marnix Nuttin, Eileen Lew, Pierre W Ferrez, Gerolf Vanacker, Johan Philips, and J del R Millán. A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots. *Clinical neurophysiology*, 119(9):2159–2169, 2008.

[66] Thorsten O Zander and Christian Kothe. Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of neural engineering*, 8(2):025005, 2011.

[67] Gert Pfurtscheller, Christa Neuper, Christoph Guger, WAHW Harkam, Herbert Ramoser, Alois Schlogl, BAOB Obermaier, and MAPM Pregenzer. Current trends in graz brain-computer interface (bci) research. *IEEE transactions on rehabilitation engineering*, 8(2):216–219, 2000.

[68] Hasan Ayaz, Patricia A Shewokis, Scott Bunce, Maria Schultheis, and Banu Onaral. Assessment of cognitive neural correlates for a functional near infrared-based brain computer interface system. In *International Conference on Foundations of Augmented Cognition*, pages 699–708. Springer, 2009.

[69] Daniel Szafir and Bilge Mutlu. Pay attention! designing adaptive agents that monitor and improve user engagement. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 11–20, 2012.

[70] Audrey Girouard, Erin Treacy Solovey, and Robert JK Jacob. Designing a passive brain computer interface using real time classification of functional near-infrared spectroscopy. *International Journal of Autonomous and Adaptive Communications Systems*, 6(1):26–44, 2013.

[71] Diane M Beck and Nilli Lavie. Look here but ignore what you see: effects of distractors at fixation. *Journal of Experimental Psychology: Human Perception and Performance*, 31(3):592, 2005.

[72] Tom Williams, Matthew Bussing, Sebastian Cabrol, Ian Lau, Elizabeth Boyle, and Nhan Tran. Investigating the potential effectiveness of allocentric mixed reality deictic gesture. In *Proceedings of the 11th International Conference on Virtual, Augmented, and Mixed Reality*, 2019.

[73] JASP Team. Jasp (version 0.8.5.1)[computer software], 2018.

[74] EJ Wagenmakers, J Love, M Marsman, T Jamil, A Ly, and J Verhagen. Bayesian inference for psychology, Part II: Example applications with JASP. *Psychonomic Bulletin and Review*, 25(1):35–57, 2018.

[75] Martin J Crowder. *Analysis of repeated measures*. Routledge, 2017.

[76] RD Morey and JN Rouder. Bayesfactor (version 0.9. 9), 2014.

[77] Jeffrey N Rouder, Richard D Morey, Paul L Speckman, and Jordan M Province. Default bayes factors for anova designs. *Journal of Mathematical Psychology*, 56(5):356–374, 2012.

[78] S. Mathôt. Bayes like a baws: Interpreting bayesian repeated measures in JASP [blog post]. https://www.cogsci.nl/blog/interpreting-bayesian-repeated-measures-in-jasp, May 2017.

[79] Harold Jeffreys. Significance tests when several degrees of freedom arise simultaneously. *Proc. Royal Society of London. Series A, Math. and Phys. Sci.*, 1938.

[80] Peter H Westfall, Wesley O Johnson, and Jessica M Utts. A bayesian perspective on the bonferroni adjustment. *Biometrika*, 84(2):419–427, 1997.

[81] James O Berger and Luis R Pericchi. The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996.

[82] Andrew F Jarosz and Jennifer Wiley. What are the odds? a practical guide to computing and reporting bayes factors. *The Journal of Problem Solving*, 7(1):2, 2014.

[83] Shelton Jose and Kumar Gideon Praveen. Comparison between auditory and visual simple reaction times. *Neuroscience & Medicine*, 2010, 2010.

[84] Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. An overview of the distributed integrated cognition affect and reflection diarc architecture. In *Cognitive Architectures*, pages 165–193. Springer, 2019.

[85] Tom Williams and Matthias Scheutz. Referring expression generation under uncertainty: Algorithm and evaluation framework. In *Proceedings of the 10th International Conference on Natural Language Generation*, 2017.

[86] Allen G Taylor. Getting started with hololens development. In *Develop Microsoft HoloLens Apps Now*, pages 19–46. Springer, 2016.

[87] Gabriel Evans, Jack Miller, Mariangely Iglesias Pena, Anastacia MacAllister, and Eliot Winer. Evaluating the microsoft hololens through an augmented reality assembly application. In *Degraded Environments: Sensing, Processing, and Display 2017*, volume 10197, page 101970V. International Society for Optics and Photonics, 2017.

[88] Jason Odom. *HoloLens Beginner's Guide*. Packt Publishing Ltd, 2017.

[89] Harald Lampesberger. Technologies for web and cloud service interaction: a survey. *Service Oriented Computing and Applications*, 10(2):71–110, 2016.

[90] Ehsan Azimi, Long Qian, Nassir Navab, and Peter Kazanzides. Alignment of the virtual scene to the 3d display space of a mixed reality head-mounted display. *arXiv preprint arXiv:1703.05834*, 2018.