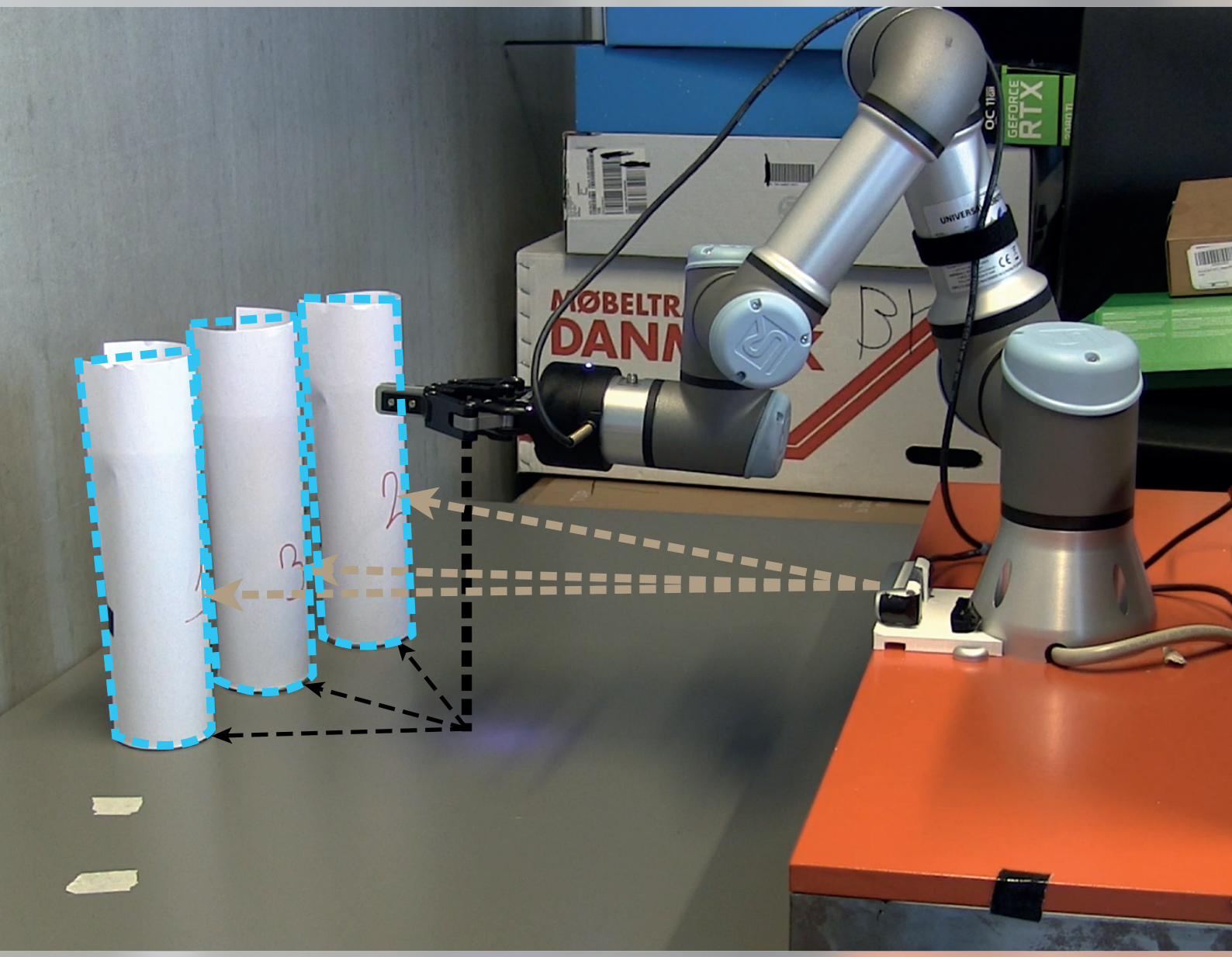
Exploring Shared Control to Improve Self-Sufficiency of Tetraplegics

Assistive Robotics using Multimodal Intent Prediction



Group 1001
Autumn and Spring
Long Master Thesis
Vision, Graphics, and Interactive Systems

AALBORG UNIVERSITY
DENMARK

Oliver Gyldenberg Hjermitslev

Frederik Falk

AALBORG UNIVERSITY
DENMARK

Vision, Graphics, and Interactive Systems
Aalborg University
http://www.aau.dk

Title:
Exploring Shared Control to Improve Self-Sufficiency of Tetraplegics

Theme:
Master Thesis

Project Period:
Autumn 2018 & Spring 2019

Project Group:
VGIS 1001

Participants:
Hjermitslev, Oliver Gyldenberg
Falk, Frederik

Supervisor:
Thomas B. Moeslund
Stefan H. Bengtson

Page Numbers: 113

Date of Completion:
June 3, 2019

Abstract:

Currently, tetraplegics have limited opportunities to perform activities of daily living (ADL) independently from caregivers. This project explores the possibility of introducing a shared control system based on computer vision and multimodal intent prediction. Following a review of previous work, we design a solution to improve simple interactions necessary for ADL. This system utilizes galvanic skin response and a novel intent prediction method based on previous user input. An evaluation with 24 able-bodied people was conducted to gather both subjective and objective data about the interactions and performance. Evaluation shows that aggressive arbitration can be a hindrance in certain measurements, but a tradeoff exists which requires more work and a longer, more comprehensive evaluation to define.

Acknowledgements

We would like to thank our supervisors Thomas B. Moeslund and Stefan H. Bengtson for their help with questions, providing their assistance with decisions, and project directions, as well as provide their insight and knowledge when needed.

We would also like to thank Hendrik Knoche for the usage of his Shimmer3 GSR+ Sensor device.

Contents

Danish Summary

I dette projekt har vi arbejdet tangentielt til EXOTIC, et større tværfagligt videnskabsprojekt som har til hensigt at udvikle tungestyrede exoskeletarme til tetraplegikere. I øjeblikket har borgere med tetraplegi begrænsede muligheder for at udføre dagligdagsaktiviteter uafhængigt af omsorgsgivere. Tests med en robotarm viser, at disse borgere bruger lang tid og mange justeringer på at styre armen for eksempelvis at samle objekter op. Vi undersøger mulighederne for at forudsige hensigter under disse forhold til brug i et delekontrolsystem for at assistere med disse aktiviteter og øge slutbrugernes livskvalitet.

Vi undersøger derfor *state of the art* teknologier og metoder og implementerer originale løsninger for at skabe et multimodalt system, som forudsiger brugeres hensigt på baggrund af tidligere input og fysiologisk data i realtid for at assistere brugere mod målet. Vi evaluerer fire forskellige metoder mod kontrolprøver og indsamler både subjektiv og objektiv data for at svare på, om disse metoder kan forbedre de nuværende omstændigheder. De fire metoder indeholder både aggressive og tilbageholdende arbitreringskurver samt elektrodermal aktivitet.

Vi påviser at valget af metode har en indflydelse på de undersøgte parametre, men at videreudvikling med fokus på at optimere arbitreringsmetoden er en nødvændighed. Derudover viser vores resultater, at der er stor forskel på brugere, og vi foreslår derfor et længere studie med mere specifikke testopgaver og et færdigt system, der kan tilpasses hver enkelt bruger.

1 | Introduction

In this project we work alongside EXOTIC, a large long-term cross-faculty research project which aims to provide tongue-controlled exoskeletal robot arms to tetraplegic citizens. Currently, citizens with tetraplegia have limited possibilities of performing activities of daily living independently. Tests with a robot arm show that much of their time is spent on performing adjusting movements. We investigate the impacts of intent prediction for shared control in order to assist users with these activities and improve their quality of life. We investigate state of the art technology and methods and implement novel solutions to create a multimodal system which predicts intent based on previous input and physiological data in order to assist users with reaching for goals with a robot arm. The evaluation yielded both subjective and objective data from 24 able-bodied participants using the system to reach three sequential goals with multiple arbitration modes and compare them to control data.

2 | Motivation

In Denmark, an estimated 3000 people live with spinal cord injury (SCI), and an additional 130-160 suffer such an injury each year [1]. This includes both traumatic and non-traumatic injuries. The subsequent damage to the nerves leaves roughly 40% tetraplegic [2]. Recent studies indicate that 43% of tetraplegics are less than 30 years old [3]. This means that these individuals are bound to live long lives with extreme disability and greatly decreased quality of life, fully dependent on caretakers. Despite the physical restrictions most of the victims of SCI maintain their cognitive abilities. This means that these citizens are fully aware of the effects of their disability, and the strains on their family- and social relations, which can lead to anxiety and depression [4].

Another increasing problem is the distribution of caretakers and care receivers, which is challenged by demographic development. This can be seen in the different municipalities in Denmark, for instance Århus Kommune shows a 29% increase in the demand for caregivers from 2008-2017 [5]. Furthermore, the current cost of caring for one tetraplegic person often exceeds 1,300,000 DKK yearly [6]. Depending on the patient's age when injured, this is often sustained for 40 or more years which totals 52 million DKK. SCI related tetraplegia in Denmark has an estimated yearly cost of almost two billion DKK based on the yearly cost and the number of people affected. Allowing the affected to reenter the job market can increase their quality of life [7]. Furthermore, studies suggest that enabling certain activities of daily living (ADL) such as eating and drinking could reduce the amount of assistance required by up to 41% and potentially increase quality of life [8].

Lately, shared control of robotic systems between human and computer in cases

of paralysis has been investigated more and more. Shared control interfaces can be difficult, as the level of autonomy required is different from user to user and from task to task [9]. The concept of shared control is especially important in extreme cases where users can only have limited control, for example due to paralysis, requiring a prediction of the user's intent [10].

Myoelectric robotics have a range of benefits, mainly that users can control them using their existing bodily functions and transfer their existing skills. However, they can be difficult to navigate, and training may take longer than previously anticipated [11]. Additionally, the controls are not usable for everyone; tetraplegics are unable to use them, as they have no control of their muscles.

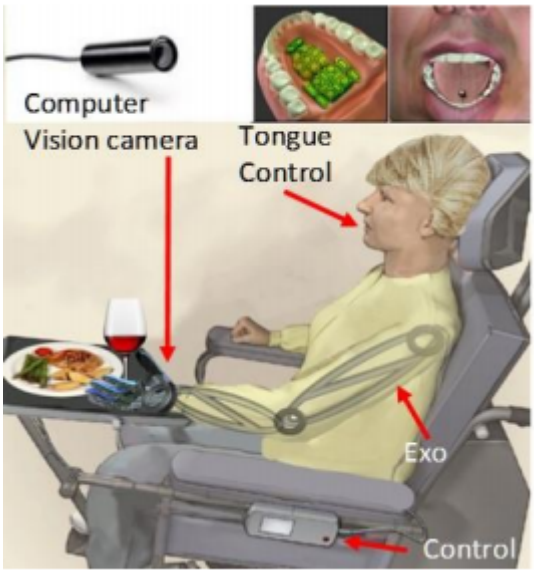


Figure 2.1: An example use case of the EXOTIC project, detailing the control scheme. Preliminary internal sketch

A proposed idea for these issues is the use of an assistive personal robotics platform with an exoskeleton, using the tongue for intelligent control (EXOTIC), as the tongue is unaffected by loss of limb control caused by SCI [12]. An early concept sketch is shown in Figure 2.1. The EXOTIC project aims to finish in 2021, meaning the research in this paper will be conducted tangentially with the project. Using the tongue-based control scheme would allow patients to control a robotic arm with their tongue and a personalized palate hanger. A problem with the palate hanger

is that the instability in the patient's tongue could translate to clunky movements of the robot arm, making it harder to grasp objects. As EXOTIC aims to implement intelligent control, a variety of sensor information about both the user and the surroundings could assist during difficult movements to improve usability and thus quality of life. This sensor information should be readily available, non-invasive, and not require additional active inputs from the user in order to ease interactions.

2.1 Initial Problem Statement

Based on the information and issues presented, we intend to study how shared control has been used in previous work in both a general sense and in rehabilitative areas and how this research can be used to define a shared control system using tongue controlled robot prosthetics allowing tetraplegics to perform ADL without aid. This entails an extensive literature review to pinpoint state of the art methods of shared control, intent prediction, and evaluation methods. Specifically, we seek to assess a selection of methods for intent analysis that apply to EXOTIC, their methods of development, and how to evaluate their capabilities. This can be summed up in the following problem statement:

> *How can shared control concepts be transferred to a tongue-controlled robotic arm to allow tetraplegic users a chance to perform activities of daily living?*

The specific knowledge required to realize this project can be divided in a number of research questions:

- What are state of the art intent prediction methods, and how do they transfer to EXOTIC?
- What scenarios necessitate shared control, and how can this process be streamlined to improve user satisfaction?
- What evaluation methods have been previously utilized and how do they fit into this project?
- What restrictions does the use of tongue controls impose on the project?

Summary

An increasing number of citizens in Denmark suffer from SCI, and the quality of care is increasingly strained due to a larger demographic change. Constant care for these individuals is expensive for the state and taxing on their mental health. Shared control between humans and robots has potential to improve this situation, allowing users to participate in ADL and increase their quality of life. EXOTIC proposes an intelligent system to assist with ADL using an exoskeletal robotic arm. We suggest an approach to analysing state of the art shared control concepts and previous work and a number of research questions to learn how to use this knowledge in the context of the EXOTIC project.

3 | Background Research

This chapter attempts to answer the research questions defined in the Motivation. Specifically, we investigate state of the art, previous work, and relevant scientific theory to determine previous solutions to similar problems and their ability to transfer to our specific case. This also requires an analysis of what issues and opportunities these might incur.

3.1 EXOTIC

As the EXOTIC project and this project are tangentially related, a clear definition between the two is required. In this section, the scope of this project is determined in relation to the overall scope of the EXOTIC project.

Figure 3.1: EXOTIC concept with all its research topics. Topics researched in this project are outlined in orange

Figure 3.1 shows the different research topics the EXOTIC project entails. This project only focuses on those marked in orange. The subjects outlined here need to be researched in a literature- and state-of-the-art review in order to understand their impact and the scope with which they can answer the questions posed in the Initial Problem Statement. Additionally, tangentially related and still important topics need to be understood such that this project transfers to the overall EXOTIC project.

3.2 Intraoral Based Controls

Struijk et al. has researched the use of intraoral tongue based computer interfaces as the cranial nervous innervation often leaves the tongue with its sensory motors intact even after high level spinal cord injuries [13]. However, none of the current systems have sufficient control schemes to facilitate full 3-dimensional controls of a robot arm with functionality resembling a human arm. This is required for robot arms to achieve full functionality in ADL which ideally reduces the amount of personal aid

required [8].

Struijk et al. used a modified version of the commercial product iTongue® with 18 programmable inductive sensors. Two participants were selected based on their previous experience with tongue control interfaces. The participants had 30 minutes of training time before being asked to perform a sequence of tasks, followed by picking up a roll of tape. Picking up the roll of tape had a success rate of 80% in 10 attempts.

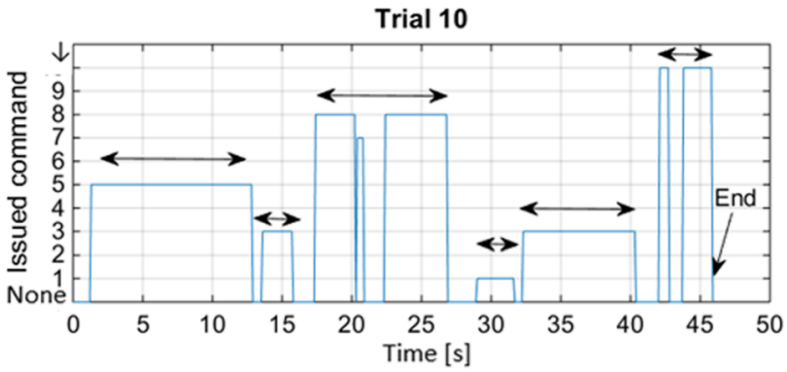


Figure 3.2: Commands issued over time. Picture source: [13]

Figure 3.2 shows the issued commands over time. Interesting to note is that the participants often corrected their initial movements to better align the robot arm with the object. This means that the total amounts of movements could potentially be reduced if an optimal path could be estimated through a collaborative interaction with a computer.

3.3 Shared Control

Shared control is a concept involving the interaction and collaboration between human and computer in tasks involving robot operability [14, 15]. Specifically, it refers to the assistance provided by each operator to the other. In many cases, this means that the human initiates an action and the computer adjusts the movement to a level of accuracy humans cannot reach, but it can also work the other way around, where a human takes control of an automated task.

Implementing shared control in a robotic system is a complicated task; the amount

of control given to the human operator can vary significantly [16, 17]. In many situations, human-robot collaboration is a necessity [18]. Thus, the levels of shared control is key to develop successful human-robot interaction.

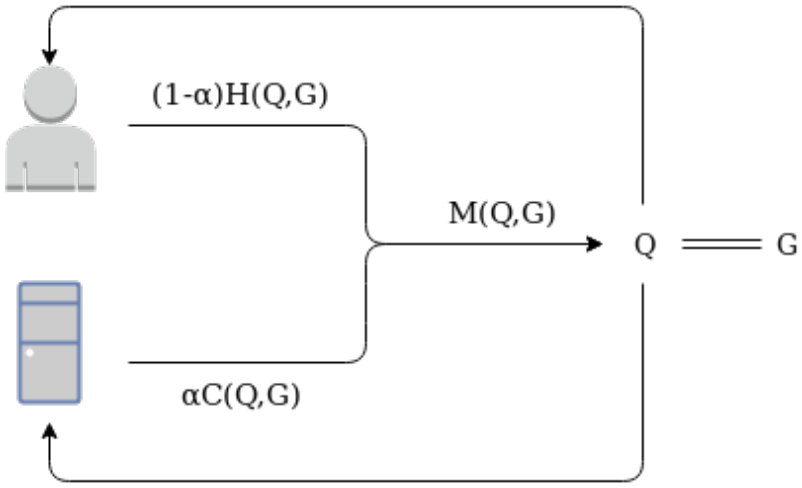


Figure 3.3: An illustration of the shared control paradigm, showing user and computer working together to achieve a goal

Shared control can be modelled as shown in Figure 3.3, where a human user and a computer attempt to complete a goal, denoted G. The human's action, denoted $H(Q, G)$, and the computer's action, denoted $C(Q, G)$, are blended using a variable α to alter the state Q. This action, or movement, is here denoted $M(Q, G)$ and results in a new Q which in turn affects the actions H and C. When $Q = G$ the goal has been reached and the collaboration can end.

This method can be extended endlessly; for example by making a series of goals, $G_{[0,1,2,..,n]}$ that the two actors must reach in unison, or by updating α on-line based on a number of variables as indicated by the current state, making α itself variable as $\alpha = U(Q, G)$, where U denotes some update function.

3.4 Intent Prediction

Much of the previous work, even recently, in the field of shared control has been made when the goal was initially clear for the computer as well as the human operator

[19, 20, 21]. However, in many real world situations this is a significant restriction. Since goals cannot always be known in changing environments, intent prediction can be used to estimate the goal of a human user's action [9]. This also necessitates a system that can detect and track objects in the scene in real time. A variety of methods have been utilized to optimize intent prediction in order to improve a shared control experience, optimizing the blend factor α either as a continuous or step function. Some of the relevant technologies are described below.

3.4.1 Movement

When operating a robotic arm, users' movements of the robot can be used to estimate the current goal [9, 22]. Dragan and Srinivasa show a version of shared control similar to Figure 3.3, except here the user has a direct influence of the computer's preferred movement, not just tangentially via an update of Q [9]. In their work, they predict the goal from a set based on the most likely candidate that best optimizes some cost function to move to that goal.

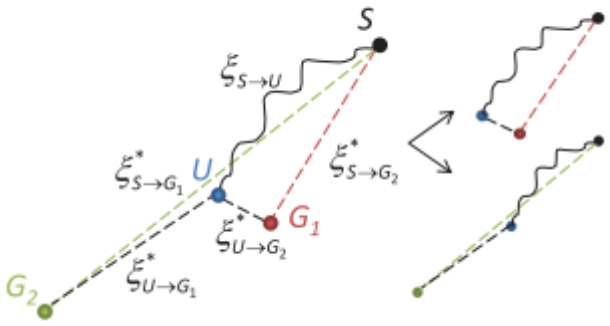


Figure 3.4: The predicted goal is one that most closely resembles the optimal trajectory. Source: [9]

In Figure 3.4, Dragan and Srinivasa predict Goal 2 (G_2) to be correct, not the closer Goal 1 (G_1), as the user's trajectory more closely resembles the optimal trajectory to G_2. The work goes on to discuss trajectory prediction, and makes an important point:

> "The idea behind sharing the autonomy is that the robot executes what the user actually wants rather than what the user command [9]"

User input is bound to be noisy, and they instead seek to predict the optimal path to the goal, not necessarily the path the user laid out.

Similarly, most work appears to be focused on Maximum A Posteri probability of some action being the result of one of many intentions and altering the trajectory based on this estimate. Wang et al. have proposed the Intention-Driven Dynamics Model (IDDM) based on the Gaussian Process Dynamical Model and taught a robot to predict strokes in a game of table tennis [23]. The IDDM directly incorporates the intention into its transition function, similar to Dragan and Srinavasa. The trained model can estimate a posterior probability using Bayes' rule:

$$p(g|z_{1:T}) = \frac{p(z_{1:T}|g)p(g)}{p(z_{1:T})} \tag{3.1}$$

Where g is the goal and $z_{1:T}$ are the observations.

In the same vein, Hauser evaluated a 6 Degrees of Freedom (DoF) robot arm operating on 2D mouse movement inputs [24]. This system operated on Gaussian mixture autoregression in order to distiguish between static and dynamic targets. Anticipating the users' intent, the system was successful in generating higher quality trajectories and improving performance.

Monfort, Liu, and Ziebart suggest a control scheme based on previous work on anticipatory temporal conditional random fields in which learned behaviour is used to predict a trajectory [25]. Here, the model is trained on a number of depth-camera recordings, and the prediction is based on a linear dynamics model and a quadratic cost function. Again, the model predicts based on the minimal cost expectations, and attempt to reason an optimal trajectory based on its learned information and the input.

In conclusion, it appears much of this area is predicated on building a model that, with or without prior learning, chooses a target from a set of targets discovered in real time based on minimizing some cost function, choosing the maximum probability and acting on that information.

3.4.2 Gaze

With the advent of driverless vehicles and human-robot collaboration in general, research in intent prediction using users' gaze has seen an increase [26, 27, 28, 29]. The idea predicates on users showing their intent by focusing their gaze on the target of their action. This is evident already in infancy, as even small children can learn from and communicate intent using gaze [30, 31].

Figure 3.5: The Tobii EyeX and Pro Glasses 2

In Zeng et al., a multimodal system utilizing gaze tracking controls a myoelectrically steered robotic arm [32]. The system uses a commercial eye tracker, the Tobii EyeX, shown in Figure 3.5 (left), to estimate the user's gaze. They filter the input from this modality using a 10-point moving average (at 60 hz) to reduce noise. Díez et al. also used a Tobii eye tracker, the Pro Glasses 2, shown in Figure 3.5 (right) [33]. In both studies, colour data from the video feed determine the distinct objects the user is looking at, the former using RGB and the latter using HSV. In these studies, the gaze highlights and selects which object to reach. This way, the user and robot collaborate on reaching and grasping an object. Crea et al. also incorporate a GUI for users to guide robot actions such as grasping and lifting an object [34].

Considering eye movement can be involuntary or meaningless, machine learning has been utilized to distinguish between intentional and unintentional gaze [28]. This research has focused on pupil dilation as well as fixations (focusing on a specific point) and saccades (quick eye movements between fixations) for predicting intent. According to them, eye movements were more accurate as predictors than pupil

dilation with 0.8 Area Under Curve (AUC) as opposed to 0.6 AUC.

Gigli et al. have created a method to determine when fixation precede a grasp in their study using a myoelectric prosthetic, looking for increases in muscle activity as well as reduced gaze volatility [35]. Gaze volatility is measured against a threshold updated every 0.5 seconds and represents "a sequence of gaze points X as Euclidean variance around the centroid". This can be denoted as such:

$$v(X, t, \tau_{gaze}) = \frac{1}{\tau_{gaze}} \sum_{i=t}^{t-\tau_{gaze}} \|x_i - \mu(X_{t:t-\tau_{gaze}})\|_2 \tag{3.2}$$

In this case, they define τ_{gaze} as 300 ms based on previous tests. The equation simply describes the normalized sum of distances from the centroid in the period of 300 ms.

In conclusion, gaze is a useful metric to determine intent and several previous studies have used it in attempts to improve human computer interaction. Specifically, we have presented evidence that fixations and saccades are important determinants of planned actions, more so than pupil dilations, and methods to implement fixation estimates, both programmatically and physically. However, gaze can be expensive to implement, both financially and computationally. Additionally, it can be somewhat intrusive, requiring users to wear specially designed glasses or install additional hardware on their wheelchair to perform activities.

3.4.3 Physiology

Brain-machine interfaces (BMI) have proven to be suitable for robotic control even without muscular control [10, 36]. However, in order to stay as uninvasive as possible researchers have worked increasingly with electroencephalography (EEG) rather than methods requiring medical procedures [32]. This study employed a commercial EEG measurement device, the EMOTIV EPOC+, to collect EEG data.

Figure 3.6: The EMOTIV EPOC+ headset

This device is shown in Figure 3.6. They use the OpenVibe toolbox to calibrate and analyse data from this headset, ultimately receiving commands from the user to the machine through this software. This method proved to be rather successful, ranging from 77.4% to 94.0% cross-validation classification accuracy, however it is important to note that the study was performed on eight able-bodied test participants and tested binary classification.

Lampe et al. performed a similar experiment, utilizing both real and imagined movements of left-hand finger tapping, right-hand finger tapping, toe clenching, as well as a relaxed state, to calibrate and test the modality [37]. The real movements elicited stronger EEG signals, but are not possible for paralyzed patients. Online results of classification accuracy using imagined movements range from 28.1% to 60.4% with a sample size of 5 able-bodied adults.

In Schirner et al's work, they describe a number of modalities to assist human-in-the-loop systems, among which are EEG [38]. In addition, they name blood pressure and skin conductivity, among others, as a significant improvement to this type of system as they can express stress or similar factors that cannot be otherwise described to a computer. These modalities can make it possible to understand intent as well as the condition the user signalling the intent is in. Galvanic Skin Response (GSR) sensors have the benefit that, unlike EEG, they are small and uninvasive, as shown in Figure

Figure 3.7: Shimmer3 GSR sensor unit

3.7. This is highly convenient if the goal is to make a system easily equippable and visually appealing. Skin conductivity has previously been studied in human-robot collaboration, namely as an indicator of comfort or engagement [39, 40]. This could potentially be transferred to a robot control system wherein GSR can signify comfort or stress about a robot's decision-making and assistance.

In conclusion, several physiological responses have previously been investigated in relation to robotic control systems, EEG being mostly studied for its role as a controller and GSR as an indicator of mental state. We also describe specific hardware capable of recording such data if needed.

3.4.4 Facial expressions

In recent years the amount of time spent watching videos online has drastically increased, including the amount of advertisements. According to Mcduff et al. predicting the likeability of advertisements is possible using facial expressions [41]. The participants were recruited from four different countries and totalled over 12000 people. The participants were asked to watch 10 video advertisements and score them immediately after watching, while their facial expression was recorded through a webcam. Comparing the facial expressions with the survery responses, they were able to achieve a prediction accuracy of 0.85. It is important to note that not all users

show significant facial expressions nor are they always an indicator of intent in the users. Additionally, neural networks have proven to be able to recognize eight distinct human emotions with an accuracy of 0.96, potentially enabling powerful predictions of intent using facial expressions [42].

3.4.5 Affordance in Robotics and Computer Vision

Affordance is the understanding of the possible set of actions an object can have in an environment [43]. This means that affordance is directly linked to the agent, while *function understanding* aims to describe all the possible actions with a given object. These concepts are both important to humans and intelligent computer agents, however they are highly complex to integrate. Initially, objects need to be recognized and have their location determined. Secondly, it is necessary to know the geometry and orientation of the object. This can become troublesome with objects that have multiple affordances, as traditional classification and detection is not possible; for instance, a bed can be both *sittable* and *layable*. This also implies that an object can have a chain of affordances, a cup is first *graspable*, then *liftable*, and finally *pourable*. However, knowing the affordance of the objects in the scene could improve the intent prediction as it is able to disregard objects that do not have an initial use and rank relevant objects higher. This is also the case when, for instance, a glass is empty and there is a water bottle next to it. Using this context the system could be more precise in its prediction of the user's intent as it knows there is no initial use of the glass.

Summary

A variety of modalities have been discussed in this section which are all usable in a shared control system. Using the movements or trajectories of the robot arm has shown to be a satisfying solution to estimate which target from a group the movement is aimed at in real time. It can, however, become a troublesome task if many objects are too close to each other or if there is a lot of occlusion between objects. Additionally, many passive sensors have been discussed partly as a means of assisting with this issue. Gaze has become popular in human-robot collaboration

and have shown impressive accuracy of intent prediction. There are many more physiologic measures available to assist in this process, but they are often used to improve current prediction by measuring the user's reactions to actions or in binary tasks. Some of these measurement devices are invasive or obstructive, such as EEG, while others are much less so, like a GSR sensor. Facial expressions have shown to be a possible input modality for expressing like or dislike of observed events. Additionally, affordance can potentially assist intent prediction by utilizing known context clues.

3.5 Arbitration

Implementing shared control systems can become troublesome when defining the system's aggressiveness, that is how quickly or how much the system assists the user. It requires a measure to determine when, or if, the human should have complete control and how and when the robot should increase the control of the system. The confidence of the system could be used as such a measure.

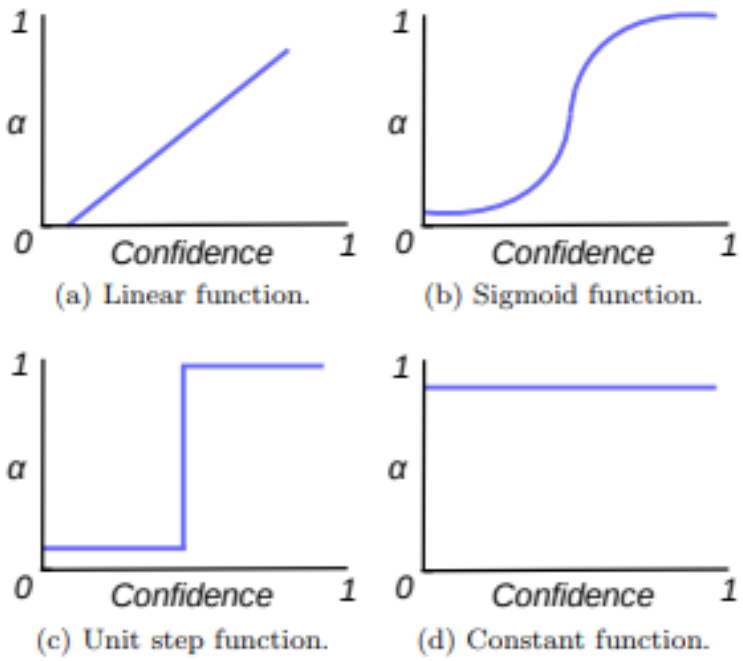


Figure 3.8: Different arbitration functions. Picture source: [44]

Figure 3.8 shows a variety of blending policies as a function of the confidence of the system. An important prerequisite for these blending methods is to accurately calculate the confidence of the system. Possible candidates to this problem are outlined in Section 3.4. It is important to recognize that the confidence of the system is dependent on the user and most likely will have to be fitted to the individual users.

In Yu et al, they describe that shared control systems used to assist users cognitively or physically should only assist the user rather than replacing them to ensure adequate performance [45].

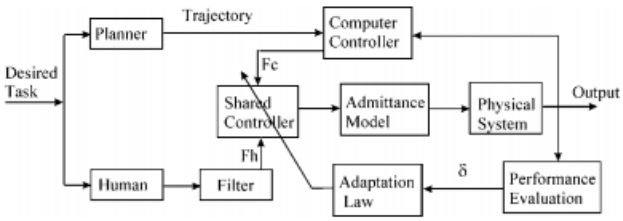


Figure 3.9: Suggested adaptive control for PAMM. Picture source: [45]

Figure 3.9 shows Yu et al's proposed adaptive control scheme. This system bases its arbitration on a prediction of the ideal movement and its knowledge of the environment. The users interact with the system through a force/torque sensor. This input is then compared to a virtual force input based on both preplanned and actual trajectories. The system then designates a certain gain to each of the inputs to calculate the point of arbitration, through a so called adaptation law, between the computer and the user much like previously discussed.

3.5.1 Autonomous Car Levels

A recently developed scale discusses the different levels of autonomy in cars, as this field is rapidly evolving [46, 47]. This scale is used to define the amount of control the car has and what input is required from the users. These levels, defined in the scale, can be used as reference when designing shared control systems as they define the difference in the amount of control the human and computer have. It also gives an indication as to how the system should display the intent or control of the system

to not confuse the users as to what is happening. A short summary of the levels can be seen below:

- **Level 0: No Automation** The car has no automated assistance technologies. This does not account for cruise control, as it still requires a human to dictate every driving action.
- **Level 1: Driver Assistance** Some modern cars fall into this category as it requires the car to have one advanced driver assistance feature. Features such as adaptive cruise control or lane-keeping technology fit in this category.
- **Level 2: Partial Automation** Vehicles with two or more advanced driver assistance technologies that can control speed, braking, or steering of the car. These technologies are the same as Level 1 but as more are included they must know how to communicate with each other. It is important to note that this level still requires a human in the loop.
- **Level 3: Conditional Automation** A vehicle in this category is fully capable of operating during select parts of a journey when certain conditions are met. This could be the car is in full control during a freeway drive, but excludes on/off ramps and city driving.
- **Level 4: High Automation** These levels of automation are currently only hypothetical as no commercial products yet exist. At level 4 a vehicle is capable of completing an entire journey without the driver intervening. The limitations could for instance be that it is geographically locked or that its not capable of exceeding certain speeds.
- **Level 5: Full Automation** This is the ultimate level for self driving cars. A vehicle at this level is capable of all driverless actions in any given location. At this level, there is no need for human control, meaning the removal of steering wheel, pedals, and gear stick.

In the context of shared control and robotics, Dragan and Srinivasa argue in the quote in Section 3.4.1 that the system needs to be able to assume control to improve interactions, reminiscent of level 3 (Conditional Automation) which can take over in select situations. Counter to that, Yu et al. argue that shared control must only assist, not replace, users to improve performance. This is similar to level 2 (Partial

Automation) or level 1 (Driver Assistance) in most scenarios. It is important to note that all automation levels save for level 5 (Full Automation) require users to actively engage them, whereas many shared control systems use gradual transitions based on system confidence.

3.6 Evaluation Methods

Evaluating the arbitration of a shared control system is highly dependent on the context and the use of the system. Some studies perform tests using different arbitration methods and then collect subjective reviews of the different methods [45]. Yu et al. tested and collected subjective data from six participants. Furthermore, objective measures can be defined depending on the context, such as completion time, object avoidance, correct prediction of goals, etc.

Many evaluations are based on the accuracy of some algorithm's guess compared to the user's actions [48]. Other sources use physiological data to predict or validate their intent prediction, however caution the fact that some physiological data is highly varying in users [49]. Both methods require the users to complete a certain task and then have the facilitator confirm whether the system correctly predicted the user's intent or not.

Carlson and Demiris used speed as a measure of accuracy of their intent prediction. According to them, increasing the speed with which a user in a shared control system can perform a given task in a safe manner, would mean the intent was correctly predicted [50].

In Struijk et al.'s work they estimate the difference in performance of a robot controlled system by tongue- and keyboard controls and measure the time spent with each modality. Furthermore, they assesed the difference in completion time dependent on the robot's velocity [13]. While their results are varying, there are only objective measurements, and no subjective measures of how the different modalities function despite a difference in performance.

As the complications of tetraplegia are highly individual, it can be necessary to include personal experiences with the system in an evaluation. The success of a

shared control system, it seems, rests as much on the likeability of the system and the usability as it does on objectively improving the interaction times and performance.

Summary

Implementing a shared control system based on user intent prediction and computer vision seems possible to assist the end-users of the EXOTIC project. Computer vision allows the localization of objects in a given scene and allows the tracking of user's facial expression. Prior movement trajectories and other intent prediction methods have been evaluated and compared, and their benefits have been discussed. Additionally, previously tested evaluation methods have been discussed to ascertain which are viable and which are necessary to assess a shared control system.

4 | Final Problem Statement

Based on the limitations and opportunities of the EXOTIC project and its end-users, as well as the methods described in the background research, a specification of the problem is required to define the scope of this project. We suggest designing a system utilizing shared control and intent prediction to improve user satisfaction with activities of daily living, specifically reaching for and touching objects in a scene. We decide to impose this limitation to narrow the scope of the project, such that an evaluation is manageable and meaningful. To this end, we must design and implement key modalities, such as computer vision and physiological data, that can aid users without significantly reducing their autonomy, both to avoid scenarios wherein the computer makes erroneous predictions and performs a wrong action and to improve the user experience. This warrants a process that tests the influence of these modalities and the aggressiveness of the system's decisionmaking to improve both objective measures, such as time spent and adjustments needed, but also subjective measures, namely the system's likeability and usability, which is necessary for it to be implemented in the real world. Specifically, likeability in this context is defined as the impact of the system on the user's perception of the interactions, namely ease of maneuvering and feeling of assistance, and an arbitration level that is appropriate for a particular user. This can all be summed up in the following problem statement:

> *"How can shared control based on intent prediction be implemented in a tongue controlled personal robotic arm to facilitate and improve performance of activities of daily living by tetraplegics?"*

5 | Design

In this chapter we describe methods and possibilities to solve the problem statement. Firstly, an overview of the proposed design is presented followed by a use case visualized as a storyboard. The many parts of the project are then described individually, focusing on how to implement them in the system as a whole and what consequences, considerations, and opportunities they introduce. Finally, an implementation plan in the form of a Need/Want/Nice to Have-list serves as a way to structure the implementation of these parts, including by presenting the minimum implementation requirements.

5.1 Design Overview

This section broadly describes the components of the proposed system and their interactions. The subjects described in Figure 3.1 still apply and provide an overview of which aspects of EXOTIC this project investigates. Some of the areas marked in blue are still included in the project, but are not topics for research. For example, computer vision provides necessary features and introduces noise that the system should be able to handle but is not part of the evaluation.

The system should only be initialized once the user commences movement of the robot arm. First, the system should detect objects in the scene, preferably within reach of the arm. Once the objects are detected, the system should predict the intent of the user by comparing the physiological data and the trajectory of the user's movement with the known objects in the scene to estimate the user's intent. A probability measure is then calculated for each of the objects. As the probability measure

increases for a given object, the more confident the system can be in its prediction of which object the user is attempting to interact with. This confidence measure will then be used to arbitrate between user input and moving directly towards the goal. Note that the system does not attempt to calculate an optimal path as this is outside the scope of the project. The arbitration model has to be defined to ensure that the system improves the functionality of the system by decreasing completion time, smoothing the movement, and potentially increasing likeability of the system (or at least does not lower it).

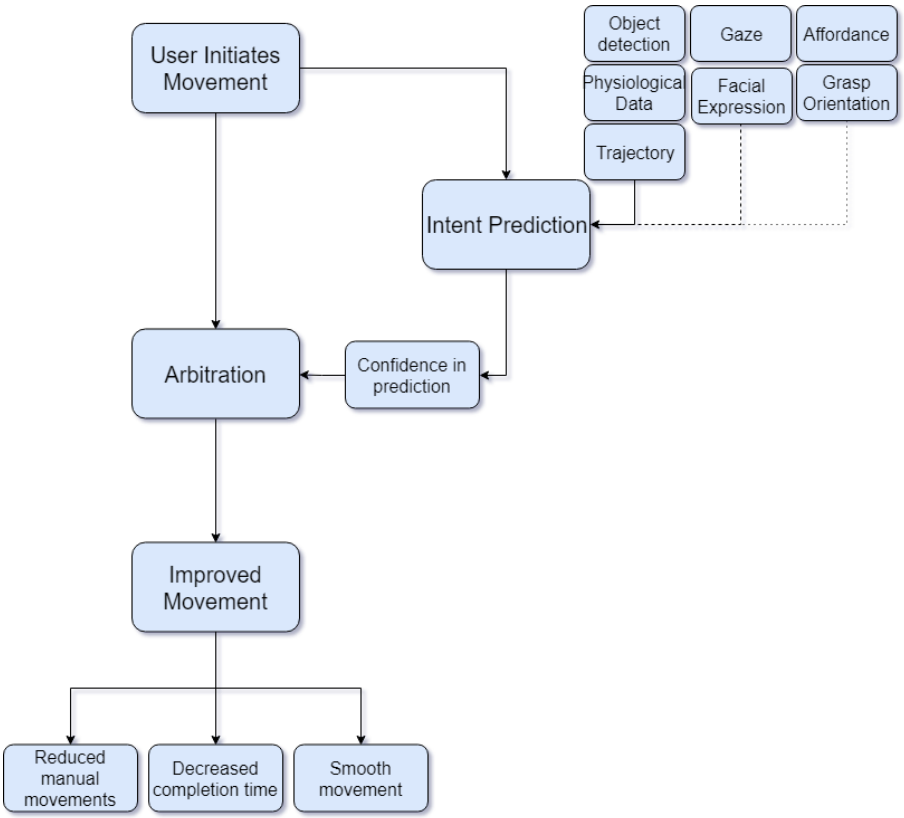


Figure 5.1: FlowChart of the proposed design

Figure 5.1 shows the proposed structure for this project. The different modules providing data to the intent prediction node are split into groups dependent on their priority, discussed later in the report. The first section describes the primary research aspects considered in this project and answer the questions posed about human-in-the-loop and how additional user data can improve prediction and likeability. The

second column requires additional equipment, specifically a user-facing camera setup, but can provide information about the user's state and additional prediction methods. The final column describes the optimal system; one which considers the objects in the scene and includes their affordance and grasp orientation in both the intent prediction and the resulting movements.

5.1.1 Storyboard

The storyboard describes a typical use-case for the proposed system from the initial preparation phase to the end goal, which in this case is reaching the target object with the robot arm.

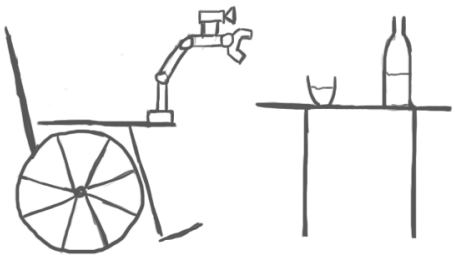


Figure 5.2: Initial proposed system setup

In Figure 5.2, a proposed setup for the design is shown. In this setup, the robot arm is attached to the wheelchair with a camera attached on top of the robot arm. The solution proposed in the EXOTIC project includes an exoskeletal arm which the user wears, but we disregard this in our project as it outside the scope. Equally of note is that the camera is positioned on the end effector - this is only one proposed position for it, and a bigger discussion is necessary to select the optimal position in this context.

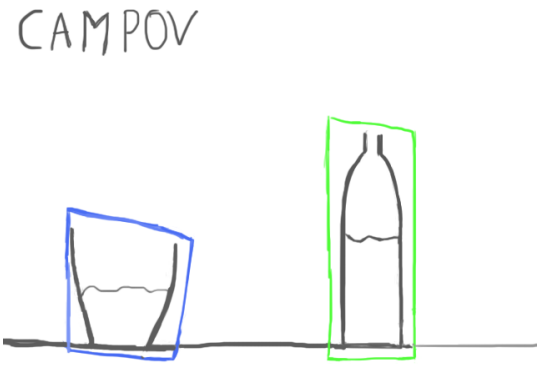


Figure 5.3: Camera's point of view

Figure 5.3 shows the point of view from the camera. This view allows the system to recognize the objects in the scene which could be potential goals for the user. The system should be able to distinguish between the table and each object on it.

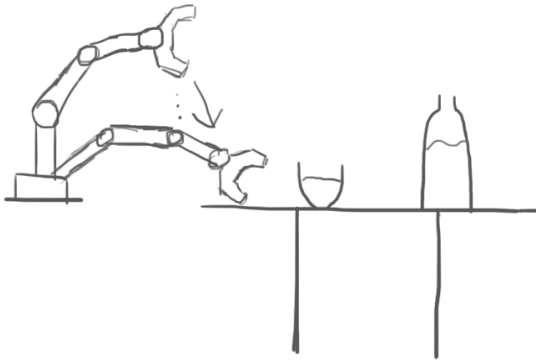


Figure 5.4: Moving the robot arm

Figure 5.4 illustrates the movement of the robot arm as initiated by the user. Once the user begins moving the robot arm, the system should detect the objects and use

the collected data to predict the user's intent based on trajectory, physiological data, and the objects in the scene. The measure should be continually updated and, based on certain parameters such as arbitration curve and aggressiveness, should change the amount of assistance the system provides.

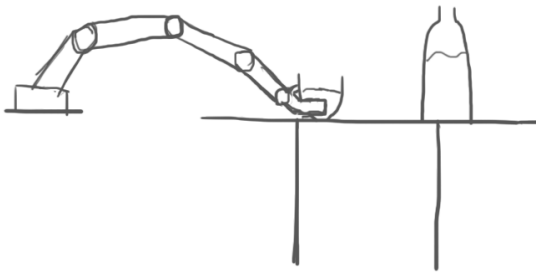


Figure 5.5: End of robot arm movement

In Figure 5.5 the robot arm has reached a goal, yet allowing the user to reach multiple goals in sequence if necessary. The movement should be smoother and faster than its unassisted equivalent. Note that the grasping aspect, that is both physically grasping and positioning for it, is outside the scope for this project and the system only considers the straight path to the predicted goal.

5.2 Modalities

The modalities for the system will be discussed in this section and are based on the previous discussed intent prediction methods in Section 3.4. This will include how they could be implemented individually and how they are going to be prioritized in the implementation phase.

The primary modalities are the modalities necessary for the system to function (*Needs*) and can be seen in the first row of modalities in Figure 5.1; Object detection,

Physiological Data, and Trajectories.

- **Object detection** should be able to detect all objects in the scene and give an estimate of their position in relation to the camera and robot arm.
- **Physiological Data** means that the system should incorporate a measure of physiological data, but it should aim to stay as uninvasive as possible. This means that using EEG based solutions is not going to be a possibility due to the invasive nature of this method. This opens the possibility of using GSR as a measure of arousal in the users. GSR sensors are easy to implement and are small and uninvasive.
- The **Trajectories** of the robot arm's movements should be used to estimate which direction the user is intending to move, thus allowing the system to calculate the most probable goal in the scene. This allows the system to assist the user in real-time based on their previous inputs.

The secondary modalities can be seen as a *Want* feature for the system and is shown in the second row of modalities in Figure 5.1; Gaze and Facial Expression.

- **Gaze** as an input modality could improve the intent prediction of the system by incorporating a prediction of the object the user is currently looking at.
- **Facial Expressions** of the users could be used as a measure of confidence in the current intent prediction, much like the proposed GSR.

The last modalities could be considered as *Nice-to-Have* features and can be seen in the final row of modalities in Figure 5.1; Affordance and Grasp Orientation.

- Automatically inferring **affordance** in the system could improve its intent prediction as it would be able to infer the sequence of events based on the objects in the scene, for example grasping the water bottle in order to pour into the cup and then picking up the cup.
- **Grasping** as a feature could further increase the usability of the system while reducing the amount of effort the users have to perform. If the system knew how to properly grasp certain objects, the only requirement of the users is to properly direct the robot arm in the direction of the correct object.

These modalities are sorted in such a way they resemble a Needs, Wants, and Nice-to-Have list, meaning that it is only a certainty that the Needs will be implemented in the final version and not necessarily the Wants and Nice-to-Haves. This list has been reformulated into an Implementation Plan shown in Section 5.7, which describes the order of implementation.

5.2.1 Object Detection

Object detection is not a major point of research in this project, but it does introduce noise into the system which it needs to be able to handle. Additionally, automatically detecting objects will reduce the amount of manual work in the evaluation. As such, we require a solution that can be relatively easily implemented, thereby freeing up resources to deal with more important aspects of the project, while still providing accurate object positions to the prediction algorithms.

Euclidean Cluster Extraction

Euclidean Cluster Extraction is a clustering method used in computer vision to divide a given image or image-stream into clusters based on the probability of the individual points belonging to a given cluster. It often utilizes Sample Consensus plane model segmentation methods which detect all the points in the point cloud that fit a plane model. This plane can then be subtracted before the Euclidean Cluster Extraction method, such that the plane, for instance a table, is removed from the point cloud thus isolating the objects on top of the table.

Camera & Camera Placement

In order to to do real-time object detection, we decide to use an RGB-D camera. One of the major considerations is the positioning of this camera. The position of the camera can have an influence on the objects detected, for example due to field of view, occlusion, and many other factors.

Camera Placement	Pros	Cons
Robot Arm (End-effector - Facing Objects)	Follows the user Confidence increases as distance decreases	Never full overview Minimum camera distance
Room (Facing Objects)	Full overview	Continuous recalibration Risk of occluded objects
Robot Arm (Base - Facing user)	Capture facial expressions Non-invasive eye gaze estimation	No object information
Robot Arm (Base - Facing objects)	Easy Calibrations Point of view from robot	Risk of robot occluding objects

Table 5.1: Pros and cons of different camera positions

Table 5.1 shows the pros and cons of possible camera placements. The primary drawbacks from the cameras mounting in the room is that they are static, however this is not an issue if a multi-camera setup is possible. The multi-camera setup would improve overall performance as much more information can be captured, but will be limited to that specific room and require calibrations between the cameras and the robot arm. As such, the final design will be based on having a camera mounted on the wheelchair as it is guaranteed to follow the user. Of these, the version mounted at the base has the sole drawback that it can potentially occlude itself, which is correctable by excluding clusters within a certain distance from the end effector position, while mounting the camera on the end effector introduces several unfixable issues, such as not being able to detect objects closer than a certain distance which means the final stretch of the movement cannot be assisted by the system. For that reason, we choose to mount the camera at the base of the robot arm.

5.2.2 Galvanic Skin Response

As the system includes the user in the loop, it is possible to include passive measures of their experience with the system. Specifically, the participants's GSR could be used as a measure of the user's engagement in the current task, specifically whether the user is either agitated or excited. In the case of agitation, the user's GSR value would increase, which could indicate that the current movements of the robot arm are not correct. Unfortunately, this is also the case with excitement, which means that this measure should account for false-positives. As such the GSR measure can not be used to predict the intent but rather as a module for validation of the intent

prediction. Additionally, it also means the measure can not increase the confidence - only reduce it.

The GSR sensors are small and non-invasive allowing for easy setup and allows the users to move freely while they are attached. The measurement from these sensors are reliable and show a direct link between cognitive load and GSR values both in healthy participants and SCI patients [51, 52]. As the GSR value is based on the physiological response to events occurring around the user, outside influence could potentially affect the measurements.

As there is a lot of variance in the GSR peaks, it is important that smaller peaks do not influence the system as much as the larger ones. In order to ensure that this is the case an exponentially decreasing function could be used. This also means the final value should be clamped from 0 to 1 to ensure we do not modulate with negative values (as the GSR keeps falling), and that the % change is only counted for negative change, so as to not reduce the confidence when the GSR rises.

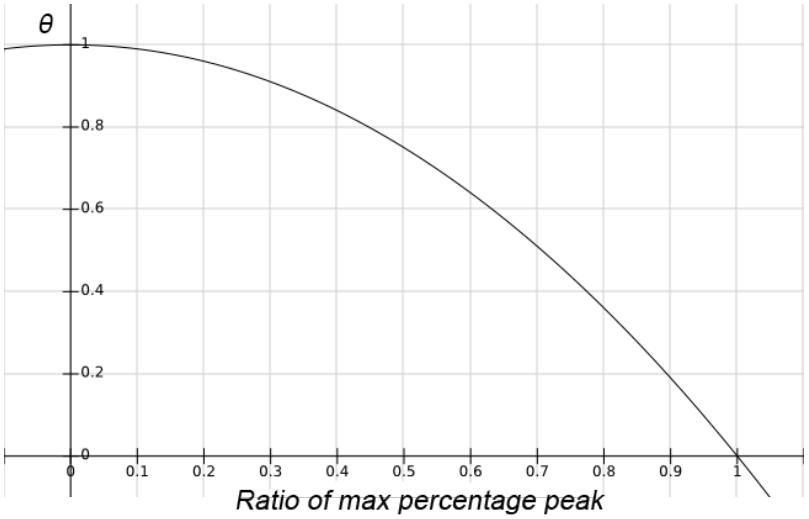


Figure 5.6: Proposed scaling of GSR modulation

Figure 5.6 shows the proposed method of modulation. The exponentially decreasing modulation value, labelled θ, ensures small changes have significantly less influence on the final confidence than larger peaks. However, in the case of %-max, the value is currently undefined. It will require an internal test with several users to see the

expected change in GSR over time whilst using the robot.

5.2.3 User Input Interface

The current state of the art input modality for tetraplegics is a tongue controlled palate hanger. This requires the user's tongue to be pierced in order to interact with the hanger. The design and layout of such a hanger is being researched in the EXOTIC project, but some commercial products already exist [53].

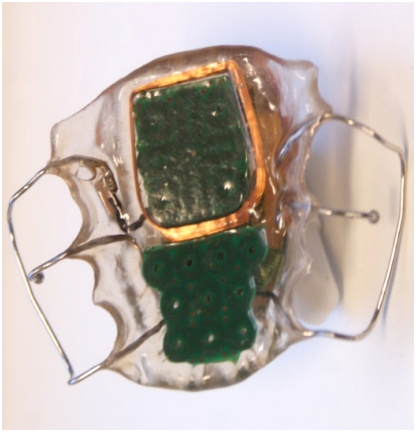


Figure 5.7: Itongue® palate hanger from TKS A/S used in the EXOTIC project. Picture source: [53]

Figure 5.7 shows the current design of the palate hanger. The buttons on the hanger have different functions mapped to each of the buttons, but are customizable, allowing users to define these functions.

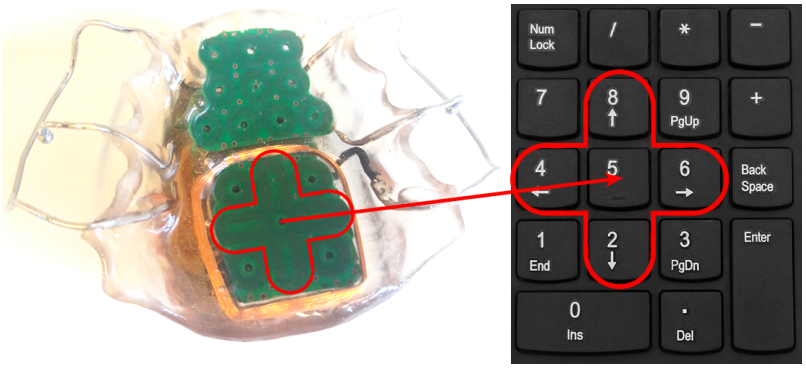


Figure 5.8: Comparison between the palate hanger and the layout of a numpad

This layout could for testing purposes also be implemented as a Numpad controlled system, as demonstrated in Figure 5.8. We decide to change the input modality as the palate hanger requires the user to have a tongue piercing as well as having the hanger fitted to their mouth. Using a similar control scheme allows for testing with able-bodied participants while simulating the original context as closely as possible. Another option could be using a joystick to mimic the control scheme of the palate hanger, but this would further complicate the control for up and down movement and has therefore been discarded. To increase the realism of the transferred control scheme, it should only allow for a single key press at any one time as is possible with the palate hanger.

5.3 Intent Prediction

Intent prediction is a core aspect of this project, as well as other scenarios where continuous collaboration with computers is a necessity, since the effectiveness of the collaboration is greatly increased by not having to explicitly mark goals.

Dragan and Srinivasa suggest that the most likely goal is one that best satisfies some cost function [9]. The most likely goal is given by:

$$G^* = \arg\max_{G \in \mathcal{G}} P(G|\xi_{S \to U}, \Theta) \tag{5.1}$$

According to them, the probability of a goal being targeted is given by:

$$\frac{exp\left(-C_G(\xi_{S \to U}) - C_G(\xi^*_{U \to G})\right)}{exp\left(-C_G(\xi^*_{U \to G})\right)} \tag{5.2}$$

Equation 5.2 indicates that probabilities decrease exponentially as their cost increase, where C_G indicates some cost function dependent on the goal and ξ indicates a trajectory. This can be true, to the extent that the outcome should be a probability distribution of all goals dependent on movement towards or away from it, however that is not Dragan and Srinivasa's outcome. Their cost function is later described as:

$$C = \sum_i ||\xi(i) - \xi(i-1)||^2 \tag{5.3}$$

Dragan and Srinivasa, in disregarding direction, rely on the ratio of total movement towards the goal and the remaining distance to the movement of the optimal path to the goal. Not only does this not yield a probability distribution, continuous movements also have equally large impacts on the result, meaning successive goals or long movements with smaller corrections are hard for the formula to consider. However, a defining feature of this method is its goal independence, in that it does not regard number of possible goals in the equation other than as a scaling factor as $P(G)$.

In trying to alleviate some of the issues of Dragan and Srinivasa's implementation, we develop a number of possible substitutions that better fit this particular use case. As the input is assumed to be fairly uniform at first, followed by a series of smaller corrections, we propose using multiple trajectories over different time intervals, each of which with a most likely goal and some confidence of that goal being the intended target, producing a probability distribution. The likeliness of the goal being the intended target is, in this iteration, linearly dependant on the dot product of the travelled trajectory and the optimal, straight path to the goal.

However, this implementation has a number of flaws. Firstly, the dot product would favour goals further away regardless of the distance travelled. This issue can be alleviated in a second version by normalizing both vectors, thus disregarding the distance travelled completely. Another issue is the voting system, wherein longer trajectories have equal voting power to shorter, more accurate ones. This can be an issue in the later phases where corrections between goals near each other happen, alleviated by reducing the relative voting power of longer trajectories.

Both of these methods have a major flaw, in that they are directly correlated to the number of possible goals in a system. The probability of a given goal is significantly reduced by the introduction of more potential candidates. One potential fix for this issue is assuming a linear-quadratic relationship, wherein a linear drop in dot product (the cost) equals a quadratic reduction in relative probability. The relative nature of the method also means that travelling towards a single goal and away from all others

would always yield a confidence of 1, which can be an issue, but would also require this to be the case for all trajectories.

In order to avoid feedback issues, wherein the system uses its own (adjusted) movements to predict where it is going, it is important to desynchronize the user's input from the actual robot movement. We propose that the logged trajectories be simulated based solely on user input, and using these trajectories in the calculations rather than the robot end effector's actual movements.

5.4 Confidence

Determining how the confidence is going to be calculated by the system is an important design choice, as this will be the deciding factor in the arbitration models. This measure is going to be based on tests, as there are no predefined methods to estimate confidence based on the chosen modalities.

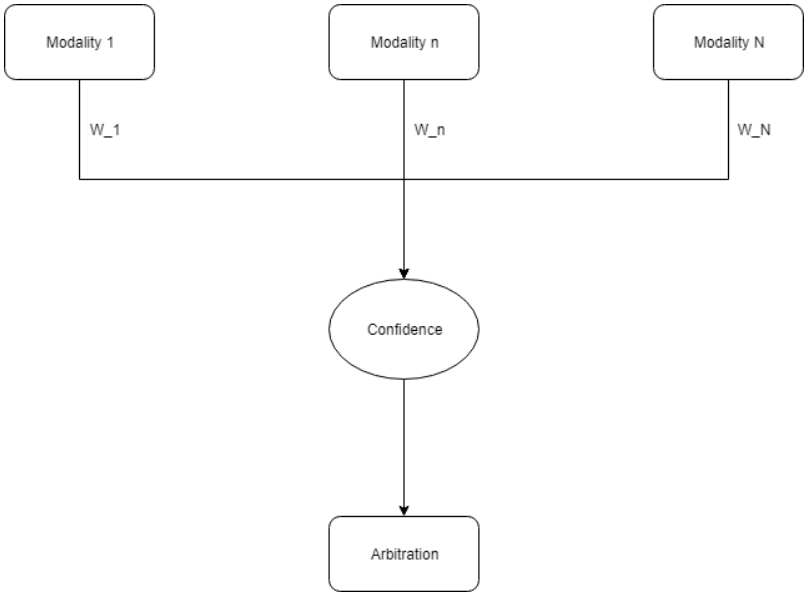


Figure 5.9: Influence of the different modalities on the confidence

Figure 5.9 shows how the confidence is calculated based on a setup with N different modalities. To ensure proper intent prediction the data from the different modalities

will have to be weighted differently. These weights should depend on the reliability of each modality as well as the certainty of data flow.

Since this is a novel solution to a specific problem, not much research has been published which relates to or is usable in this context. We assume that, having a user in the loop which enables it, multiple modalities can improve the likeability and efficiency of the system. However, we cannot assume that all modalities can accurately predict a target or that they have the same reliability. For example, we do not predict a goal based on GSR. Instead, we can use GSR to modulate the confidence of the intent prediction algorithm before it is converted to a blending value, α, through some arbitration curve.

5.5 Arbitration

Arbitration is the point of shared control which decides the ratio of human input to computer input. This process can either be done in a gradual manner, once the confidence reaches a certain threshold, or a mix of both. It is important to choose an arbitration curve which yields an α value that is specific and impactful in this context without reducing likeability, since all shared control systems are different.

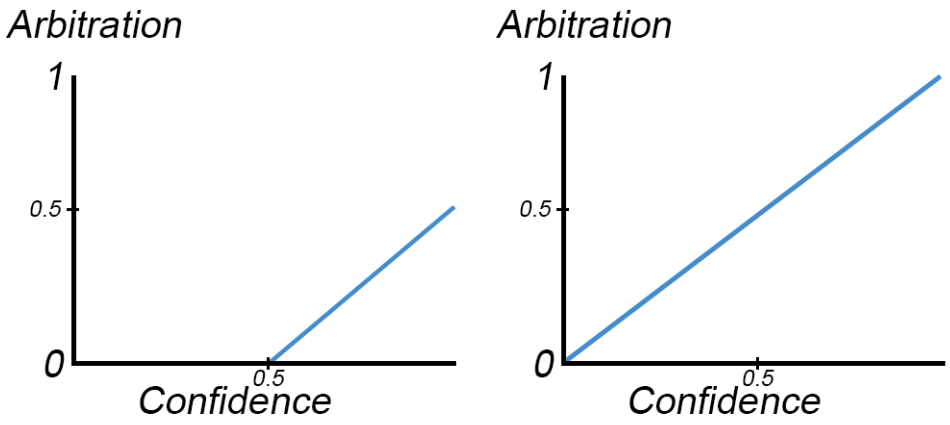


Figure 5.10: Timid and aggressive arbitration curves

We decide to test two arbitration curves; a timid one which never assumes more

control than the user, shown in Figure 5.10 (left), and an aggressive one which can assume full control of the robot arm, shown in Figure 5.10 (right). We decide on these two curves to test a number of things, but namely how a system which can exact itself more on the system than the user can influence the objective measures, such as time or adjustments. However, users might like the system less even if their performance increases since the robot arm can move in directions the user never intended. Furthermore, as the two curves are linear, users should experience a gradual change in arbitration levels, and as such the change should be recognizable and not overwhelming or confusing. These two arbitration curves showcase two "extreme" cases, but in real-world implementations the end users may need the ability to personally adjust these to suit their needs.

5.6 Measuring Results

Since we produce a system with a user in the loop, it is important to gather subjective data about their experiences with it. Additionally, as the system is developed to improve performance in reaching tasks, we need to be able to measure this objectively.

An important objective measure is the amount of adjusting movements required by the users in order to move the robot arm to a given position. Currently, many users have to perform a large number of adjustments in order to correctly move the robot arm in front of a given object to correctly grasp it. These movements could potentially be done faster and in fewer adjustments by the robot, thus saving the users from doing extra work. Reducing the amount of inputs from the users is important in order to facilitate prolonged use of tongue controlled robots [54]. Furthermore, decreasing the time spent on each individual task as well as increasing the physical self-sufficiency of the users is linked to a higher quality of life [55].

The subjective measures are primarily how the users feel about the system using shared control. It is important that the users are not disturbed by the movements of the robot and the system as a whole should have at least the same likeability as the control. Likeability is how the users feel about the system; the feeling of assistance or hindrance, and how easy or hard it is to achieve the goal. We decide that these measures are sufficient to determine the users' perception of the system

and their limited interaction possibilities, and that standardized usability tests are beyond the scope of this test. As this might be affected by the intuitiveness of the control scheme, this is also necessary to gauge.

5.7 Implementation Plan

This section aims to describe the Needs, Wants and Nice-to-Haves for the system. The Needs in this case also function as the minimum implementation requirements; the most important features that is required for the system to be functional. The list can be seen below:

Needs:

- The system must detect all possible objects in the scene and discard objects out of reach of the robot
- The user's intent should be estimated by the system based on object positions and the trajectory of the current movement
- The system must include a confidence rating that defines the arbitration level of the shared control system

Wants:

- The system should incorporate additional physiological data in the intent prediction, such as gaze or facial expressions

Nice-to-Haves:

- The system should rate the detected objects in the scene based on their affordance and the probability of the user wanting to interact with the object
- The system should be able to correctly grasp the detected objects based on their affordance and classification

Summary

An overview of the design has been created to define key components in order to ensure that the goal of the implementation and the purpose of the system is specified. The important modules required to create a shared control system, as well as their pros and cons, have been discussed. Furthermore, objective and subjective measures have been discussed for evaluation purposes. Based on this, a tiered implementation list has been created, outlining the Needs, Wants, and Nice-to-Haves of the system.

6 | Implementation

This chapter describes the implementation of the system as defined by the design. The tools used are described, and an overview of the code is presented. This is followed by a description of the individual nodes, how they are implemented, their functions, and possible considerations. Lastly, internal tests are performed on key functionalities to ensure they are working as intended and to determine parameters for this use case. A final test of the system as a whole is then conducted to ensure the system is functional and ready for evaluation.

6.1 Tools Used

The tools used are shown below:

- ROS
- Shimmer3 GSR+ Sensor
- Realsense D435
- UR3 robot arm

Robot Operating System or ROS, is a flexible framework utilized to write software for robots. It consists of a collection of tools, libraries, and conventions that all aim to simplify the implementation of complex robot software for a variety of robotic platforms. This allows for the implementation of high complexity tasks which vary dependent on environment and are difficult to develop single-handedly. ROS is currently only available on Linux, but a Windows 10 port is currently in experimental use.

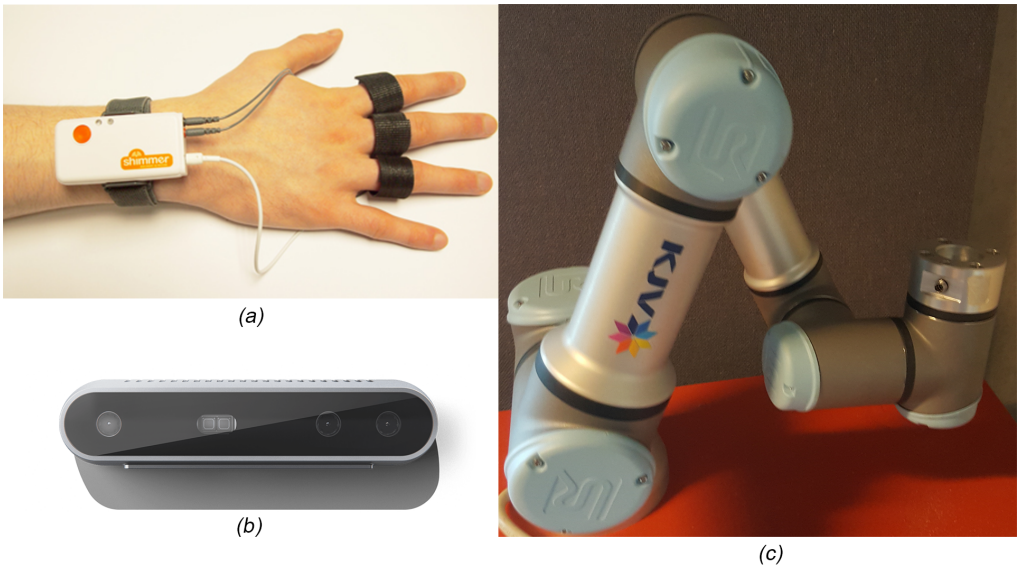


Figure 6.1: Shimmer3 GSR+ Sensor, Realsense D435, Universal Robot 3

Shimmer3 GSR+ is a commercial product capable of measuring the GSR of wearers as well as other physiological measures. The device is made to be small and uninvasive to the users and uses two small bands around the fingers of the user, see Figure 6.1 (a). It transmits data via Bluetooth.

Realsense D435 is a camera developed by Intel® that uses stereo vision and infrared to calculate the depth of a frame, see Figure 6.1 (b). It consists of a pair of depth sensors, RGB sensor, and infrared projector. Using the depth cameras limits the resolution to 1280x720p at 30 fps with a range of $0.3m - 10m$. The D435 unit has a horizontal field of view of 87°.

Universal Robot 3 or UR3 is a flexible and simple robotic arm, see Figure 6.1 (c). It allows for 360°rotations in all joints and unlimited rotation for the end effector. It has a reach of roughly $60cm$ and reproducible accuracy of $\pm 0.1mm$. It is capable of lifting up to $3kg$ and is used for many purposes in the manufacturing sector.

6.2 Code Overview

This section aims to give an understanding of the system and how its many nodes communicate with each other, as well as how they operate internally.

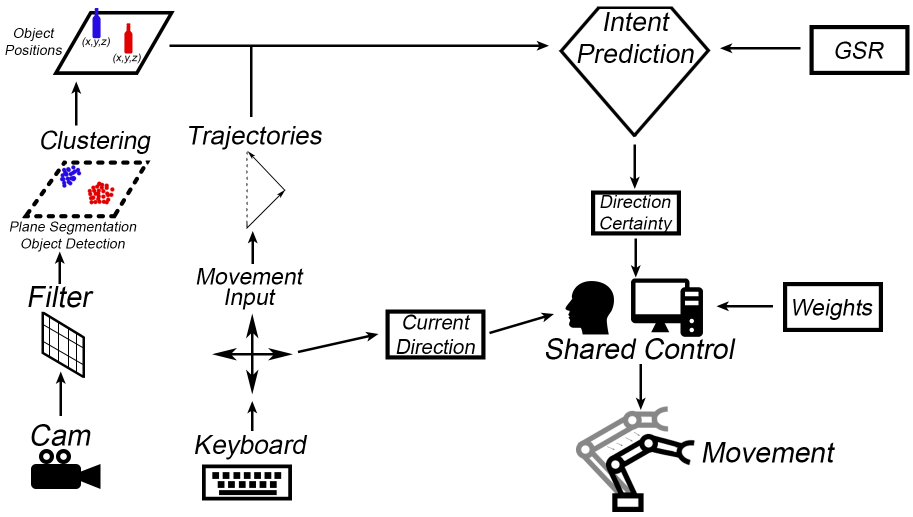


Figure 6.2: Pipeline for the nodes in the implementation

Figure 6.2 shows the pipeline for the implementation. As ROS currently only supports Linux the code has also been written in the same environment. Programming in ROS utilizes a form known as *nodes* which essentially are blocks of code describing a functionality. Therefore, this pipeline can also been seen as all the nodes that have to be created in order for the system to work.

Each block of code, or node, can *subscribe* to and *publish* information in data structures known as messages. Messages can range from simple primitives such as integers to complex structures of information such as pointclouds.

6.2.1 Camera Nodes

These nodes include everything in the diagram from "Cam" to "Object Positions". They serve the purpose of providing the system with information about the area in front of the camera, specifically which objects of interest are located in front of it.

The "Cam" node itself is published by the Intel Realsense ROS package, which can be made to publish a Pointcloud containing thousands of points, each with a position in 3D space and a colour in RGB format.

Filter

The distance filter filters out all points beyond $1m$. In order to utilize the PCL voxelgrid object, a 3-dimensional grid established using an input pointcloud to downsample and filter input data, the amount of points needs to be significantly reduced since PCL uses 32-bit indices to save memory.

- Subscribes: PointCloud (the camera stream from the D435)
- Publishes: PointCloud (filtered by distance, only leaving points closer than $1m$)

Whenever the filter node receives a pointcloud from the camera, it first transforms it to a PCL pointloud. The PCL library has a number of ROS-specific conversions to ease this process, but cannot otherwise work with ROS-type pointclouds.

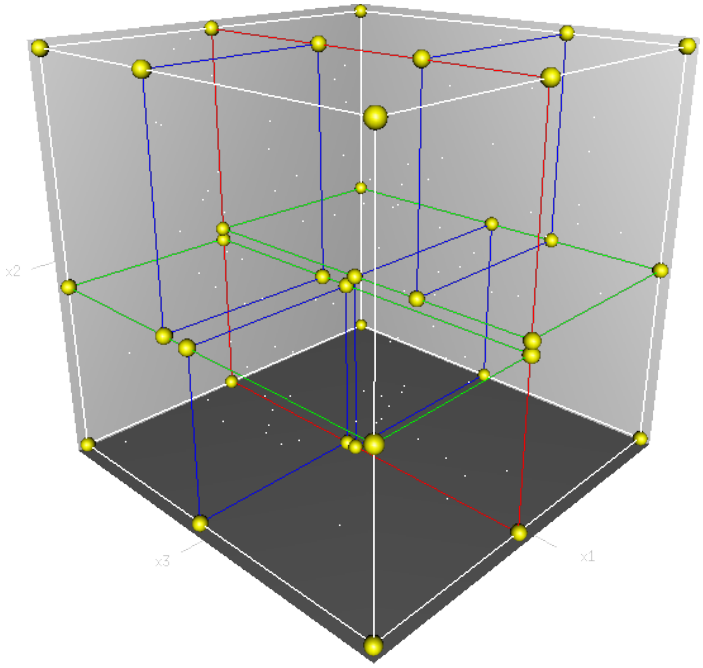


Figure 6.3: Visualization of a 3-dimensional tree. Source: [56]

In order to ensure the program does not find objects outside the robotic arm's reach, it initializes a K-D tree. This data structure is a k-dimensional tree that, at worst, enables searching in O(n) time. A visualization of a k-d tree with 3 dimensions is shown in Figure 6.3. This implementation uses FLANN (Fast Library for Approximate Nearest Neighbours) to efficiently estimate nearest neighbours. Once all points within $1m$ of (0,0,0) have been found, it pushes these to a new pointcloud, converts it back to ROS message so it is able to transmit through the ROS pipeline, and publishes it.

Clustering

This node segments the plane and clusters objects based on euclidean distance.

- Subscribes: PointCloud (distance filtered points)
- Publishes: PointCloud (clusters of objects)
- Publishes: Point Array (the middle of each cluster)

This node makes use of two computer vision techniques, namely plane model segmentation and euclidean cluster extraction. Firstly, it establishes a voxelgrid based on the input pointcloud, which downsamples a pointcloud to a grid and filters it.

The plane is separated from the rest of the points in the point cloud using Random Sample Consensus (RANSAC), an iterative method that estimates parameters of a model when the data contains outliers that should not influence the outcome. The probability of the resulting model being correct increases with additional iterations. It samples a number of points chosen randomly, and selects the model that best fit its sample. This is because a sample of inliers will usually be more linearly related than a sample of both inliers and outliers or consisting entirely of outliers. The algorithm continuously segments the RANSAC segmented points from the voxelgrid until it reaches a certain ratio. This can remove tables and walls underneath and behind the objects, but will remove parts of objects if not enough large surfaces are present.

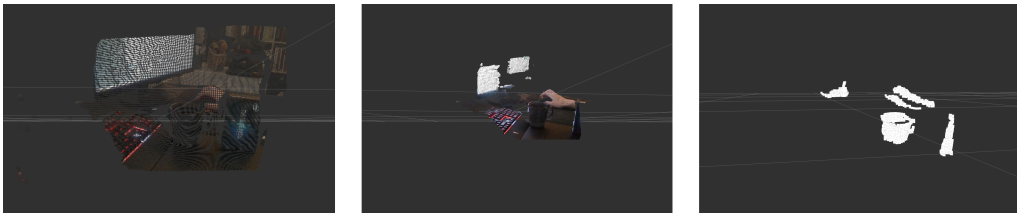


Figure 6.4: Left: Raw pointcloud data. Middle: Distance filtered pointcloud. Right: Clusters.

Having separated the plane from the objects in the cloud, another K-D tree is initialized as a search method. Now we make use of the PCL object EuclideanClusterExtraction which takes into account a cluster tolerance (in meters), as well as a minimum and maximum cluster size (in amount of points). This method now iterates through the remaining point cloud, pushing each cluster that fulfills all the requirements to an array. Both the final cloud containing all points in these clusters and the average positions of each cluster are published.

The pipeline can be seen below [57]:

1. A K-dimensional tree is created for the input point cloud dataset P
2. An empty list of clusters C is set up, and a queue of points that needs to be checked Q
3. For every point $p_i \in P$, the following steps i performed:
   (a) add p_i to the queue Q
   (b) For every point $p_i \in Q$:
      i. Search for the set ${P_k}^i$ of point neighbours of p_i in a sphere with radius $r < d_{th}$
      ii. For every neighbour ${p_i}^k \in {P_i}^k$ check if the point has already been processed and if not add it to Q
   (c) When the list of all points in Q has been processed, add Q to the list of clusters C, and reset Q to an empty list
4. Terminate the algorithm when all points $p_i \in P$ have been processed and are a part of the list of point clusters C.

To better illustrate the effects of these methods, a simple version is shown in the documentation with the results attached. These visualization can be seen below:

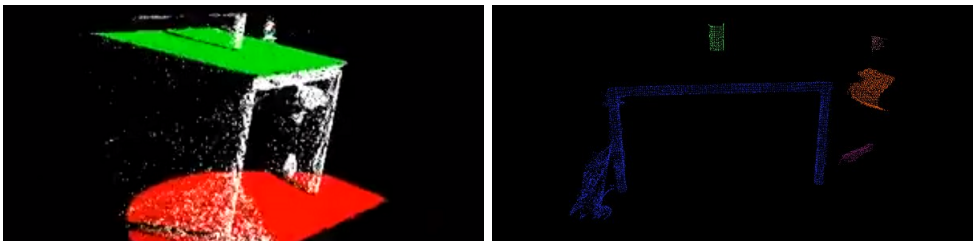


Figure 6.5: Left: Visualization of the plane segmentation, showing the floor and table as candidates for a plane model. Source: [58] Right: Visualization of the cluster extraction, showing the different clusters while removing the planes. Source: [57]

6.2.2 Galvanic Skin Response

This node measures the Galvanic Skin Response of the user.

- Publishes: Float (The confidence of the GSR, from 0-1)

To collect the GSR data of the user the Shimmer3 GSR+ device is used. This device requires a stable bluetooth connection to a PC in order to transmit the data. The package BlueZ is used for this purpose. The data received from the GSR device is stored in an 8 byte package, which is split as follows:

1 byte packet type + 3 byte timestamp + 2 byte GSR + 2 byte PPG

Photoplethysmogram (PPG) is used to detect changes in blood volume in microvascular beds of tissue and is not used in this project. The incoming GSR data is measured in kiloohms, and while conductance is measured in siemens, a conversion is not required as they are inversely correlated. The data stream is stored as GSR_{raw} as it is not converted based on the internal resistors in the GSR device. The current resistor is calculated by the upper two bits in the data stream, and is used to estimate the resistance in the skin of the users. The internal resistor (Rf) is determined as follows:

Data: GSR values
Result: Internal Resistor value

```
Range = ((GSR_raw » 14) & 0xff);
if Range == 0 then
    Rf = 40.2
else if Range == 1 then
    Rf = 287.0
else if Range == 2 then
    Rf = 1000.0
else if Range == 3 then
    Rf = 3300.0
end
```

Algorithm 1: Deciding internal resistor value based on upper two bits

The data stream of GSR_{ohm} is calculated by the internal resistor and the voltage it is given, and should give an indication of whether the user is aroused when using the system. These believed agitations should resemble small spikes in the measurements. As the users could get more or less agitated over time, it is important that a moving average is calculated such that sudden spikes can be noticed. Furthermore, as GSR values are individual, these spikes needs to be calculated based on their relative increase.

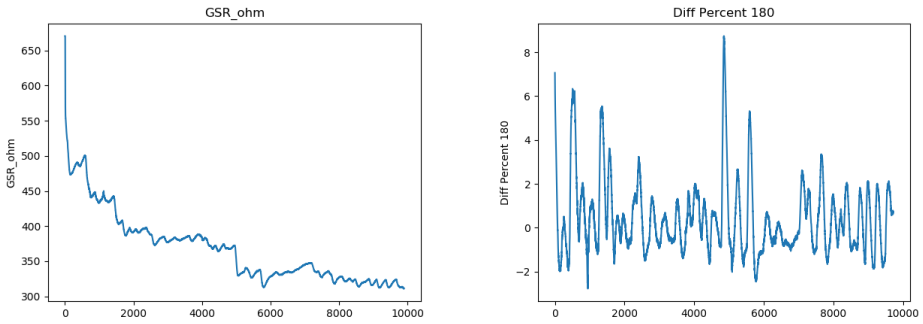


Figure 6.6: Left: test example of GSR values. Right: relative GSR peaks

Figure 6.6 (Left) shows the magnitude of the GSR and how much it varies for an individual. Figure 6.6 (Right) shows the magnitude of GSR peaks and are calculated based on the relative peaks compared to the moving average. These relative peaks are then estimated in percent based on the current moving average. If these peak percentages increase significantly we assume the participants is agitated and the current prediction is wrong or the movement is off. The threshold has to be defined in internal tests.

6.2.3 Keyboard Input Nodes

These nodes handle the user input from the keyboard and essentially make the robot move. They include everything from "Keyboard" to "Trajectories" and "Current Direction". User input is necessary for the robot to receive a direction and for the system to begin detecting inputs it can use to predict the user's intent.

Keyboard

Keyboards are a necessary input device to a system on which this program runs. As mentioned in the design chapter, the keyboard was chosen due to its similarity to the tongue input device. However, a keyboard does nothing in and of itself. It does not publish any relevant information to ROS, as it requires us to actively gather information about its current state.

Movement Input

This node acquires input from the keyboard and publishes it to ROS.

- Publishes: Vector3 (movement vector)

This node uses the Python library *keyboard* to hook the keyboard and acquire constant raw inputs from it. Since the keyboard is a "protected" input device on Linux, hooking it requires superuser privileges.

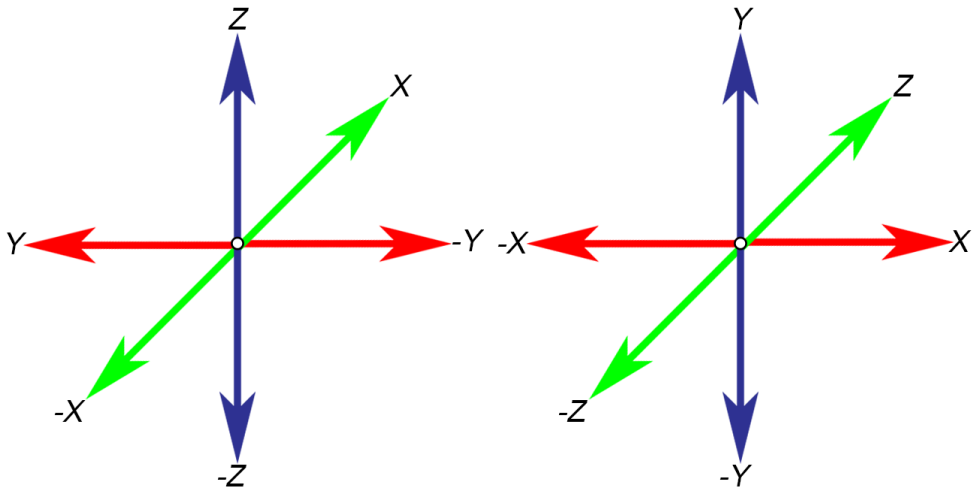


Figure 6.7: Left: Coordinate system used by UR. Right: Conventional coordinate system used by cameras

The script continuously spins at a rate of $60Hz$, acquiring the current keyboard state on each frame. It checks specifically for whether certain buttons are currently pressed. Using this input, it creates a 3-d vector. In order to interpolate the input and dampen movements of the robot, each input dimension is clamped to $\pm\frac{1}{30}$ from its value in the previous frame and limited to ± 1. This means that it takes half a second of holding a button to reach 1 in a single input dimension (30 frames of holding the button in a $60Hz$ system) and for a dimension to return to 0 from 1. It is important to note that the output vector format is not conventional so as to align with the robot's input dimensions. UR movement space and conventional space is shown in Figure 6.7.

Trajectories

This node creates a number of trajectories based on movement vectors.

- Subscribes: Vector3 (movement vector)
- Publishes: Vector Array (array of trajectories over different time intervals)

The node initializes a list of positions as long as the longest interval (in seconds, determined by internal tests) $\times 60$, since movement inputs are published at a rate

of $60Hz$. First, it remaps the input movement vector according to Figure 6.7 to align with the camera space (and for readability). It then prepends a new (current) position to the list which is equal to $input \times scaling + lastposition$. It is necessary for the scaling factor to be equal to the scaling factor used in the robot movement node later, as the movement needs to be accurate to world space. It then creates a number of vectors based on known times by simply subtracting the position at that time (in frames) from the current position.

Note that this node does not use the robot's actual movement so as to desynchronize the intended movement vectors from the actual robot movement. However, the node also does not have any information about the actual current position. As discussed in the design, this feature exists to ensure that only the user's movement input has an effect on the path so the robot cannot influence the intent prediction and thus creating a feedback loop.

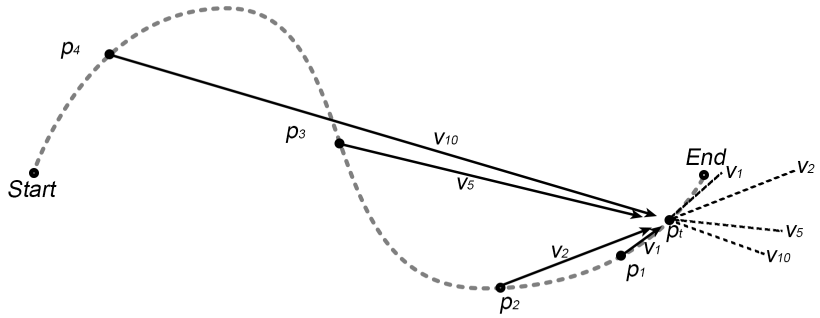


Figure 6.8: Difference in trajectories based on starting point

Figure 6.8 shows a number of different example trajectories and their continuation lines. Fixed time steps were chosen over distance as the robot moves at a near constant velocity, and it would allow the system to recognize breaks. Furthermore, users are expected to make longer unchanging movements the further the robot arm is from the target and making smaller, more precise movements the closer it gets.

6.2.4 Intent Prediction

This node predicts the intended target by bringing together the information from the previous nodes.

- Subscribes: Point Array (Robot positions at relevant times)
- Subscribes: Point Array (Positions of goals)
- Subscribes: Vector Array (User input trajectories of relevant intervals)
- Publishes: Point (The most likely goal's position)
- Publishes: Float (The confidence of the system, from 0-1)

Intent prediction is one of the core topics of this project and thus takes up a large portion of the work. This node estimates which object out of those published in the *clustering* node the user is most likely reaching for using the trajectories published by the *trajectories* node and the current robot end effector position published by *robot movement*.

Data: Found goal positions
Data: Distance threshold of goals being the same
Data: Time threshold to remove saved goals
Result: Array of potential goals

```
for Goal in Current List do
    for Input Goal in Input List do
        if Distance(Goal, Input Goal) < Distance Threshold then
            Goal Timer ← 0;
        end
    end
    if Goal Timer > Time Threshold then
        remove goal from list;
    else
        Goal Timer ← Goal Timer + 1;
    end
end
```

Algorithm 2: Checking goal renewal and removal

Whenever this node receives an array of goals from the *clustering* node, it checks each of them for similarities to known goal positions. If any are below a certain distance threshold, it is considered the same goal. This is necessary because the input is inherently noisy, and as such the goal position will likely not be exactly the same as it was in the previous frame. It also means that any point further than this distance threshold is considered a separate goal, and thus is calculated separately further in the node. All known goals are also evaluated for time known, and any that have not been found again for an amount of time equal to the time threshold are discarded from the list of known goals. This process can be seen in Algorithm 2.

Data: List of goals
Data: List of end effector positions
Data: List of trajectories
Result: Array of goals and confidences as long as the number of trajectories

```
1  Guesses ← Matrix of length Trajectories and height 2;
2  for Trajectory in Trajectories and Position in End Effector Positions do
3      Probabilities ← Array of length Goals;
4      for Goal in Goals do
5          Optimal Path ← normalized vector (Goal,Position);
6          Taken Path ← normalized vector (Trajectory);
7          Probabilities[Goal] = Max(0, Optimal Path · Taken Path);
8      end
9      Total ← Sum(Probabilities);
10     Probabilities ← Probabilities / Total;
11     Guess ← Argmax(Probabilities);
12     Confidence ← Max(Probabilities) - Max(Probabilities \Guess);
13     Guesses[Trajectory] ← Guess, Confidence;
14 end
```

Algorithm 3: Creating a list of guesses and confidences, one for each trajectory considered

The main algorithm runs whenever an input is gathered from the *trajectories* node in the form of a Vector Array. This indicates that there has been user input. In order to avoid race conditions, first the node makes a *deep copy* of both the current goals list and robot end effector positions list, which copies bit for bit the information in these arrays to create new arrays. In some scenarios, these might have updated whilst this section was running and thus completely altered the order and outcome.

Second, the program starts iterating through each trajectory. After checking all the data necessary is present, it initializes an array of probabilities as long as the number of goals. These will, for each trajectory, store the relative probability of this trajectory being the result of intending to reach that goal.

Then the program iterates through each goal, calculating the optimal path from the trajectory's starting position. It normalizes both this optimal path and the travelled path and, after checking that none have a length of 0, inserts the dot product of the two vectors into the array of probabilities. After each dot product has been calculated, it normalizes the array. The result is a probability distribution linearly dependant on the dot product.

The best guess for this trajectory is the maximum probability in the distribution. The confidence is calculated as Dragan and Srinivasa, by subtracting the second best guess from the best guess. The result is a number from 0 to 1, where 1 indicates complete confidence. The entire process, disregarding safety checks, is shown in Algorithm 3.

Data: Matrix containing Guesses (Goals) and Confidences
Data: Known Weights for each considered Trajectory
Result: A single Goal and Confidence
1 **for** *Confidence in Confidences and Weight in Weights* **do**
2 | Confidence ← Confidence × Weight;
3 **end**
4 Total ← Sum(Confidences);
5 Confidences ← $\frac{Confidences}{Total}$;
6 Votes ← Array with length Goals;
7 **for** *Guess in Guesses and Confidence in Confidences* **do**
8 | Votes[Guess] ← Votes[Guess] + Confidence;
9 **end**
10 Final Goal ← Argmax(Votes);
11 Final Confidence ← Max(Votes) - Max(Votes \ Final Goal);

Algorithm 4: Weighted majority voting based on previously calculated confidences and known weights

Now the program knows a number of goals and confidences in those goals. However, as discussed in the design it would be wrong to assume that each of these are equally important. This final part weighs the confidence based on weights from the internal tests and normalizes the resulting confidences. It then simply performs a majority

vote with the new weighted confidences. The final goal output from the node is the top voted goal, and the confidence is again the second highest confidence subtracted from the maximum confidence. This process is shown in Algorithm 4.

6.2.5 Arbitration

This node collects the confidences of the different modules and combines them to calculate arbitration levels:

- Subscribes: Float (The confidence of the GSR, from 0-1)
- Subscribes: Float (The confidence of the intent prediction, from 0-1)
- Publishes: Float (Arbitration level, from 0-1)

The confidence of the system is based on the previously discussed intent prediction and is then combined with the GSR confidence in this node. The GSR confidence in itself is not able to give a confidence in whether a certain goal is the target or not, but rather work as a variable confirmation of whether the current prediction is correct. This means that the confidence from the intent prediction is multiplied by the confidence from the GSR resulting in an overall confidence ranging from 0-1 on the current target. This confidence is used as arbitration level and determines how much the system controls the robot movements compared to the user.

6.2.6 Robot Movement

This node controls the robot arm, sending commands to make it move according to both user inputs and internally calculated values:

- Subscribes: Vector3 (The user input)
- Subscribes: Goal (Position in 3D space and arbitration level)
- Publishes: Point (Current robot position)
- Publishes: Point Array (Robot positions at relevant times)
- Publishes: Vector3 (Input to the robot telling it how to move relative to the end effector)

This node utilizes the MoveIt package to calculate trajectories for the robot to follow. It starts by setting up necessary MoveIt objects, such as the RobotCommander and a scene such that the inverse kinematics can be calculated in a virtual space in real time. After this is complete and a variety of parameters, such as Timeout and Maximum Attempts, have been set, the node spins, waiting for user input.

Whenever user input, in the form of a vector with 3 elements corresponding, is received, the node first determines if it has recently received a Goal with a position and arbitration level from the Arbitration node. It also publishes the robot's current position for use in the other calculations, as well as an array of previous positions that correspond to the start positions of each trajectory. If the user input has a very small magnitude (< 0.05) the callback stops as to not cause stutter or run when not needed.

If a Goal has recently been sent to the node, it creates a vector directly from the current position to the position of the goal by converting the camera-space position to end effector space. It then determines the magnitude of the keyboard input and normalizes this optimal path vector to the same magnitude as to reduce stuttering. It then adds the two together by a scaling factor of α, as such:

$$M = H * (1 - \alpha) + C * \alpha$$

Where M represents the final movement performed in this frame, H the user input vector, and C the optimal path. α is derived from the Goal sent from the Arbitration node. Once this direction has been determined, the inverse kinematics service is used to create an IK command to send to the robot. This is done by creating a ROS message called JointTrajectory and add the response from the IK service as the position, publishing it to a topic the robot listens to when it is hooked by the computer.

6.3 Internal Tests

This section describes the internal tests performed after the implementation of a node to estimate parameters for it or its performance. Each test will first describe

the parameters tested, then the method of testing, and finally the results so that the nodes each should be replicable.

6.3.1 Filtering

Distance filtering is necessary for a number of reasons, but has mainly been implemented because it excludes objects further away than the arm can reach, and improves performance with PCL, which uses 32-bit indices to save memory and thus cannot compute all the points in the cloud without a significant reduction in resolution. The outcome of this test was a filtering method and a time threshold for the *intent prediction* node.

Distance filtering was implemented three times; a naive method iterating through each point and calculating its distance to the center point, another creating a bounding box around the center, and a k-d tree looking for points within the distance. The two first methods also had multiprocessed versions attempted. The methods were tested for performance, and the distance threshold was set at $1m$.

Method	Min Time (s)	Max Time (s)	Avg Time (s)
Naive Distance	0.312	0.396	0.347
Naive Bounding Box	0.303	0.359	0.322
Multiprocessed Distance	0.809	1.399	0.931
Multiprocessed Bounding Box	0.685	1.416	0.927
K-d Tree	0.252	0.338	0.297

Table 6.1: Results from the performance test of the distance filtering algorithms

A few objects were placed in front of the camera, and the script was set to run for a few minutes. It printed the time taken each frame, and the minimum and maximum times were recorded. The results are shown in Table 6.1. The results indicate that multiprocessing has a significant overhead not negated by the increased processing power with such simple calculations. Furthermore, calculating euclidean distance rather than simply checking each coordinate did not impact the processing time significantly. This likely means that the sheer number of points to check, and iterating through them, is the significant factor. Based on the results, we implement the K-d

tree method in the distance filtering node. With roughly $300ms$ compute time, a point can exist in the *intent prediction* node for 4 frames before being discarded as not being found again.

6.3.2 Clustering

A number of different parameters are needed in the *clustering* node, such as RANSAC iterations and distance thresholds. Seeing as these are all context dependant, a potential test scenario was set up. For this test, a number of objects were placed in front of the camera, and parameters changed until:

- Table cloud segmented
- Objects were visible in the cloud
- The amount of objects found was equal to the amount of physical objects on the table

The parameters were changed until the above conditions were met. The RANSAC parameters were first, as it was obvious when the table was consistently removed in each frame. Since the input came each $300ms$, there was plenty of time to compute the data from the frame and the parameters could be increased to match. The number of iterations run by the plane segmentation was set at 10000, and the distance threshold was set at $1cm$.

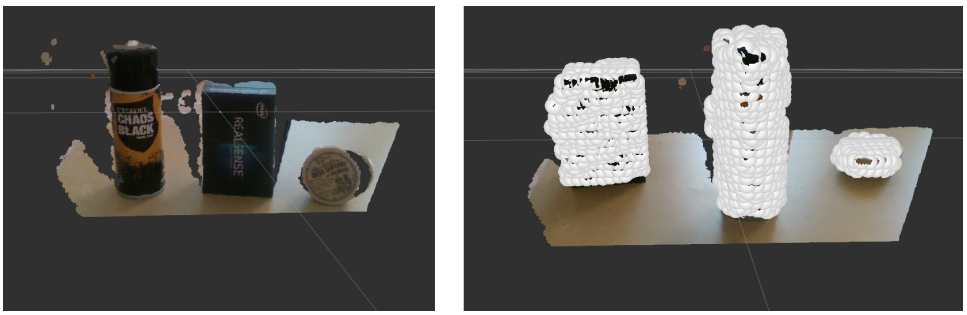


Figure 6.9: Left: Basic setup. Right: Clusters found in the basic setup

The three objects seen in Figure 6.9 (left) were used in the test to simulate different sizes of objects. By moving them around and changing the parameters until each

of them were discovered, the parameters were estimated in a simple qualitative test. The resulting clusters are shown in Figure 6.9 (right).

Parameters	Value
Remaining Voxels Ratio	0.6
Cluster tolerance	$2cm$
Minimum points	60
Maximum points	50000

Table 6.2: Parameters estimated by qualitative analysis

The clustering parameters, described in Section 6.2.1, are shown in Table 6.2.

6.3.3 GSR Peak Tests

As discussed in Section 6.2.2 we are interested in the %-peak increase in GSR values of the participants to estimate their arousal. However, as this percentage increase varies between participants, it is important to sample from multiple users such that a global threshold can be set for the system. A simple test setup was created, with a total of three participants, to measure their GSR response to a simple maneuver test with the robot arm.

Participant	GSR-range	Amount of Noticeable Peaks	Average %Peak	Highest %Peak
1	520-1100	6	7.2	13
2	220-420	25	4.8	10
3	320-660	10	4.5	8.5

Table 6.3: Results from the GSR percentage-peak test

The value for GSR ranges, amount of noticeable peaks, and highest %Peak differs from participant to participant, as shown in Table 6.3. As we want to estimate a proper baseline %peak value for the system, it could prove troublesome that some participants have only a few high peaks and multiple smaller peaks. In this case, if the baseline is set too high, some participants might never reach the maximum, and

thus GSR modulation will have a significantly reduced impact on their arbitration levels, whereas others might influence it too much.

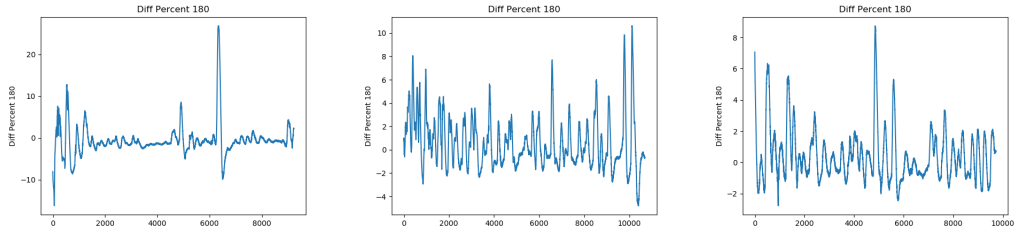


Figure 6.10: The different %peaks of the test participants

Figure 6.10 shows the %peak difference over time in the test. This %peak is calculated based on a moving average based on the last six seconds of values. We calculate the moving average over six seconds, as that is an average length of a GSR peak including onset delay and rise. The graphs show a noticeable difference in both amount of peaks and magnitude. One of these peaks in Figure 6.10 (left) was disregarded as it was the response to a loud noise from the room. Based on the tests, the base %peak value chosen for the system is 8%. This means that if the GSR %peak is larger than or equal to 8% percent the GSR confidence used in the arbitration level modulation is equal to 0.

6.3.4 Trajectory Durations

In the *trajectories* node, a number of trajectories are considered in order to make the final prediction. As previously discussed, using trajectories based on travel time rather than travelled distance appears to have some benefits such as being able to automatically reset by not moving. This internal test will result in a number of trajectory times, and weights for each of them, for use in the final majority vote.

Since the method used to predict the goal is novel, no prior research into such timings exists. We propose a test scenario wherein the robot arm is continually moved between objects placed around it, timing the travel times. This way, we can see the times taken in a potential scenario to reach objects with the robot arm. Of course, any change in robot arm length, movement velocity, and similar parameters will

require a new test.

21 positions were marked on the table in a semicircle around the robot in 7 rows, and a number of movements were performed based on randomly generated positions. Since certain positions were too close, randomly generated positions immediately next to the end effector were discarded. The times were logged and visualized in a box plot.

We performed two such tests; one wherein the controller was able to move along several axes at once, and one that more closely matched the input from the tongue controller in that the robot could only move along a single axis at one time.

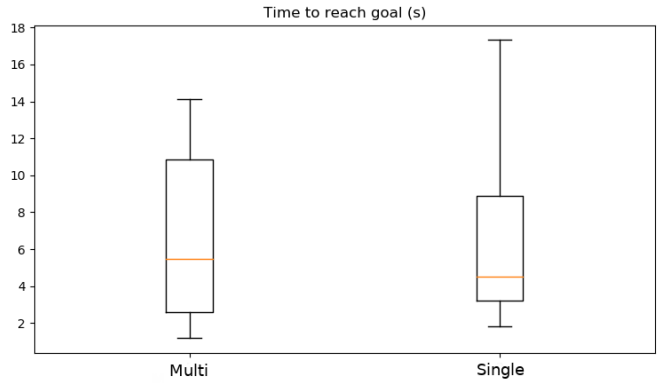


Figure 6.11: Results from the timing tests. Left: Multiple axes allowed. Right: Single axis movement

The results from the test are shown in Figure 6.11. We decide that five trajectories is a decent amount to consider and select both whiskers (minimum and maximum), both quartiles (lower and upper) and the median as trajectory times.

Trajectory	Multi Time (s)	Single Time (s)
Minimum	1.18	1.82
Lower Quartile	2.59	3.24
Median	5.46	4.49
Upper Quartile	10.84	8.90
Maximum	14.13	17.32

Table 6.4: Trajectory times

In using the version of the program where only one axis of movement is allowed at one time, the times shown in Table 6.4 (Single) have been used to establish these trajectory durations. These results show that most movement times have been more closely distributed in the single axis movements. It also shows that it is slightly slower with only one axis of movement, evident in both minimum and maximum being slower than their multi-axis counterpart, which was also expected. The mean difference from multi-axis to single was -0.314.

Another major part of the test was to establish weights for each trajectory for weighted majority voting. One such way would be to weigh trajectories based on their difference from the median time, which would make sense save for the fact that closer objects, and thus smaller, more different movements, would still be weighed less, and thus introduce a significant error to the system. We decide not to place any weight on the significance of the median and instead weigh trajectories based on their duration relative to the minimum, the smallest and thus expectedly most precise trajectory considered by the system. The weights can then be calculated as follows:

$$w = \frac{d_{\min}}{d} \tag{6.1}$$

Where w is the total weight, $d_{\min}$ is the time of the minimum trajectory, and d is the duration of the trajectory. This would mean that the minimum trajectory's weight is 1, but the weights are adjusted based on the sum of all weights following the above calculation.

Trajectory	Multi Weights	Single Weights
Minimum	0.53	0.44
Lower Quartile	0.25	0.25
Median	0.12	0.18
Upper Quartile	0.06	0.09
Maximum	0.04	0.05

Table 6.5: Weights calculated by their duration relative to the minimum

The relative weights are shown in Table 6.5. One significant difference is that in the single-axis version, no trajectory is weighted higher than 0.5, which means that in the event that each trajectory has equal confidence no single weight has majority voting power. Though it is an edge-case scenario, this could matter in the event that a slight corrective movement has been made towards a single goal whilst all other trajectories point towards another.

Intent Prediction Model

We decided to perform a low fidelity test of the different proposed intent prediction models in order to discover the differences, determine which flaws exist in each, and select the best method for this purpose. In order to do this, we create a number of pseudorandomly generated points on a 2D plane, intending to use these as the travel path. We also generate two random goals to calculate the probability of the movement being a result of an intent to reach either of them.

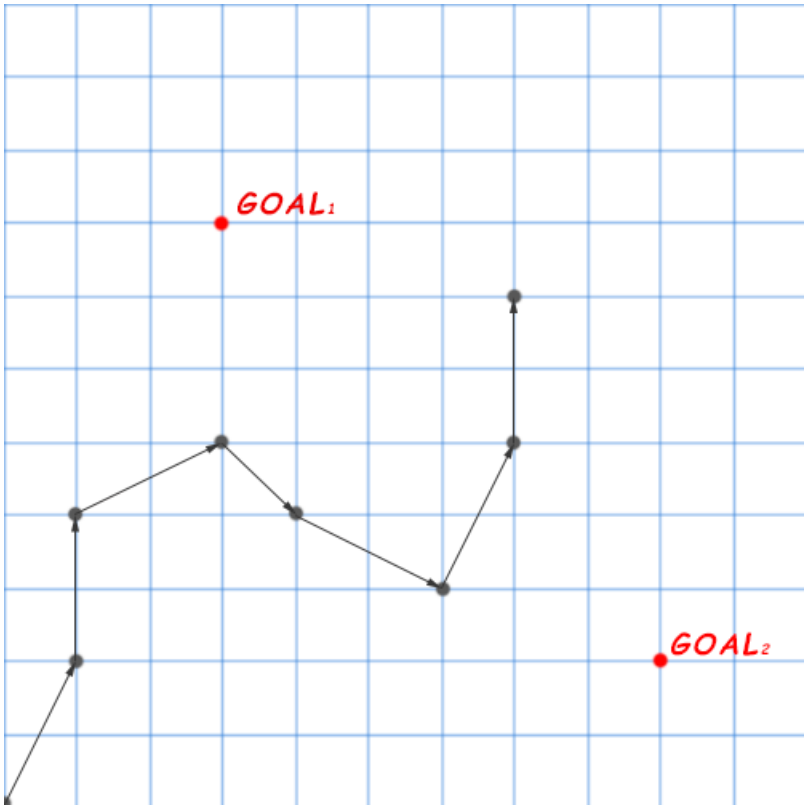


Figure 6.12: Test scenario of 2-dimensional pseudorandom movement

The path and two goals are shown in Figure 6.12. A script was created which, for each timestep, calculates the probabilities for goals 1 and 2 for each method. Note that this test does not use the trajectory times discovered in the time test, but does use the weights. This, coupled with the fact that it is a hypothetical scenario in 2 dimensions, means that the test is not entirely indicative of real-world performance. The methods are described in Section 5.3. With just two goals, the predicted goal is noted as $\arg\max(G_1, G_2)$ and the confidence of the model is calculated as $\max(G_1, G_2) - \min(G_1, G_2)$. This means that, in order to improve readability, we can simply display the probability of G_1 being the targeted goal.

Step	Naive	Normalized	Linear-Quadratic
1	1	1	1
2	1	1	1
3	0.679	1	1
4	0	0.01	0.017
5	0	0	0
6	0.226	0.385	0.447
7	0.721	0.982	0.966

Table 6.6: Probability of G_1 as calculated by different models

Table 6.6 shows the results from the test. Step 5 is the first test wherein all 5 trajectories are calculated differently, since the calculations take into account 5 steps backwards, otherwise defaulting to the first entry (in this case 0,0). However, this table indicates that a model that merely passes 5 confidences to a (weighted) majority vote, then normalizes them, is prone to return high confidences. This is because, in the case all 5 guesses are the same goal, even with minimal confidence, the result is always 1. In order to alleviate this, we develop a method which passes probabilities to a final vote. This reduces the confidence calculations to single one at the very end of the calculations. It also ensures that the probabilities, and thus the different confidences, are continuously considered.

Step	Naive	Normalized	Linear-Quadratic
1	0.594	0.612	0.713
2	0.738	0.825	0.905
3	0.530	0.567	0.63
4	0.326	0.237	0.202
5	0.164	0.088	0.048
6	0.318	0.424	0.455
7	0.623	0.848	0.853

Table 6.7: Probability of G_1 as calculated by newly modified models

Immediately, the probabilities displayed in Table 6.7 seem more realistic than those in Table 6.6. Not once does any of the new methods predict a goal with 100% or 0%

confidence. In this case, this indicates that the models are more accurate and better suited to guide a robotic arm. Of these, the naive implementation differentiates least between points, thus yielding the smallest confidences. The linear-quadratic version slightly increases the confidence score of whichever point it favours as predicted. In this scenario, the distinction is not particularly necessary but it also does not alter the results significantly.

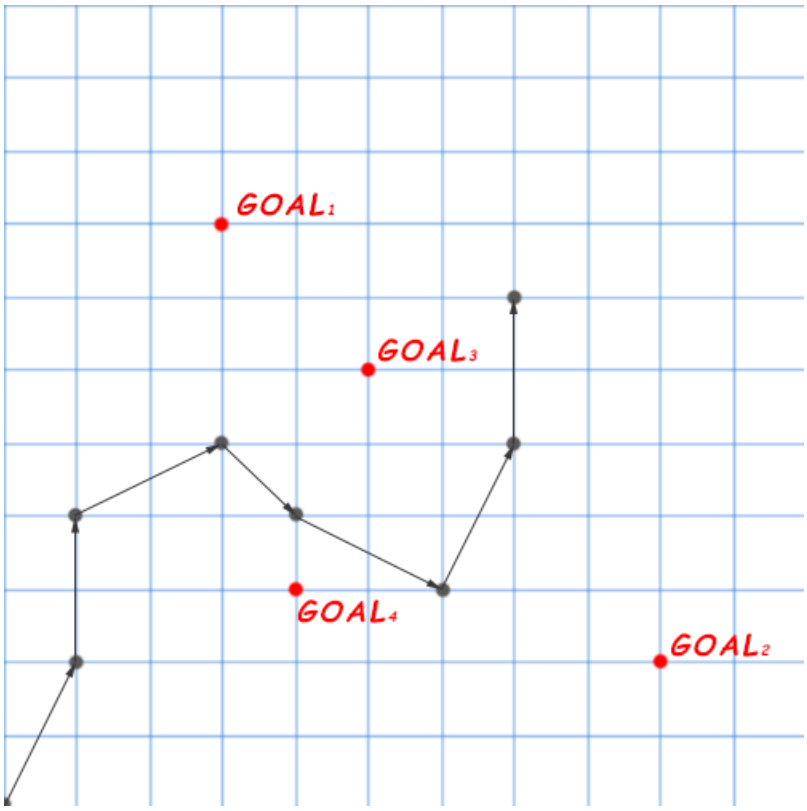


Figure 6.13: Second test scenario, multiple goals

In general, both the normalized version and the linear-quadratic version yield the most optimal results for this situation with very little difference between the two. However, the linear-quadratic version was proposed as a solution to situations where many goals would reduce the probabilities of each goal, and thus reduce the confidence so much that the system could not assist properly. In order to test this hypothesis, we introduce two additional goals, G_3 and G_4 shown in Figure 6.13.

	Naive		Normalized		Linear-Quadratic	
Step	Pred	Conf	Pred	Conf	Pred	Conf
1	G_1	0.034	G_1	0.006	G_1	0.013
2	G_1	0.086	G_1	0.07	G_1	0.154
3	G_1	0.01	G_3	0.028	G_3	0.061
4	G_2	0.181	G_4	0.01	G_4	0.02
5	G_2	0.404	G_2	0.179	G_2	0.318
6	G_2	0.167	G_3	0.171	G_3	0.244
7	G_1	0.109	G_3	0.027	G_3	0.057

Table 6.8: Predictions and confidences of models with four potential goals

The results of the test with four goals are shown in Table 6.8. As expected, the naive version consistently guesses the goals furthest away, which means it does not ever guess goal 3 or 4. The normalized and linear-quadratic methods consistently guess the same goal but with relatively large differences in confidence. Still, though, the confidence scores calculated by the linear-quadratic version are even lower than expected with only one exceeding the "random guess" probability of 0.25.

Step 5

Method	2 Goals	3 Goals	4 Goals
Norm	0.824	0.503	0.178
Quad	0.905	0.568	0.318

Table 6.9: Confidences of Normalized and Linear-Quadratic methods at time step 5 with goals 1 and 2, goals 1, 2, and 3, and all 4 goals

As expected, there is a major difference in the confidences produced by the system depending on the number of goals. The difference is shown in Table 6.9 which shows confidences calculated by the two still relevant methods with 2, 3, and 4 potential goals in the first time step with all different estimates. The largest relative difference can be seen when the number of potential goals increases to four, yielding a significant 0.14 difference in confidence which constitutes almost a doubling from the normal probability-dependant method. For this reason, we select the linear-quadratic version to calculate confidence in our solution.

6.3.5 Target Location Test

As the objective of the tests is for the user to reach different goals in the scene it is important to determine whether the positions of these goals influence the time spent in each task. The test was performed with three participants with each having to reach target 1, 2, and 3, in that order, a total of 10 times. The location of the targets were randomized between each participant.

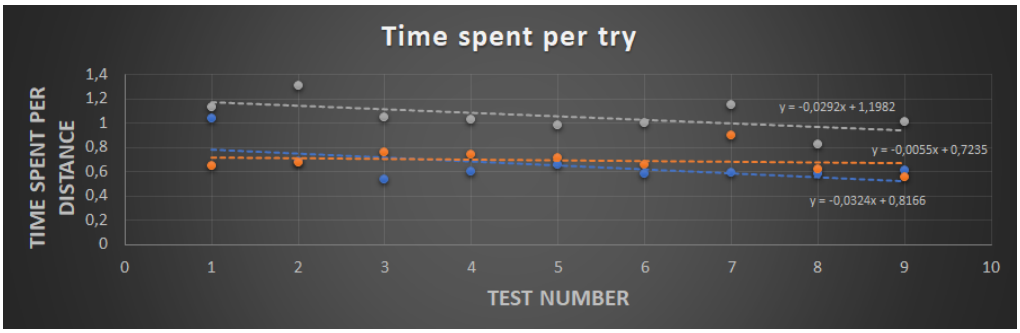


Figure 6.14: Time spent per distance per try

The time spent per distance per try is illustrated in Figure 6.14, showing that each participant, on average, decreased their completion time on subsequent trials. The data does, however, not show whether the performance increase is due to the same location of the targets or if the participants got more comfortable with the system. While the decrease is relatively small, it might influence the results in the evaluation and the location of the targets should therefore be randomized between tries.

Summary

We developed a system based on design considerations described in the previous chapter with the purpose of answering the problem statement. This system consisted of multiple ROS nodes written in C++ and Python, as well as physical products such as the UR3 and Shimmer GSR+. A number of internal tests were made in order to establish parameters for a variety of these, such as weights, and for the evaluation, such as in Section 6.3.5. Evaluating this system with a number of participants should be able to answer the underlying questions posed in the problem statement.

7 | Evaluation

In order to test the viability of the solution, an evaluation of the final product is necessary. This chapter describes the purpose of the test, including the hypotheses proposed, the method, and the results. The results are largely split in two categories, namely subjective and objective measurements. However, none have greater value than the other. The evaluation additionally is not expected to provide a conclusively "best" version of the system, but instead discover which (if any) areas each version excels and falls behind in.

7.1 Purpose

The purpose of the evaluation is to answer the problem statement posed in Chapter 4. In order to do that, we establish a number of measures that can describe likeability, as well as objective measures that could quantify the amount of assistance the methods provided in this scenario. Specifically, we want to answer whether the likeability and interaction is affected by the arbitration model and whether GSR is taken into account, and whether the system can provide objectively more efficient interactions than without it.

7.2 Method

The evaluation was performed at CREATE, Aalborg University and included 26 test participants, however due to complications 2 of these are blocked. The participants had a mean age of 24 with a standard deviation of 2. The participants were asked

to perform a maneuver task in which three targets should be reached with the robot arm in order. The participants were asked to perform the task a total of five times with different parameters, with the targets pseudo-randomly placed on the table. The different parameters were as follows:

- C - Control
- Tp - Timid Arbitration Curve with GSR
- T - Timid Arbitration Curve without GSR
- Ap - Aggressive Arbitration Curve with GSR
- A - Aggressive Arbitration Curve without GSR

The order of variables was randomized for each participant. We separate versions with and without GSR entirely, as we expect their inclusion to have different effects based on the arbitration curve they are modifying. After each trial, participants were asked to rate two statements from 1-7. The two questions were:

- "On a scale from 1-7, where 1 is hindered and 7 is assisted, I felt:"
- "On a scale from 1-7, where 1 is hard and 7 is easy, hitting the goals was:"

These two statements should provide enough information about the perceived interaction and the likeability of the system. Additionally, we recorded each participant's input and total time in order to determine if the arbitration methods had any quantifiable effects on the efficiency of the interaction.

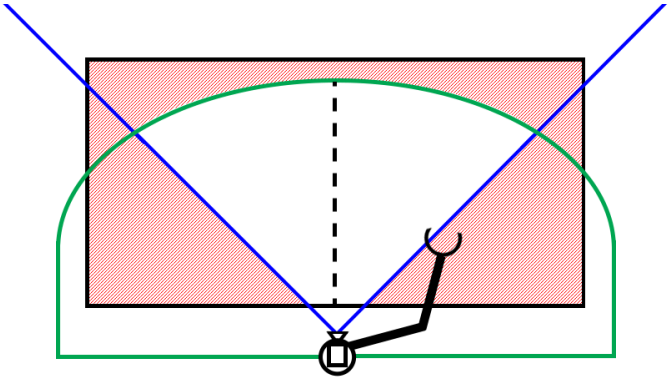


Figure 7.1: Setup limitations. Blue indicates the field of view of the camera and green indicates the reach of the robot arm. Areas marked in red denote parts of the table that cannot be used in the evaluation

The current evaluation setup is limited by the camera and the robot arm. The current camera's field of view is 87° and the range of motion of the robot arm is limited to roughly 70cm with the gripper attachment. This is illustrated in Figure 7.1, where large parts of the table is out of reach of the robot arm, thus limiting the amount of space for the different targets. This means that the amount of targets is reduced compared to initial evaluation setup ideas, but still allow for test setups with multiple targets for the system to detect and the user to reach. Furthermore the table is placed closely to the wall to decrease the amount of light artifacts created by the light in the room when viewed through the RGB-D camera. Based on these constraints and conditions, the evaluation setup was created as shown below in Figure 7.2.

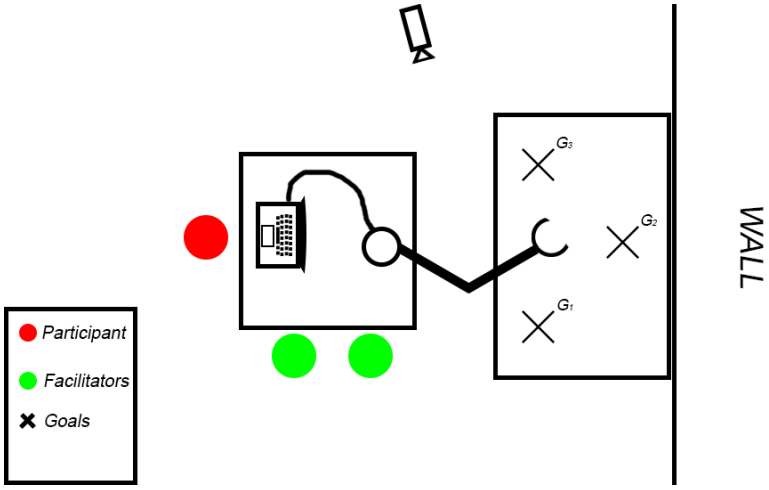


Figure 7.2: Illustration of evaluation setup

7.3 Results

The results in this section incorporate both the subjective measures as well as the objective measures to determine the effectiveness of the proposed modes. The subjective measures include the results from the questionnaire answered in between trials, while the objective measures were recorded during the test.

7.3.1 Subjective Measures

This section shows the results from the questionnaires scored by participants in between trials. The consent form and questionnaire can be seen in Appendix A and Appendix B respectively.

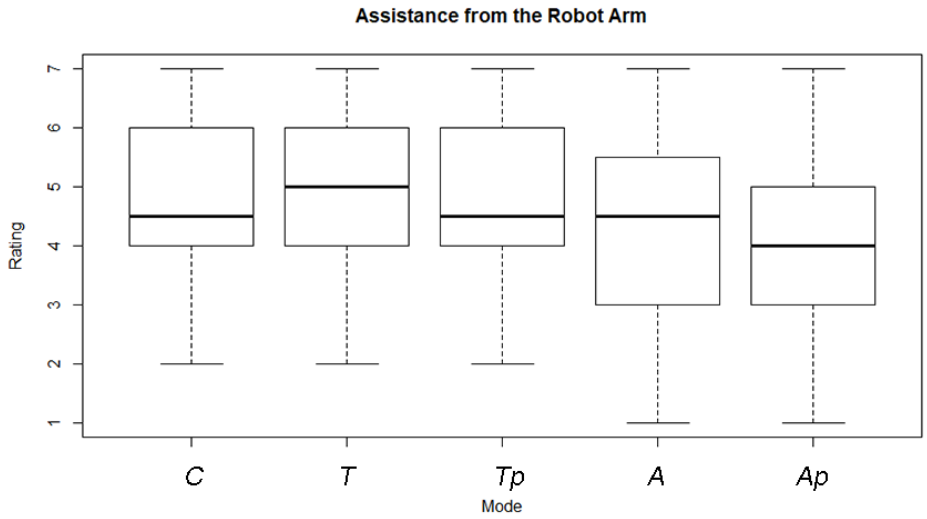


Figure 7.3: Results from questionnaire on the assistance from robot arm

Figure 7.3 shows a box plot of the results from the first question posed in the questionnaire. The results show that all of the modes have a large range of values with only the aggressive modes reaching the entire scale. All of the modes have a mean value between 4 and 5.

```
> summary(model)
             Df Sum Sq Mean Sq F value Pr(>F)
ind           4  12.72   3.179   1.595   0.18
Residuals   115 229.25   1.993
```

Figure 7.4: ANOVA test of assistance from the robot arm

In order to determine whether there was a significant difference between one or more mode in feelings of assistance, we perform an ANOVA test on the data shown in Figure 7.4. Since $p > 0.05$ we cannot prove such a relationship in this data.

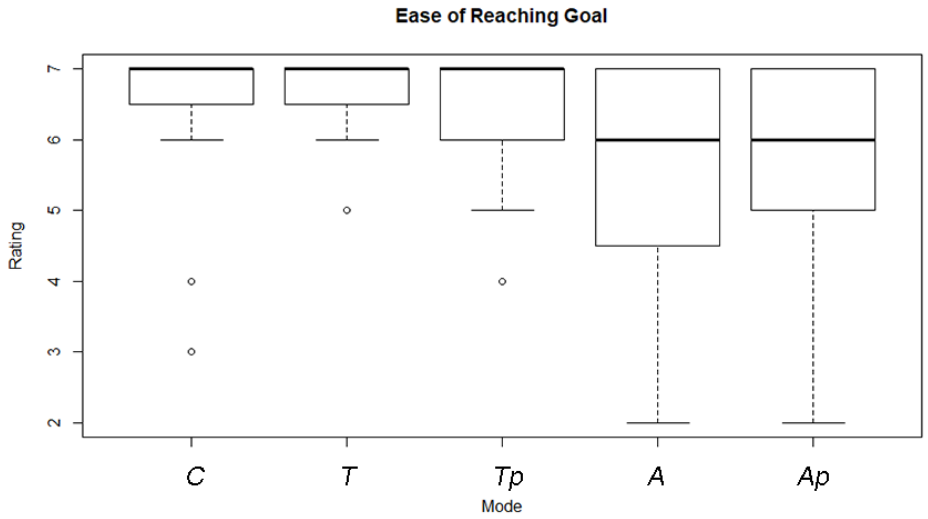


Figure 7.5: Results from the questionnaire on the ease of reaching goal

Figure 7.5 shows a box plot of the results from the second question in the questionnaire. The results show that the median from C, T, and Tp are at 7 whereas both A and Ap is a 6. Additionally, both A and Ap have significantly lower edge ranges, while the rest have very few low outliers.

The participants were also asked whether they felt the control scheme was intuitive and to rate in on a 7 point likert scale. 20 out of the 24 participants rated the intuitiveness of the control scheme to be 7, while 2 rated it 6, and another 2 rated it 5.

While the subjective measures indicate that the aggressive modes are performing below the control sample we can also analyze the objective measures from the evaluation. The idea of the these different modes was to decrease the amount of time spent on each task as well as decrease the amount of adjustments made by the user.

7.3.2 Objective Measures

While participants performed the test, their time and amount of adjustments made were recorded. An adjustment was defined to be any change in input, not a series of

similar inputs.

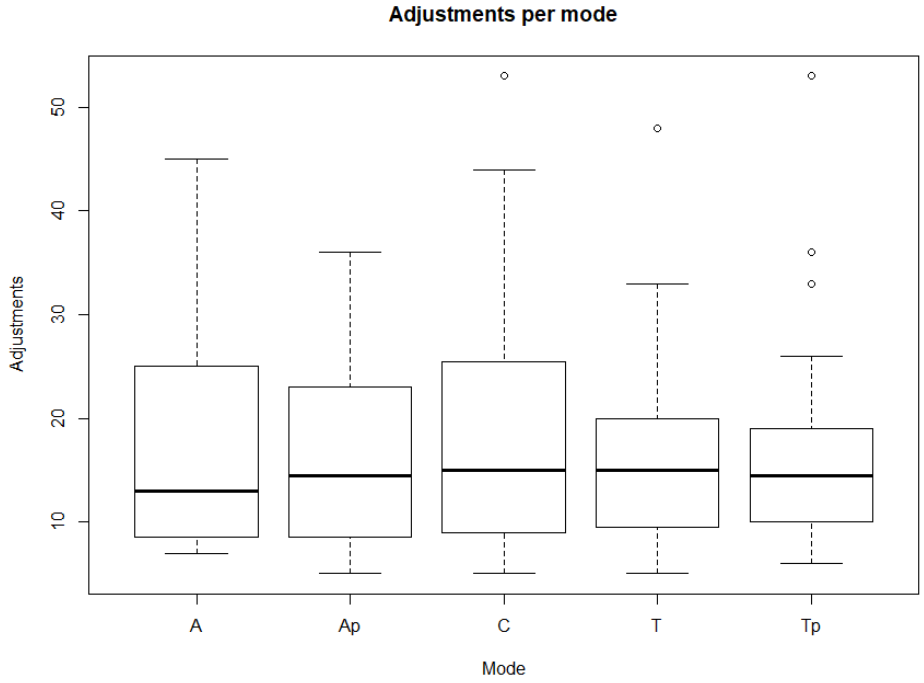


Figure 7.6: Adjustments for the different arbitration methods

Figure 7.6 shows the amount of adjustments made in each trial per mode. This figure shows that versions with GSR modulation enabled have lower upper ranges than their non-enabled equivalents. Additionally, both Timid versions appear to use fewer adjustments in the upper ranges than their Aggressive counterparts, while the median values of the Aggressive versions are slightly lower than the Timid ones. The highest median value is Control and Timid with 15, and the lowest is Aggressive with only 13.

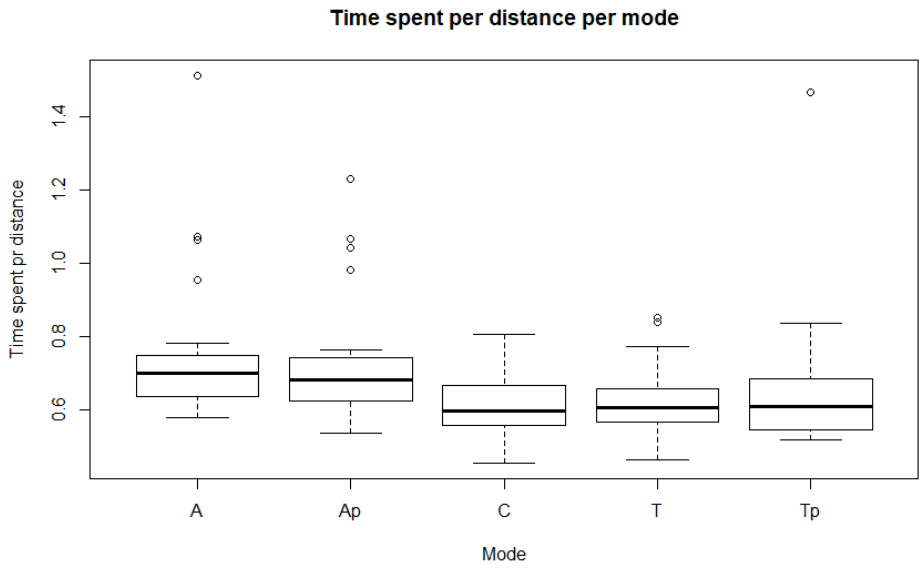


Figure 7.7: Time per distance used for the different arbitration methods

Figure 7.7 illustrates the amount of time spent normalized by the distance travelled per mode. This is done to indicate average time spent on each different mode while considering the distance travelled by the robot arm. This is done as the targets' locations are randomized between tries, and thus the distance changes between each test. The means of both of the aggressive modes are slightly higher than the control, whereas the timid ones are close to the control. Furthermore, the aggressive modes have some high outliers, while the timid only had a few.

As we are testing whether different modes are affecting both time spent on each task and the total amount of adjustments made by the users, we can use the ANOVA method in order to determine if there is a significant difference in performance. To do so we have to define a null hypothesis and an alternative hypothesis:

- *Null-hypothesis*: No mode performs better than the others
- *Alternative-hypothesis*: At least one mode performs better than another

We will examine these hypotheses based on previously mentioned criteria; time spent and adjustments, and determine if any of the modes are significantly improving performance.

```
> model2 <- aov(Time$Adjustments~Time$Mode)
> summary(model2)
             Df Sum Sq Mean Sq F value Pr(>F)
Time$Mode     4    130   32.42   0.274  0.894
Residuals   115  13586  118.13
```

Figure 7.8: ANOVA analysis of the Adjustments per mode

The results of the ANOVA test for the amount of adjustments made is shown in Figure 7.8 and shows that the p-value for the test did not result in a value lower than 0.05, thus concluding that there is no significant improvement between any of the modes.

```
> kruskal.test(TimeOverDist ~Mode, data = Time)

        Kruskal-Wallis rank sum test

data:  TimeOverDist by Mode
Kruskal-Wallis chi-squared = 21.125, df = 4, p-value = 0.0002991
```

Figure 7.9: Kruskal-Wallis analysis of the time spent per distance per mode

```
> dunn.test(Time$TimeOverDist, Time$Mode, method="bonferroni")
  Kruskal-Wallis rank sum test

data: x and group
Kruskal-Wallis chi-squared = 21.1248, df = 4, p-value = 0


                           Comparison of x by group
                                 (Bonferroni)
Col Mean-|
Row Mean |          A         Ap          C          T
---------+--------------------------------------------
      Ap |   0.531124
         |     1.0000
         |
       C |   3.394220   2.863095
         |    0.0034*    0.0210*
         |
       T |   3.132807   2.601682  -0.261413
         |    0.0087*    0.0464     1.0000
         |
      Tp |   3.087163   2.556038  -0.307056  -0.045643
         |    0.0101*    0.0529     1.0000     1.0000

alpha = 0.05
Reject Ho if p <= alpha/2
```

Figure 7.10: Dunn's test with Bonferroni correction of the time spent per mode

The results from the time test were non-parametric, so we opted to use the non-parametric ANOVA equivalent, the Kruskal-Wallis test, shown in Figure 7.9. Seeing

a significant difference in the data, we analyze the methods pairwise with Dunn's test, using the Bonferroni correction to adjust the p-value aggressively, shown in Figure 7.10. In this analysis we are able to see that the control sample outperforms both aggressive arbitration modes with p-values of 0.0034 and 0.021. Additionally, both T and Tp appear to outperform A, but only C outperforms Ap.

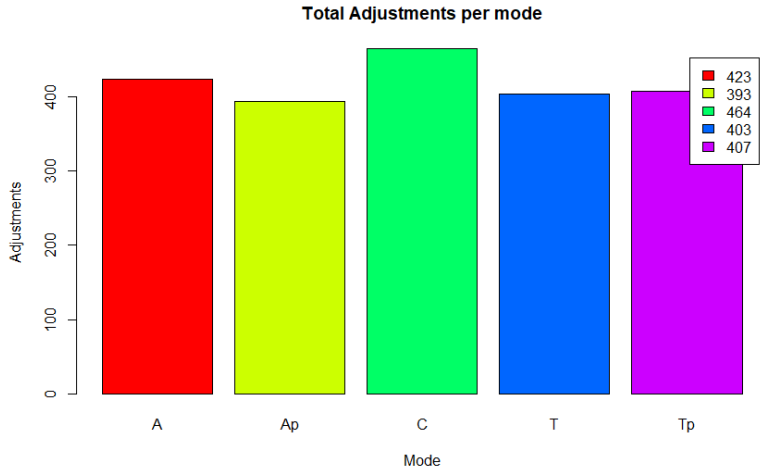


Figure 7.11: Total adjustments per mode

Figure 7.11 shows the total amount of adjustments per mode. It shows that the control mode had a noticeably higher count than the rest of the modes. We can further analyse this data and express how it was performed by the individual users.

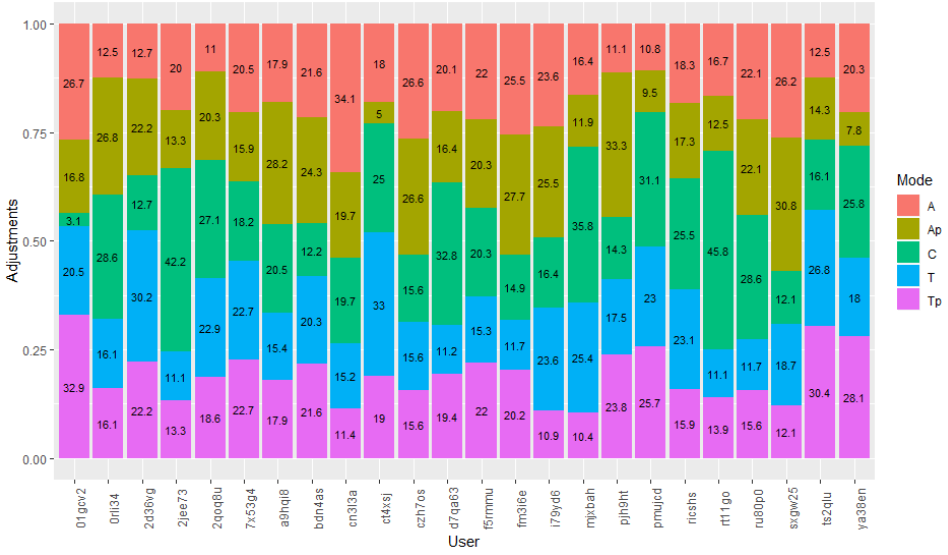


Figure 7.12: Adjustments per mode per user in percentages

Figure 7.12 shows how the adjustments were distributed by each participant and shows the percentage for each user.

7.4 Discussion

The aggressive modes have a noticeable higher completion time which could be caused by some participants wanting to hit the target in the front. As the system is just trying to find the shortest possible trajectory to the center of the object, there has been some conflict between the system and these users. This made the users move the robot backwards multiple times in order to get their wanted trajectory to the target, increasing their time spent on the task.

Considering the results in Figure 7.7, it seems that Ap fares slightly better than A, while Tp fares slightly worse than T. This seems to correlate with the prediction that aggressive arbitration modes can be overwhelming and require temporary adjustments, while timid versions do not. It also indicates that a minimum of assistance can be valuable at all times, and that modulating modalities should not reduce confidence to negligible levels. However, in Figure 7.6 this seems to be reversed, at least when looking at median levels. This seems to indicate that there is a tradeoff

between the two which needs to be considered for each individual user.

Another issue that happened during testing was the vibrations of the robot sometimes offsetting the camera slightly, resulting in misclassification of the objects in the scene. This happened on at least one occasion where the participant tried to reach the left most object, but was not able to fully reach the object, as the system saw the target had been reached.

It is important to note that every participant had their own distinctive trajectory pattern, meaning that some would go straight from one target to the next, while others would move the robot backwards, to the side, and then forwards. This should be accounted for by the randomization of the target, however in some cases - when the targets are close - a clear distinction could be made from the same participant.

After testing, the participants were asked whether they had any comments about the system and a noticeable comment was that the aggressive mode felt more like a hindrance than an aid. The participants explained that the robot moved in a different way than intended and thus had to make more corrections. However, the participant did understand that the robot was aiding them in reaching the correct target, however in a different trajectory than the participant would have preferred.

8 | Discussion

This chapter reflects on the project, specifically the results of the evaluation, to provide insights in certain choices and outcomes and to present considerations for future iterations. This will be split in two parts; *Technology* will discuss both the solution and the premise of the project in order to asses how they compare, and their impact on the results. *Wider Context* will discuss the current solution's viability and improvements in potential future works as well as open the scope to discuss its implementation in other contexts.

8.1 Technology

The solution is part of a much greater scope, the EXOTIC project, hence it required a number of tangentially related functionalities to work. For instance, computer vision was not entirely necessary, but it does introduce a degree of noise into the system that is necessary for it to be able to handle. Additionally, the hardware available in the test had an influence both on the design and the evaluation. The time test which determined the weights of the trajectories in the intent prediction, for example, was predicated on a set speed of the robot arm. Furthermore, the interactions with the robot arm and the limited range in combination with the camera means that the test had a limited area, hence the limited and sometimes frustrating test premise.

Expanding on the evaluation, the test setup did not introduce much variability in the depth axis after the first touch and none whatsoever in the height axis. This means there is a large area of interactions still not evaluated, a considerable issue to note when examining the results of the project. This is somewhat explained by the

limited range of the robotic arm and camera combination, as well as a consideration for test participants and an assumption that most ADL interactions are separate mainly on a plane such as a table.

Another issue with the test, which has been observed multiple times throughout, is the split between two types of participants; those that allow the system to make the interaction faster and those that struggle against it to make it more precise, or at least do it their way. As explained previously, this group of participants wanted the interaction, the touching of the goal, to happen straight on, while the robot never considered this. Instead, it was programmed to move directly towards the intended goal. This skews the results quite significantly, but also provides some information about the intent of some users; it has to be their way, regardless of efficiency. It can also be a symptom of the limited test interaction opportunities, and an evaluation incorporating grasping of specific points may be required to accurately ascertain the impact of intent prediction and shared control.

The method used is a novel idea based on the specific problem of the EXOTIC project, the necessity of moving between objects to interact in sequences as humans normally do. This is necessary for supporting ADL. Though the only significant results were that aggressive arbitration methods performed slightly worse than others in some metrics, the reasons for which are largely described in previous sections, there have been some key conclusions made besides it. Firstly, the user interactions, fighting against the system as described in the previous section, is key to further development and evaluations. Secondly, there are a number of visual changes that support the timid arbitration method in many cases. It slightly improved the feeling of assistance, shown in Figure 7.3, had fewer negative outliers in ease of reaching goals, shown in Figure 7.5, and fewer average adjustments made than control. Of course, these results are speculatory as none are statistically significant, but the coincidence is interesting enough to consider in future tests and long term evaluations.

The project has worked with two input modalities, namely the keyboard input used in the intent prediction node and the GSR input used to modulate the confidence hereof. However, we also discuss a variety of other modalities, such as gaze and facial expressions. These additional modalities fall into the same two categories of the ones used; predictors and verifiers. This opens the scope to a host of new opportunities

and issues; more predictors and verifiers can potentially improve the intent prediction aspect, but a thorough study of their inclusion and relation to one another is required, namely how to mix two predictors and how verifiers can modulate these.

The arbitration curves used in the evaluation have been described as timid and aggressive. These were used for their inherent properties, namely that one cannot assume more control than the user and the other could assume full control. In the autonomous car level analogy described in Section 3.5.1, these can be described as levels 2 and 3, respectively. However, the results indicate that a tradeoff between some measures exists, and using these two extreme cases could potentially have skewed the results. Additionally, arbitration curves exist in more variety than merely linear models. Though we did not make efforts to find an optimal level or model of arbitration, this is certainly required for a final product.

8.2 Wider Context

As mentioned, the current system only incorporates two arbitration methods, timid and aggressive, which has caused some issues with users and their intended movement of the robot arm. This might be relieved by implementing customizable arbitration curves which could be fitted to the individual user, while also allowing the user to define whether they want to hit the target in the front or the shortest path. Allowing the users to have their individual settings, might also allow them to change between modes, such that if they for instance are tired or struggling they could change the settings to receive more or less aid from the system.

There are many possibilities for improvements of the systems including the introduction of more modalities and solutions to increase the usability of the system. Introducing an additional camera might increase the object detection of the system, while reducing the amount of errors caused by camera displacement or occlusion. This could also prove beneficial for systems with increased range of motion as well as movement in the height axis. As the object detection could be improved by the usage of multiple cameras, a more precise classification of the objects is possible. This would improve the effectiveness of an affordance algorithm, as it would allow the system to correctly estimate the uses of every object and the interaction possi-

bilities. It should be noted that the introduction of multiple modalities as well as more algorithms would increase the complexity of the system which might have an impact on the performance due to increased computational load.

As this system is used for ADL for tetraplegics, it is important to investigate the long-term capabilities and effectiveness of the shared control system. The users in this evaluation had never tried such a system before and were only able to use it for the duration of the tests. This means that no tests have been performed on users with experience with the system, thus only first hand performance has been reviewed. As previously tested, an increase in performance was noticeable in users trying the system with the same setup, however we were not able to distinguish whether that was a result of having the same test setup or due to increased familiarity with the system. This also begs the question whether learning to work with the system rather than against it can improve results over time, opening the possibility for a long-term evaluation.

9 | Conclusion

This project has focused on answering the following problem statement:

> "How can shared control based on intent prediction be implemented in a tongue controlled personal robotic arm to facilitate and improve performance of activities of daily living by tetraplegics?"

To this end, we developed a system that predicted user intents for reaching goals with a robot arm based on previous input and physiological data. We evaluated with a number of able-bodied participants, investigating whether such a system could improve performance or interaction likeability in simple tasks. Concluding on the tests, we cannot see any significant improvements in such limited interaction opportunities compared to the control, only that both aggressive methods had significantly reduced scores in certain measures. However, some results indicate that improvements are possible with such a system, and we make a number of other important observations. Some results indicate that there are slight improvements to likeability or amount of adjustments needed to reach a number of goals depending on arbitration levels, however we see no significant improvements over the control samples.

We find some evidence of learning curves being an inhibitive factor in the evaluation, noting that one group of participants worked against the system's assistance to reach the goals their way. This interaction was not something we had predicted. We speculate that more specific goals, such as reaching handles on cups or grasping objects in a specific way, will improve results of both arbitration methods since users will have to rely more on the assistance provided by the computer. Additionally, it is very relevant to perform long-term tests with such a system, since more experienced users can learn to use the robot's movements instead of working against them.

The project has been made tangentially to the EXOTIC project, which begs the question whether this study is relevant to their research. We believe that it is, as we have developed a solution which estimate intent and use that information in a shared control system in context. Additionally, we make a number of observations that could be vital to further development. This includes the tradeoff between likeability and assistance based on arbitration curves, and the necessity of optimizing the shared control levels to each user. Furthermore, this research is relevant in contexts outside the scope of EXOTIC, specifically in projects that aim to provide real-time sequential assistance to changing environments based on sensor data, and research in including additional sensor data in multimodal systems.

Bibliography

[1] Rigshospitalet. Fakta om rygmarvsskade, September 2018. URL https://www.rigshospitalet.dk/afdelinger-og-klinikker/neuro/klinik-for-rygmarvsskader/undersogelse-og-behandling/Sider/fakta-om-rygmarvsskade.aspx.

[2] Robohub Editors. Robotics 2020 Multi annual roadmap, H2020, Release B 02/12/2016. *Robotics 2020 Multi-Annual Roadmap*, page 331, 2016.

[3] Yuying Chen, Ying Tang, Lawrence C. Vogel, and Michael J. DeVivo. Causes of Spinal Cord Injury. *Topics in Spinal Cord Injury Rehabilitation*, 19(1):1–8, 2013. ISSN 1082-0744. doi: 10.1310/sci1901-1. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3584795/.

[4] A R Craig, K M Hancock, and H G Dickson. A longitudinal investigation into anxiety and depression in the first 2 years following a spinal cord injury. *Spinal Cord*, 32(10):675–679, October 1994. ISSN 1362-4393, 1476-5624. doi: 10.1038/sc.1994.109. URL http://www.nature.com/articles/sc1994109.

[5] Rune Schmidt. Nedslidning på det danske arbejdsmarked. page 18, 2010.

[6] Århus Kommune. Timetakster: BPA Aarhus, 2017. URL http://www.aarhuskommune.dk/sitecore/content/Subsites/Hjaelperordningen/Home/Handicaphjaelper/Timetakster.aspx?sc_lang=da&sc_lang=da.

[7] Satoko Yasuda, Paul Wehman, Pamela Targett, David X. Cifu, and Michael West. Return to work after spinal cord injury: a review of recent research. *NeuroRehabilitation*, 17(3):177–186, 2002. ISSN 1053-8135.

[8] V. Maheu, P. S. Archambault, J. Frappier, and F. Routhier. Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–5, June 2011. doi: 10.1109/ICORR.2011.5975397.

[9] Anca D Dragan and Siddhartha S Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, June 2013. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364913490324. URL http://journals.sagepub.com/doi/10.1177/0278364913490324.

[10] J. D. R. Millán, R. Rupp, G. R. Müller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kübler, R. Leeb, C. Neuper, K.-R. Müller, and D. Mattia. Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. *Frontiers in Neuroscience*, 4, 2010. ISSN 1662-453X. doi: 10.3389/fnins.2010.00161.

[11] A. Tabor, S. Bateman, and E. Scheme. Evaluation of Myoelectric Control Learning Using Multi-Session Game-Based Training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(9):1680–1689, September 2018. ISSN 1534-4320. doi: 10.1109/TNSRE.2018.2855561.

[12] Eric R. Kandel, editor. *Principles of neural science*. McGraw-Hill, New York, 5th ed edition, 2013. ISBN 978-0-07-139011-8.

[13] Lotte N. S. Andreasen Struijk, Line Lindhardt Egsgaard, Romulus Lontis, Michael Gaihede, and Bo Bentsen. Wireless intraoral tongue control of an assistive robotic arm for individuals with tetraplegia. *Journal of NeuroEngineering and Rehabilitation*, 14(1), December 2017. ISSN 1743-0003. doi: 10.1186/s12984-017-0330-2. URL http://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-017-0330-2.

[14] Joanna Lumsden, editor. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology:*. IGI Global, 2008. ISBN 978-1-59904-871-0 978-1-59904-872-7. doi: 10.4018/978-1-59904-871-0. URL http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-871-0.

[15] Mehdi Khosrow-Pour, editor. *Dictionary of Information Science and Technology*

(2nd Edition):. IGI Global, 2013. ISBN 978-1-4666-2624-9 978-1-4666-2674-4. doi: 10.4018/978-1-4666-2624-9. URL http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2624-9.

[16] R. Chipalkatty, G. Droge, and M. B. Egerstedt. Less Is More: Mixed-Initiative Model-Predictive Control With Human Inputs. *IEEE Transactions on Robotics*, 29(3):695–703, June 2013. ISSN 1552-3098. doi: 10.1109/TRO.2013.2248551.

[17] Hocheol Shin, Seung Ho Jung, You Rack Choi, and ChangHoi Kim. Development of a shared remote control robot for aerial work in nuclear power plants. *Nuclear Engineering and Technology*, 50(4):613–618, May 2018. ISSN 1738-5733. doi: 10.1016/j.net.2018.03.006. URL http://www.sciencedirect.com/science/article/pii/S1738573318300810.

[18] Yanan Li, Keng Peng Tee, Shuzhi Sam Ge, and Haizhou Li. Building Companionship through Human-Robot Collaboration. In Guido Herrmann, Martin J. Pearson, Alexander Lenz, Paul Bremner, Adam Spiers, and Ute Leonards, editors, *Social Robotics*, Lecture Notes in Computer Science, pages 1–7. Springer International Publishing, 2013. ISBN 978-3-319-02675-6.

[19] D. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal. How Autonomy Impacts Performance and Satisfaction: Results From a Study With Spinal Cord Injured Subjects Using an Assistive Robot. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(1):2–14, January 2012. ISSN 1083-4427. doi: 10.1109/TSMCA.2011.2159589.

[20] J. Kofman, Xianghai Wu, T.J. Luu, and S. Verma. Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Transactions on Industrial Electronics*, 52(5):1206–1219, October 2005. ISSN 0278-0046, 1557-9948. doi: 10.1109/TIE.2005.855696. URL http://ieeexplore.ieee.org/document/1512452/.

[21] Wentao Yu, R. Alqasemi, R. Dubey, and N. Pernalete. Telemanipulation Assistance Based on Motion Intention Recognition. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1121–1126, April 2005. doi: 10.1109/ROBOT.2005.1570266.

[22] Christopher Schultz, Sanket Gaurav, Mathew Monfort, Lingfei Zhang, and Brian D. Ziebart. Goal-predictive robotic teleoperation from noisy sensors. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5377–5383, Singapore, Singapore, May 2017. IEEE. ISBN 978-1-5090-4633-1. doi: 10.1109/ICRA.2017.7989633. URL http://ieeexplore.ieee.org/document/7989633/.

[23] Zhikun Wang, Katharina Mülling, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic movement modeling for intention inference in human–robot interaction. *The International Journal of Robotics Research*, 32(7):841–858, June 2013. ISSN 0278-3649. doi: 10.1177/0278364913478447. URL https://doi.org/10.1177/0278364913478447.

[24] Kris Hauser. Recognition, prediction, and planning for assisted teleoperation of freeform tasks. *Autonomous Robots*, 35(4):241–254, November 2013. ISSN 1573-7527. doi: 10.1007/s10514-013-9350-3. URL https://doi.org/10.1007/s10514-013-9350-3.

[25] Brian D. Ziebart, Mathew Monfort, and Anqi Liu. Intent prediction and trajectory forecasting via Predictive inverse linear-quadratic regulation. *29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, IAAI 2015*, 5:7, June 2015.

[26] A. Doshi and M. Trivedi. A comparative exploration of eye gaze and head motion cues for lane change intent prediction. In *2008 IEEE Intelligent Vehicles Symposium*, pages 49–54, June 2008. doi: 10.1109/IVS.2008.4621321.

[27] A. Doshi and M. M. Trivedi. On the Roles of Eye Gaze and Head Dynamics in Predicting Driver's Intent to Change Lanes. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):453–462, September 2009. ISSN 1524-9050. doi: 10.1109/TITS.2009.2026675.

[28] Roman Bednarik, Hana Vrzakova, and Michal Hradis. What Do You Want to Do Next: A Novel Approach for Intent Prediction in Gaze-based Interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 83–90, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-

1221-9. doi: 10.1145/2168556.2168569. URL http://doi.acm.org/10.1145/2168556.2168569.

[29] Chien-Ming Huang, Sean Andrist, Allison Sauppé, and Bilge Mutlu. Using gaze patterns to predict task intent in collaboration. *Frontiers in Psychology*, 6, 2015. ISSN 1664-1078. doi: 10.3389/fpsyg.2015.01049. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2015.01049/full#B4.

[30] Michael Morales, Peter Mundy, and Jennifer Rojas. Following the direction of gaze and language development in 6-month-olds. *Infant Behavior and Development*, 21(2):373–377, January 1998. ISSN 0163-6383. doi: 10.1016/S0163-6383(98)90014-5. URL http://www.sciencedirect.com/science/article/pii/S0163638398900145.

[31] George Butterworth. The ontogeny and phylogeny of joint visual attention. In *Natural theories of mind: Evolution, development and simulation of everyday mindreading*, pages 223–232. Basil Blackwell, Cambridge, MA, US, 1991. ISBN 978-0-631-17194-2.

[32] Hong Zeng, Yanxin Wang, Changcheng Wu, Aiguo Song, Jia Liu, Peng Ji, Baoguo Xu, Lifeng Zhu, Huijun Li, and Pengcheng Wen. Closed-Loop Hybrid Gaze Brain-Machine Interface Based Robotic Arm Control with Augmented Reality Feedback. *Frontiers in Neurorobotics*, 11, October 2017. ISSN 1662-5218. doi: 10.3389/fnbot.2017.00060. URL http://journal.frontiersin.org/article/10.3389/fnbot.2017.00060/full.

[33] Jorge A Díez, José M Catalán, Luis D Lledó, Francisco J Badesa, and Nicolás Garcia-Aracil. Multimodal robotic system for upper-limb rehabilitation in physical environment. *Advances in Mechanical Engineering*, 8(9):168781401667028, September 2016. ISSN 1687-8140, 1687-8140. doi: 10.1177/1687814016670282. URL http://journals.sagepub.com/doi/10.1177/1687814016670282.

[34] Simona Crea, Marius Nann, Emilio Trigili, Francesca Cordella, Andrea Baldoni, Francisco Javier Badesa, José Maria Catalán, Loredana Zollo, Nicola Vitiello, Nicolas Garcia Aracil, and Surjo R. Soekadar. Feasibility and safety of shared EEG/EOG and vision-guided autonomous whole-arm exoskeleton control to perform activities of daily living. *Scientific Reports*, 8(1):10823,

December 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-29091-5. URL http://www.nature.com/articles/s41598-018-29091-5.

[35] Andrea Gigli, Arjan Gijsberts, Valentina Gregori, Matteo Cognolato, Manfredo Atzori, and Barbara Caputo. Visual Cues to Improve Myoelectric Control of Upper Limb Prostheses. *arXiv:1709.02236 [cs]*, August 2017. URL http://arxiv.org/abs/1709.02236. arXiv: 1709.02236.

[36] Andrew B. Schwartz. Movement: How the Brain Communicates with the World. *Cell*, 164(6):1122–1135, March 2016. ISSN 0092-8674. doi: 10.1016/j.cell.2016.02.038. URL http://www.sciencedirect.com/science/article/pii/S009286741630188X.

[37] Thomas Lampe, Lukas D.J. Fiederer, Martin Voelker, Alexander Knorr, Martin Riedmiller, and Tonio Ball. A Brain-computer Interface for High-level Remote Control of an Autonomous, Reinforcement-learning-based Robotic System for Reaching and Grasping. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pages 83–88, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2184-6. doi: 10.1145/2557500.2557533. URL http://doi.acm.org/10.1145/2557500.2557533.

[38] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir. The Future of Human-in-the-Loop Cyber-Physical Systems. *Computer*, 46(1):36–45, January 2013. ISSN 0018-9162. doi: 10.1109/MC.2013.31.

[39] N. Hanajima, T. Goto, Y. Ohta, H. Hikita, and M. Yamashita. A motion rule for human-friendly robots based on electrodermal activity investigations and its application to mobile robot. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3791–3797, Edmonton, Alta., Canada, 2005. IEEE. ISBN 978-0-7803-8912-0. doi: 10.1109/IROS.2005.1545301. URL http://ieeexplore.ieee.org/document/1545301/.

[40] G. Perugia, D. Rodríguez-Martín, M. Díaz Boladeras, A. C. Mallofré, E. Barakova, and M. Rauterberg. Electrodermal activity: Explorations in the psychophysiology of engagement with social robots in dementia. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communica-*

tion (RO-MAN), pages 1248–1254, August 2017. doi: 10.1109/ROMAN.2017.8172464.

[41] D. McDuff, R. E. Kaliouby, J. F. Cohn, and R. W. Picard. Predicting Ad Liking and Purchase Intent: Large-Scale Analysis of Facial Responses to Ads. *IEEE Transactions on Affective Computing*, 6(3):223–235, July 2015. ISSN 1949-3045. doi: 10.1109/TAFFC.2014.2384198.

[42] Nianyin Zeng, Hong Zhang, Baoye Song, Weibo Liu, Yurong Li, and Abdullah M. Dobaie. Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing*, 273:643–649, January 2018. ISSN 09252312. doi: 10.1016/j.neucom.2017.08.043. URL https://linkinghub.elsevier.com/retrieve/pii/S0925231217314649.

[43] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual Affordance and Function Understanding: A Survey. *arXiv:1807.06775 [cs]*, July 2018. URL http://arxiv.org/abs/1807.06775. arXiv: 1807.06775.

[44] Stefan Hein Bengtson. A Review of Computer Vision for Semi-autonomous Control of Assistive Robotic Manipulators (ARMs). *A Review of Computer Vision for Semi-autonomous Control of Assistive Robotic Manipulators (ARMs)*, 2019, 2019.

[45] Haoyong Yu, Matthew Spenko, and Steven Dubowsky. An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly. *An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly*, page 14, 2003.

[46] Smith, Bryant Walker. SAE Levels of Driving Automation, January 2019. URL http://cyberlaw.stanford.edu/blog/2013/12/sae-levels-driving-automation.

[47] Kyle Hyatt. Self-driving cars: A level-by-level explainer of autonomous vehicles, January 2019. URL https://www.cnet.com/roadshow/news/self-driving-car-guide-autonomous-explanation/.

[48] Pradipta Biswas, Gokcen Aslan Aydemir, Pat Langdon, and Simon Godsill. Intent Recognition Using Neural Networks and Kalman Filters. In Andreas

Holzinger and Gabriella Pasi, editors, *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, pages 112–123, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-39146-0.

[49] D. Kulić and E. A. Croft. Estimating Intent for Human-Robot Interaction. In *in IEEE Int. Conference on Advanced Robotics*, page 810815, 2003.

[50] T. Carlson and Y. Demiris. Human-wheelchair collaboration through prediction of intention and adaptive assistance. In *2008 IEEE International Conference on Robotics and Automation*, pages 3926–3931, May 2008. doi: 10.1109/ROBOT.2008.4543814.

[51] Yu Shi, Natalie Ruiz, Ronnie Taib, Eric Choi, and Fang Chen. Galvanic skin response (GSR) as an index of cognitive load. In *CHI '07 extended abstracts on Human factors in computing systems - CHI '07*, page 2651, San Jose, CA, USA, 2007. ACM Press. ISBN 978-1-59593-642-4. doi: 10.1145/1240866.1241057. URL http://portal.acm.org/citation.cfm?doid=1240866.1241057.

[52] Michèle Hubli and Andrei V. Krassioukov. How reliable are sympathetic skin responses in subjects with spinal cord injury? *Clinical Autonomic Research*, 25(2):117–124, April 2015. ISSN 1619-1560. doi: 10.1007/s10286-015-0276-z. URL https://doi.org/10.1007/s10286-015-0276-z.

[53] Sence22. Products – TKS A/S, 2019. URL http://tks-technology.dk/index.php/products/.

[54] Mayank Goel, Chen Zhao, Ruth Vinisha, and Shwetak N. Patel. Tongue-in-Cheek: Using Wireless Signals to Enable Non-Intrusive and Flexible Facial Gestures Detection. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pages 255–258, Seoul, Republic of Korea, 2015. ACM Press. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702591. URL http://dl.acm.org/citation.cfm?doid=2702123.2702591.

[55] Patricia J. Manns and Karen E. Chad. Components of Quality of Life for Persons with a Quadriplegic and Paraplegic Spinal Cord Injury. *Qualitative Health Research*, 11(6):795–811, November 2001. ISSN 1049-7323, 1552-7557. doi: 10.1177/104973201129119541. URL http://journals.sagepub.com/doi/10.1177/104973201129119541.

[56] Wikipedia. 3dtree, 2019. URL https://upload.wikimedia.org/wikipedia/commons/b/b6/3dtree.png.

[57] Point Cloud Library. Documentation - Point Cloud Library (PCL), November 2018. URL http://pointclouds.org/documentation/tutorials/cluster_extraction.php.

[58] Point Cloud Library. Documentation - Point Cloud Library (PCL), November 2018. URL http://pointclouds.org/documentation/tutorials/planar_segmentation.php#planar-segmentation.

List of Figures

A Consent Form

Consent Form

The purpose of this study is to investigate how a shared control system can be implemented such that tetraplegics and other non-able-bodied are able to perform simple activities of daily living without additional personal assistance.

I understand, that at any point I can ask questions to the project and to the methods used by the students in the study.

I understand that all the data collected from the study will be used by the students in their project and under no circumstances will include names or other identifying characteristics. I understand that my anonymity will be protected and all the data I present will be classified. My participation in this project is voluntary and I have the right to not participate in the study. I can freely choose to not answer any of the questions without consequences. I can at any point choose to not participate in the study without consequences.

I understand that the interview may be recorded given my permission and these recording will be deleted once the project is completed. I understand that the recording and the project will be kept safe and secured. The interview will not be public and will only be presented to individuals connected to Aalborg University.

I hereby, give my consent to be interviewed and participate in the project:

Yes ☐

No ☐

I hereby give my consent to be recorded:

Yes ☐

No ☐

Participant: _____ Date_____

Contact:

Frederik Falk

Ffalk13@student.aau.dk

Oliver G. Hjermitslev

Ohjerm14@student.aau.dk

B | Questionnaire

Questionnaire – EXOTIC Research

Name:

Age:

Please only mark one circle and not in-between.

Using the robot arm I felt my movements were:

	1	2	3	4	5	6	7	
Hindered	○	○	○	○	○	○	○	Assisted

Reaching the end goal was:

	1	2	3	4	5	6	7	
Hard	○	○	○	○	○	○	○	Easy

Using the robot arm I felt my movements were:

	1	2	3	4	5	6	7	
Hindered	○	○	○	○	○	○	○	Assisted

Reaching the end goal was:

	1	2	3	4	5	6	7	
Hard	○	○	○	○	○	○	○	Easy

Using the robot arm I felt my movements were:

	1	2	3	4	5	6	7	
Hindered	○	○	○	○	○	○	○	Assisted

Reaching the end goal was:

	1	2	3	4	5	6	7	
Hard	○	○	○	○	○	○	○	Easy

Using the robot arm I felt my movements were:

	1	2	3	4	5	6	7	
Hindered	○	○	○	○	○	○	○	Assisted

Reaching the end goal was:

	1	2	3	4	5	6	7	
Hard	○	○	○	○	○	○	○	Easy

Using the robot arm I felt my movements were:

	1	**2**	**3**	**4**	**5**	**6**	**7**	
Hindered	○	○	○	○	○	○	○	Assisted

Reaching the end goal was:

	1	**2**	**3**	**4**	**5**	**6**	**7**	
Hard	○	○	○	○	○	○	○	Easy

The controls were intuitive (Easy to understand and use):

	1	**2**	**3**	**4**	**5**	**6**	**7**	
No	○	○	○	○	○	○	○	Yes