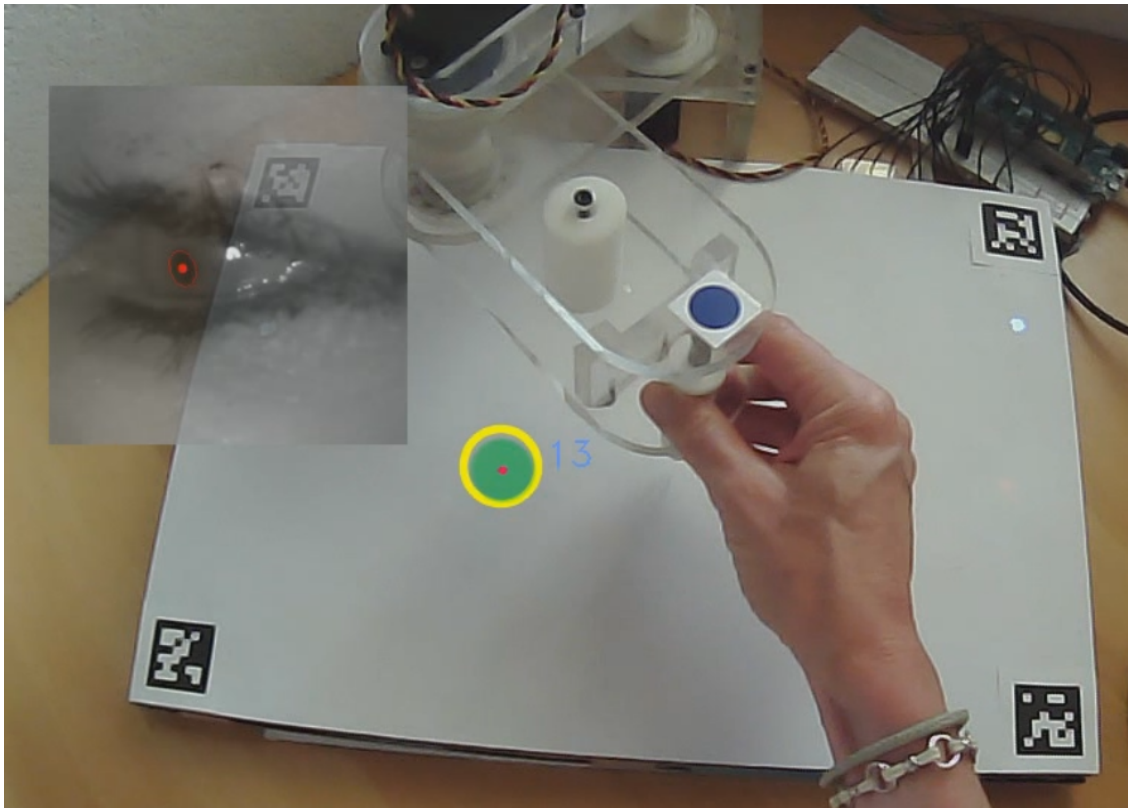

Human intent-recognition system for safety-critical human-machine interactions

Master's thesis in Complex Adaptive Systems

Simon Künzler

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Human intent-recognition system for safety-critical human-machine interactions

SIMON KÜNZLER

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Human intent-recognition system for safety-critical human-machine interactions
SIMON KÜNZLER

Supervisor: Pinar Boyraz Baykas, Department of Mechanics and Maritime Sciences
Examiner: Pinar Boyraz Baykas, Department of Mechanics and Maritime Sciences

Master's Thesis 2020:52
Department of Mechanics and Maritime Sciences
Vehicle Safety Division
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Subject performing experiment under assistance of shared-control algorithm

Typeset in LaTeX
Gothenburg, Sweden 2020

Abstract

The aim of this thesis was to investigate the potential of eye tracking technology, to help recognizing the intent of humans when working with a machine under shared control. An experiment was designed to study the eye gaze behaviour of test subjects, while manipulating a two degrees-of-freedom (DOF) SCARA robot. The subjects were given the task to maneuver the end-effector of the robot through a sequence of LEDs located on the robot action plane. The LED sequence was different for each experiment run and not known by the subjects before the start of each run. In the first step, eye gaze data was collected while the robot was unactuated. The fixation point of the subjects gaze was 4.5 times more likely to be in the proximity of the goal LEDs they intended to connect, opposed to fixating a point outside of the intended area. In addition, when the subjects planned to move from one LED to the next, the subject's gaze tended to fixate on the next LED between one and two seconds before reaching the position with the robot end-effector, depending on how much distance the subject had to cover when moving from the current LED to the next. After reaching the fixated position, the gaze is shifted almost immediately (with 0.1-0.2s delay) onto the next LED, while movement onset is delayed about 0.5 seconds. This information was then used to develop an algorithm to predict which LED a subject is intending to reach. While performing a second set of tests, more data was collected, but this time under shared-control with the robot. The implemented algorithm was able to successfully identify the next goal LED in the subject's planned path and to provide assistance in the movement of the robot arm. How far ahead of time the goals were recognized was dependent on how soon the subjects gaze shifted from a reached LED to their next planned goal LED. If the subject fixates on a goal LED 0.3s before initiating the movement towards it, the robot was able to perform the whole movement between the LEDs. In most cases the algorithm initiated the support half-way through the planned motion of the subjects. No significant differences in the subjects gaze data between passive robot manipulation and shared control could be identified.

Keywords: Human intent-recognition, Eye tracking, Shared-control, Human-robot shared manipulation

Contents

Abbreviations

RW	Real-world
MDP	Markov Decision Process
POMDP	Partially observable Markov Decision Process
DOF	Degrees of freedom
SCARA	Selective Compliance Assembly Robot Arm
IR	Infrared
HSV	Hue, saturation, value

Chapter 1

Introduction

1.1 Background

Human machine interactions play a major part in our daily lives. To improve user experience and efficiency, recent systems succeeded by adapting to their operators. Dialogue systems, autonomous driving and intelligent user interfaces are only few of the emerging technologies that heavily rely on predicting people's intentions, goals or next planned actions. Eye gaze has been proven to be a rich source of information when humans are performing a task. For example, gaze fixations in the scene can reveal how humans perceive a task and pupil size is an important indicator of cognitive load. If these indicators can be factored into a shared-control application, a better collaboration between humans and automated machines/systems could be accomplished.

1.2 Aim

The aim of this thesis is to investigate the potential of eye tracking technology, to help recognizing the intent of humans when working with a machine under shared control. While several studies have been conducted on identifying key objects in a scene depending on directed gaze onto these objects, the literature on eye gaze data regarding human task planning is rather sparse. When interacting with a robot arm in a shared autonomy setting, eye gaze could yield additional information about path planning and action sequencing. While some studies have focused on teleoperation of a robot arm for compensating

motoric disabilities of humans, this thesis focuses on the direct co-manipulation of a robot's end-effector. The experiment setting will include a two degrees-of-freedom (DOF) SCARA robot and the experiment participants will be presented with a path planning task to guide the robot's end-effector. The test subjects will be equipped with an eye tracker to study emerging gaze patterns during the robot manipulation and in a later stage the gaze data will be incorporated into a shared control algorithm, providing a cue on human intention for the planned motion. A survey on the user experience and an investigation of potential differences in the gaze data between the free and supported task execution will be held. In summary, this thesis is dedicated to answer the following research questions:

- How can eye gaze data during a shared manipulation task between a robot and a human subject (path guidance of end-effector) be leveraged to infer the users plan and goal of the task?
- Can eye gaze be integrated into a shared control algorithm to enhance task performance?
- Is there a noticeable difference in gaze data when a user encounters support by the robot system, compared to when the user freely moves the end-effector?
- What are the limitations of the predictions? How notable is the accuracy of the eye tracker when factoring into shared control? Can it be used for identifying fine movements or only to recognize the underlying plan of the interacting subjects?

1.3 Limitations

- **Accuracy of eye tracker:** The accuracy of the subject's gaze location can have a direct impact on the limitations of the predictive algorithm.
- **Diversity and amount of test subjects for data collection:** To guarantee the validity of the experiment and safety for the experiment participants, a research permit has to be applied for. This can take up to several months and thus is a bottleneck for this thesis. Because the research permit can not be obtained in time, the main focus of the thesis is a proof of concept (involving 3-5 human subjects) and the ground work for larger test subject involvement will be laid.
- **Sensory inputs limited to eye gaze and subject's direct influence on robot dynamics:** The experiment is focused on eye tracking and motion input from human subjects and further human-machine interfaces, such as Electroencephalography (EEG) will not be considered. Since the thesis is conducted by one student, the reasons for this limitation are time restriction and the fact that the experiment would require at least two people to simultaneously install the EEG electrodes and to perform the synchronization of the computers.

- **Servo motors of robot:** The servo motors of the robot do not have a feedback signal. The motors are controlled by setting their position with a PWM signal. This made it difficult to assess the impact of the user's forces on the robot dynamics. In addition this also limited the motion planning of the robot, since more complex motion profiles are difficult to realize.
- **Self isolation inhibits availability of test subjects**: During the later stage of the thesis the outbreak of the COVID-19 pandemic heavily limited the number of test subjects that could be involved in the experiment. It is important to reduce the interactions between people and guarantee a minimum interaction distance. The safety-distance during the experiment execution was difficult to obey at all time and thus the amount of test subjects had to be limited as much as possible. This mainly affected the recordings and testing of the shared-control algorithm, which were performed towards the end of the thesis.

Chapter 2

Theory

2.1 Intent recognition

The concept of intent recognition is to infer a human agent's plan and goal based on observed actions. Depending on which user actions are being monitored, several different approaches for intent recognition can be taken. For example in [9] gestures were used to communicate the users intention to an autonomous "servant" robot. Notably they stressed the importance of context and interaction history to distinguish between similar gestures with different goals. They included object recognition in the scene to derive additional context. In [4] video footage of dual-agent interactions (such as handshake, hug, push) was used to study behavioural patterns between humans. By recording the actions of one person, they were able to predict the reaction of the second, unobserved person. They achieved that by developing a novel algorithm based on the principles of maximum causal entropy and inverse optimal control, which will be further elaborated on in 2.2. The advances in autonomous driving also had a major impact on the demand for intent recognition. Autonomous driving is a multi-agent problem, where an intelligent system constantly has to predict the actions of its surrounding environment. Autonomous vehicles have to consider different types of human agents when planning a route. In [10] they divided the human agents into three main categories: Humans in the vehicle cabin of the autonomous vehicle, humans around the vehicle and humans in surrounding vehicles. When planning trajectories of surrounding vehicles, a great deal of driving intent can be extracted from the road layout and by identifying the lanes being chosen [5]. In practice, a partially observable Markov decision process (POMDP) is often used as an underlying framework to model intent recognition. Partially observable refers to the fact that the data an agent receives about its environment and interaction partner(s) is incomplete and often stochastic interference is present in the process. Heuristic methods such as probability distributions over the set of possible states

can be used to deal with information incompleteness. Another common method to predict user intent are neural networks, but their black-box nature make reasoning about the produced outputs hard and thus the focus of this thesis lies on more interpretable models.

2.2 Shared control

The main difference of shared control, compared to traditional control systems, is the integration of the user into the control loop. Shared control aims to merge an automated system with a user to reach a common goal in a safe and collaborative manner. Shared control is heavily reliant on intent recognition. It is important to model the uncertainty arising from interactions with humans. As shown in [8], humans are rarely rational decision makers. By integrating the more risk oriented nature of humans into the model, improvements to predicting human actions can be made. A relevant framework to account for the stochastic nature of the human influence on the control loop is a Markov decision process (MDP). More precisely, the user's effect on the control task is included in the state transition function. The state-of-the-art algorithms developed in [14],[4] use inverse optimal control (inverse reinforcement learning) to recover an unknown reward function of an MDP. This recovery is based on samples from the behavior of the human-robot interaction, usually in the form of a probability distribution. In [3], they developed a concept called policy blending for shared control. Their algorithm considers two policies, the user's input and the robot's prediction of the user's intent. Depending on the confidence level of the intent policy, the robot applies weaker or stronger corrections to the user's input to reach the predicted goal.

2.3 Eye tracking / Hand-eye coordination

Eye tracking refers to the concept of recording eye metrics and movements and mapping them onto a scene view, which usually represents the field of view of the user. The recording of the eye is typically done with an infra-red (IR) camera. The image obtained by the IR camera is then processed by an algorithm to obtain pupil position and eye angle with regard to a reference frame, as well as pupil diameter. A calibration procedure is then applied to retrieve the gaze location in the scene view. Since the eye parameter detection algorithm and the scene view mapping varies between the different eye tracking manufacturers, we limit the explanation to the device used in this thesis, which is a head mounted device from Pupil Labs [12]. The device explanation is given in chapter 3. The gaze data can then be further processed to analyze formats such as scan-paths (for temporal information) and heat-maps (to identify areas of interest), to study correlations and implications on a task. For example eye, head and hand movements coordination has been studied in [11], where subjects performed a mechanical building task with LEGO blocks. The

recorded eye gaze when building simple block patterns is ahead of the motoric execution and this latency is depending on the building strategies of the different subjects, as well as the complexity of the task. Another important measure in eye tracking is pupillometry, which describes the size and shape of the pupil. In [6] it was shown that the rate of change of pupil diameter directly correlates to cognitive load and task difficulty. When subjects were presented with a digit memorization task, the pupil diameter tended to dilate during mental processing of the string and constrict while they were reporting the string. The more digits the participants had to memorize, the more significant was the rate of change of pupil diameter.

Chapter 3

Experiment design and setup

3.1 Hardware

The experiment setup consists of a two-degrees-of-freedom SCARA robot, which operates on a 2D plane. A top mounted camera records the position of a color marker placed on the end-effector. A total of 12 LED's are located on the 2D operating plane of the robot.

3.1.1 Eye tracking device

The human subjects of the experiment were equipped with an eye tracker, consisting of an infrared (IR) eye camera, recording the eye movements of the subjects at 120Hz, and a scene view camera, recording the field of view of the subjects at 30Hz. The eye tracker used throughout the experiments is the Pupil Core device manufactured by Pupil Labs [12]. The pupil detection algorithm attempts to find a 2D ellipse in the IR eye camera image that represents the pupil geometry. To do so, a series of image processing methods are applied to filter for the dark pupil. To map the detected pupil positions of the eye camera to the scene view, a calibration process needs to be performed. The result of the calibration routine is a transfer function consisting of two bivariate polynomial. During calibration, the degree of the polynomials are determined by making the user focus on markers in the scene view [7]. Using the surface tracking plugin provided by Pupil Labs open source software, the LED surface of the robot's operating plane was

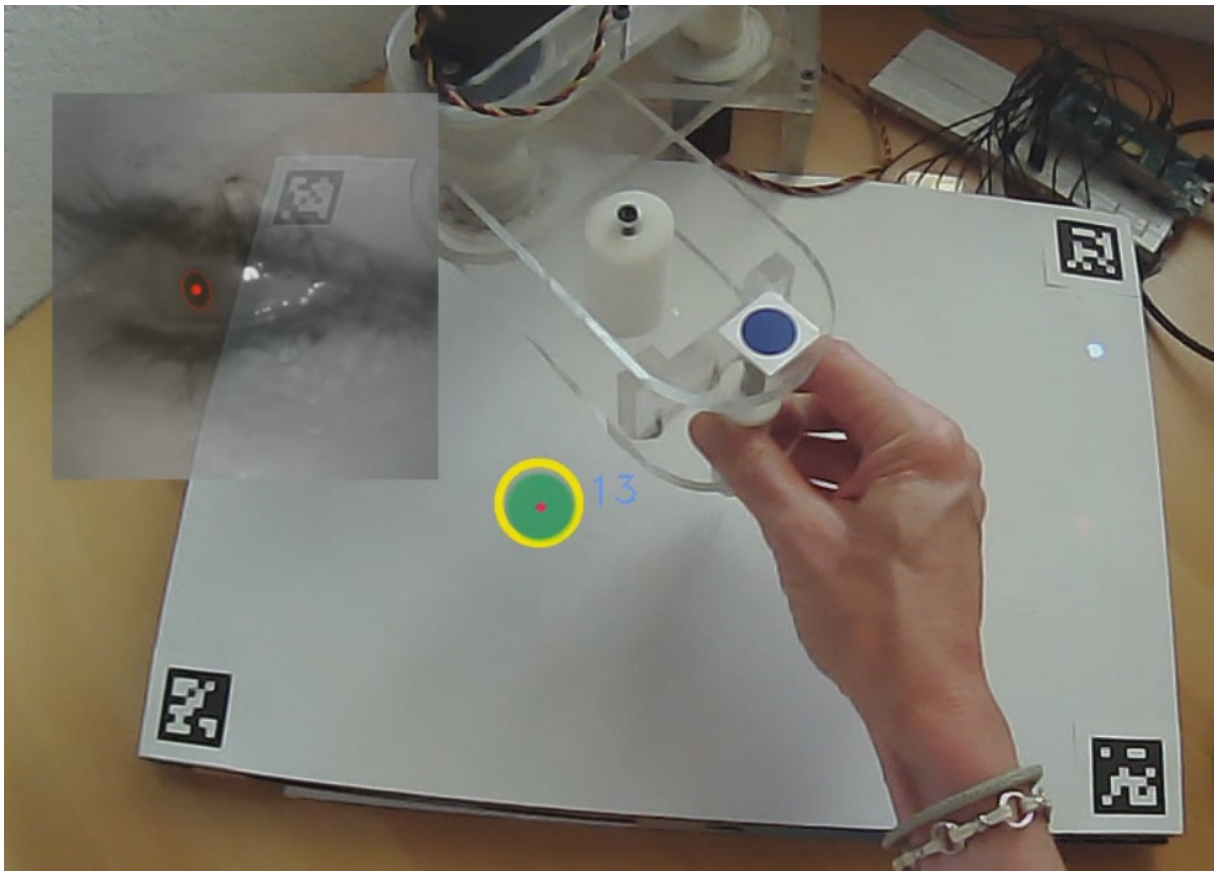

Figure 3.1: Experiment setup

isolated with the help of four AprilTag markers [1]. This allowed to limit the subjects gaze points to the one's located on the robot's operating plane.

3.1.2 SCARA robot

The two joint axes of the robot (shoulder and elbow) are driven by two hobby servos operating under 5V. Their angular range is up to 180 degrees. The control signal of the motors is in the pulse-width modulation (PWM) form and the effective voltage of the signal corresponds to an angle between 0 and 180 degrees. Attached to the robot's end-effector is a color marker, which is detected by a top mounted camera. After obtaining the end-effector position, the angles of the joint motors can be calculated with the robot's inverse kinematics.

3.1.3 LED surface

To obtain a common reference frame for both the robot arm and gaze positions, the LED surface is defined with four AprilTag markers representing the corners. The same four markers are detected by both cameras - the scene view camera of the eye tracking device and the top view camera of the robot arm.

3.2 Experiment design

The LED surface consists of the start-location (blue LED on the right-hand-side of the participant) and the end-location (blue LED on the left-hand-side of the participant). Between the start- and end-location are a total of 10 LED's from which five (randomly chosen) will light up during each experiment trial. The participant's goal is to guide the robot's end-effector from the start LED to the end LED while connecting all the light up LED's. The order of the inter-connecting LED's can be chosen freely by the participants but with some regard to the shortest path. This condition should reflect the fact that time is a somewhat critical factor to task performance. The experiment can be divided into two parts. First the participants execute the task without robot assistance and in a later stage with robot assistance (shared control).

1. **Without robot assistance:** The focus lies on collecting participants gaze data while executing the manipulation task. In addition, the position of the robot's end-effector will be recorded at all times.
2. **Shared control:** The second part of the experiment aims to evaluate the performance of the shared control algorithm while performing the same manipulation task. This evaluation is done by the participants and focuses on the following criteria:
    - How intuitive or counter-intuitive does the robot's assistance feel during task execution?
    - Could an increase in task performance be achieved?
    - Does the confidence threshold apply correctly to the robot support?
    - Does the subjects gaze data differ between shared control manipulation and without robot assistance?

Chapter 4

Data collection

4.1 Experiment participants

A total of 5 people took part in the experiment. The participants parameters are summarized in table 4.1:

Table 4.1: Participants parameters

Nbr.	Age	Gender	Vision impairment	experiment runs with valid data
1	29	Male	No	0/5
2	36	Female	No	0/5
3	25	Male	No	4/5
4	27	Female	No	5/5
5	31	Male	No	4/5

Since the accuracy of the eye tracking device heavily depends on parameters such as eye shape, eye lashes and calibration accuracy, the quality of gaze data varied among subjects. Also for subject 2 the surface detection was incomplete, due to a large portion of frames not having all four markers in the scene (i.e. not the whole surface was visible on some frames). To guarantee a meaningful data analysis, data with poor quality (all data of subjects 1 and 2, as well as one experiment run each for subjects 3 and 4) was not included in the evaluation.

4.2 Eye tracking data

The eye tracking data recorded from the participants consists of pupil diameter, blink frequency and the gaze position on the LED surface. The gaze position, which is mapped to the LED surface by pupil lab's open source software, can be further divided into the following subcategories [2]:

- **Fixations:** If a participant's gaze is resting on a single location for a prolonged time, it is considered a fixation. Based on the task, the duration of a fixation is typically between 100-500ms, which are the limits applied to pupil labs software throughout this thesis. Fixations above 500ms are split into multiple fixations by the software, but are labeled with the same fixation index.
- **Saccades:** Saccades are rapid, jerking movements of the eye, usually occurring between fixations. They last between 20-200ms and the amplitude can have a narrow or wide range, depending on the distance between two fixations. Saccades can be an involuntary and show up even during fixations.
- **Smooth pursuit:** Allows the eyes to slowly track a moving target by adjusting the eye's angular velocity to the target's angular velocity. This type of eye movement is voluntary by the observer. Only practiced people are able to make smooth pursuit movements without a moving stimulus.
- **Vestibulo-ocular movements:** When focusing on a scene, vestibulo-ocular movements counteract head movements, so that the visual image remains stable and is not slipping. Head-mounted eye tracking devices perceive these eye movements as identical to smooth pursuit movements and can only be differentiated by recording head movements. Since the pupil labs eye tracking device is not recording head movements, a head rest was used to minimize vestibulo-ocular movements.

The pupil lab software assigns a detection confidence value (between 0 and 1) to each gaze measurement taken. This confidence value can be used to filter out gaze data with potentially low accuracy. The manufacturer recommends to only use gaze data with above 0.6 confidence, which is the filter criteria used throughout this thesis.

The fixation detection algorithm implemented by Pupil Lab's software uses a dispersion-based method to identify the fixations. This means fixations are identified as groups of consecutive points in the scene view within a particular dispersion, or maximum separation. The number of consecutive points considered is dependent on the minimum fixation duration, in this case 100ms. The dispersion of these consecutive points is defined as [13]:

$$D = [max(x) - min(x)] + [max(y) - min(y)] \tag{4.1}$$

While the definition of gaze dispersion in [13] is calculated with consecutive x- and y-positions in the scene view, Pupil Lab's software is calculating the dispersion by identifying the maximum angle between all eye vectors recorded during the fixation time window. If this maximum eye angle is below a chosen threshold, a fixation is detected.

4.3 End-effector position

The end-effector's real-world position was recorded during each trial at 100Hz. The timestamp format used for the end-effector recording is the UNIX epoch time of the computer on which the data collection was performed. Pupil Lab's software has its own time scale for the collected gaze data, which allows synchronization to the UNIX epoch time. This guaranteed a valid comparison between the data.

4.4 Experiment time constraints

The experiment participants were not subject to any time constraints apart from a start countdown. After the start countdown, which also started gaze and position recordings, the participants were given no completion time constraint. The task completion time averaged around 10 seconds and recordings were stopped after reaching the goal LED.

Chapter 5

Data analysis

5.1 Preprocessing

The fixation coordinates on the LED surface are given as normalized coordinates with respect to a different point of origin than the end-effector coordinates. Thus, the fixations are mapped to the real-world coordinate system of the robot action plane. The mapping is shown in fig. 5.1

$$X_{RW} = (0.5 - X_N)X_S \tag{5.1}$$

$$Y_{RW} = Y_O + (1 - Y_N)Y_S \tag{5.2}$$

X_{RW}	X RW coordinates	
Y_{RW}	Y RW coordinates	
X_N	X gaze normalized	
Y_N	Y gaze normalized	
X_S	RW length of LED surface	404 mm
Y_S	RW height of LED surface	280 mm
Y_O	Y offset of Robot origin from LED surface	43 mm

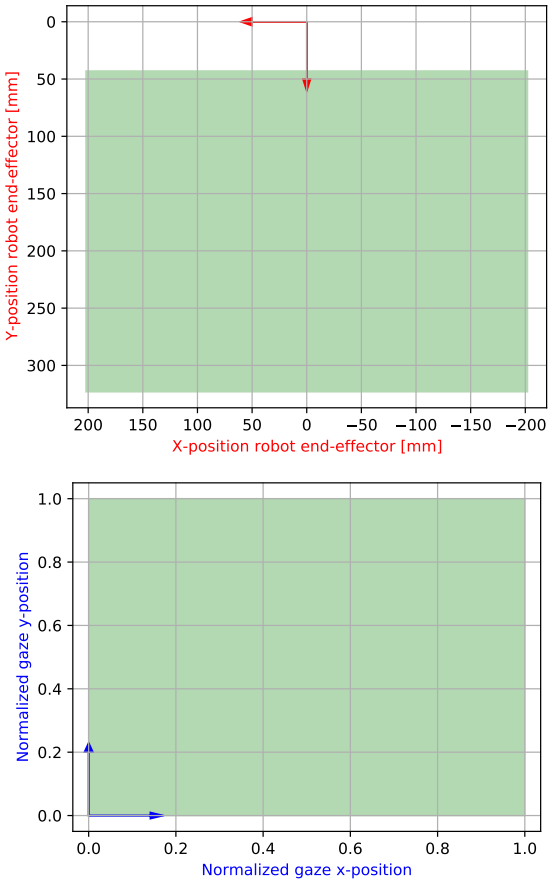

Figure 5.1: Surface mapping of gaze coordinates to robot RW coordinates

5.2 End effector position

For each trial the path of the end-effector, as well as the LED's specific to each trial, was visualized as in fig. 5.2:

The end-effector position is determined with an accuracy of approximately 5mm and as can be seen in fig. 5.2, the test subject guided the end-effector through the LED sequence within roughly 10mm proximity. Some test subjects were more precise than others, but all test subjects were usually within the 10mm, apart from some out-liners of single positions. For example, in fig. 5.2 the goal LED (176,148) was not ended on very precisely.

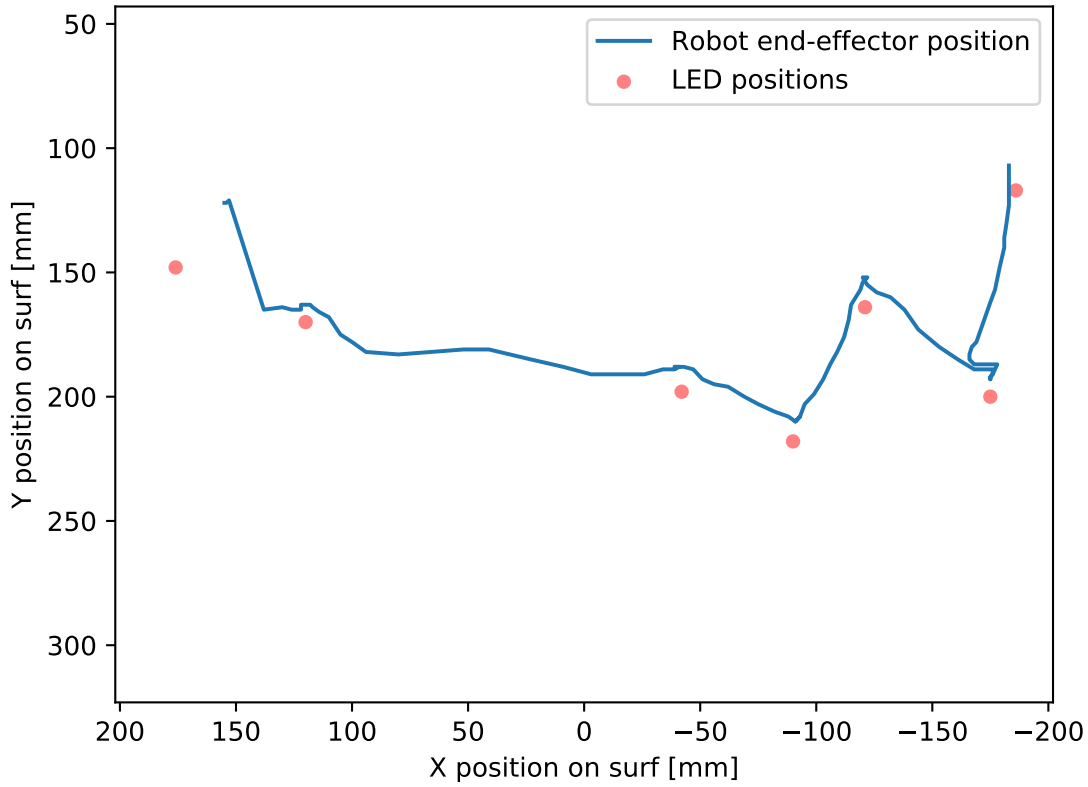

Figure 5.2: End-effector and LED positions in RW coordinates, subject 4

5.3 Fixations on surface

In the top half of fig. 5.3, the average position during each fixation span is plotted in relation to the LED positions. The size of the fixation markers corresponds to the fixation duration. The bottom half shows exact fixation duration of each successive fixation. The positional accuracy of the fixations on the surface is dependent on the surface detection, as well as the gaze mapping precision of pupil lab's software. Since in some frames of the scene view camera of the eye tracker, one or more of the four surface markers might be obscured by either the test subjects arm or the robot arm, the surface can not be fully detected. In those cases, the surface detection algorithm calculates the missing corners with the given side ratios of the LED surface. While in most of these cases only one marker is obscured at a time, which still yields accurate surface positions, the accuracy drops significantly with only two or fewer detected markers. These trials, containing portions of the video only showing two or less detected markers, were discarded. The gaze mapping accuracy is mainly dependent on the calibration and can vary dependent on the position on the surface, e.g. gaze positions closer to the boundaries of the surface tend to be less accurate due to squinting of the eyes. Only eye gaze data with positional accuracy below 20mm was evaluated.

As can be seen in the upper part of fig. 5.3, most fixations fall approximately onto the LED positions. If compared with the end-effector position in fig. 5.2, the fixations are very close to the position where the end-effector passes the LED position. Notable exceptions are the first four fixations at the start. As we will see in section 5.6, some of these fixations are actually smooth pursuit eye movements.

5.4 Fixations vs end effector

In fig. 5.4, the fixations and end-effector positions were plotted for the total trial duration to illustrate timing differences between arm and gaze coordination. The subject's gaze tends to fixate the next LED between one and two seconds before reaching the position with the robot end-effector, depending on how much distance the subject has to cover when moving from the current LED to the next. After reaching the fixated position, the gaze is shifted almost immediately (with 0.1-0.2s delay) onto the next LED, while movement onset is delayed about 0.5 seconds.

5.5 Pupil diameter during task execution

The pupil diameter during task execution is measured in pixels, which is a valid measure, since the relative pupil diameter is the information we are interested in. As can be seen in fig. 5.6, pupil diameter is largest at the start of the experiment and gradually decreases with time. This indicates increased mental load of the subject at the start, i.e. during path planning. 7.3s into the experiment run the pupil dilates again for approximately 0.7s. The relative change in pupil diameter is roughly 22% .Zero pupil diameter occurred when the subjects blinked.

5.6 Fixation dispersion

When comparing all gaze points on the LED surface with the fixations, it shows that some of the fixations could rather be considered as smooth pursuit eye movements. This makes sense, since slow smooth pursuit movements have a small enough dispersion to be considered a fixation. This effect is also reflected in the position of the gaze points. In fig. 5.7 this effect is apparent in the fixations 1-3. Usually smooth

pursuit eye movements are hinted at by a short fixation duration. During smooth pursuit eye movements the dispersion increases until it surpasses the dispersion threshold used to detect a fixation.

5.7 Fixations comparison between test subjects

The fixation durations are compared between subjects 3-5 and split into fixations lying within 20mm distance of an LED and fixations further away than 20mm from any LED positions. The median of the fixation durations close to an LED position is on average 40ms longer than the median of those not close to an LED position. The maximum fixation duration is limited to 0.5s and the minimum to 0.1s, by the Pupil Lab software. No clear distinction can be made for the maxima and minima of the fixation durations, because all three subjects cover the whole spectrum of fixation durations between maximum and minimum. Notable for subject 3 is that the fixations not falling onto an LED position are more spread out, resulting in a longer box and a shorter lower whisker. The number of fixations used from each subject are summarized in table 5.1.

Table 5.1: Nbr. of fixations used for box plot

Subject nbr.	On LED	Nbr. of fixations	total nbr. of trial
3	Yes	79	4
3	No	17	4
4	Yes	103	5
4	No	24	5
5	Yes	85	4
5	No	16	4

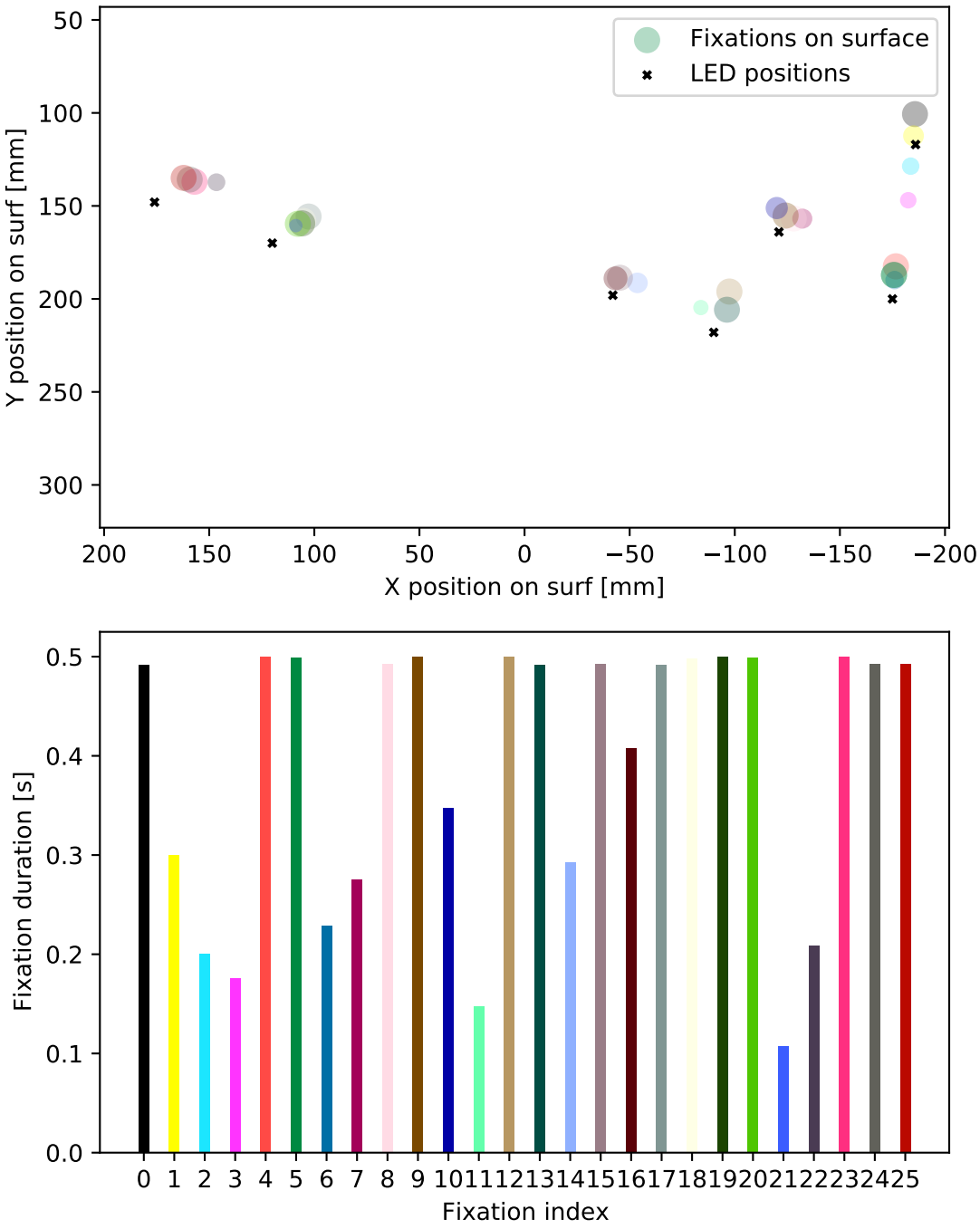

Figure 5.3: Average positions of fixations on surface, subject 4

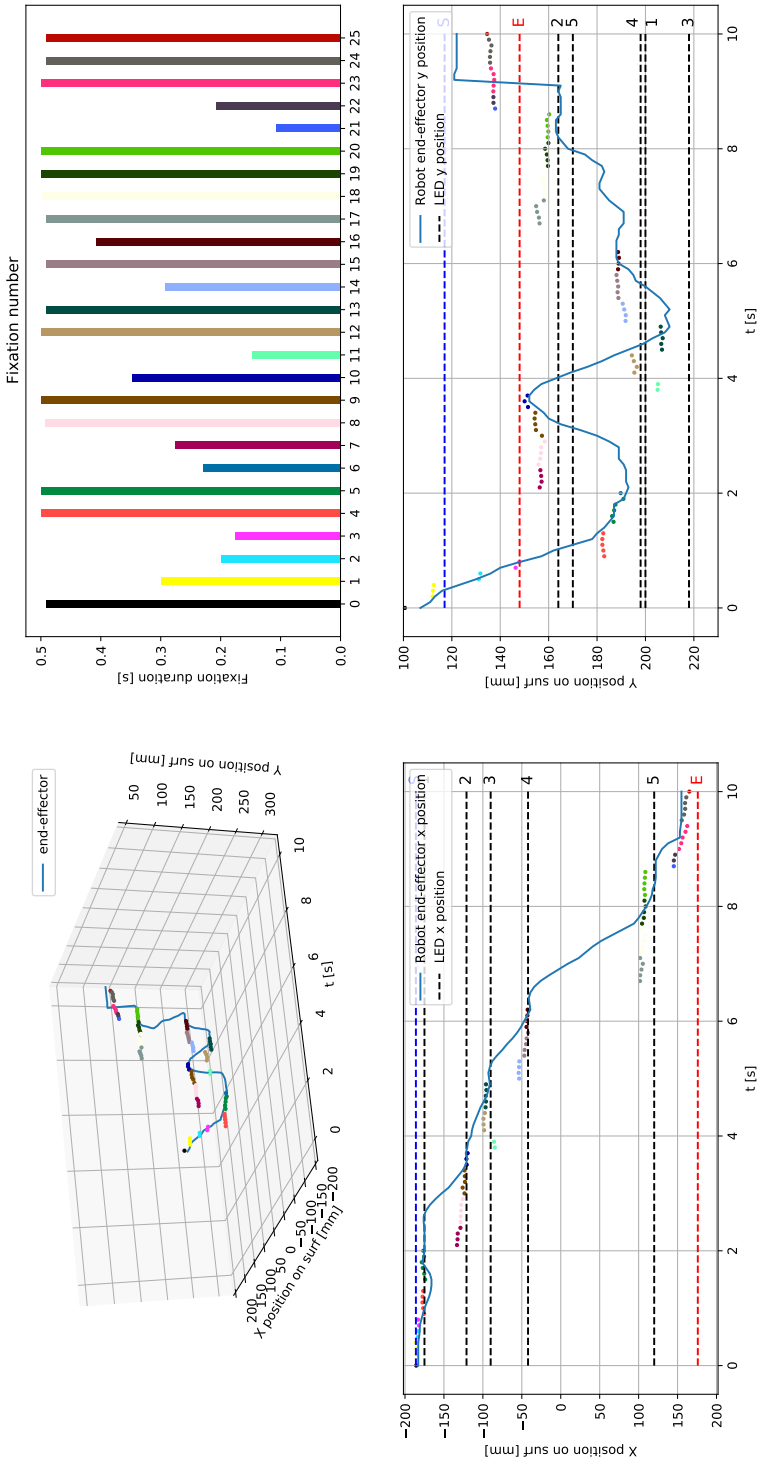

Figure 5.4: Fixations and end-effector through time, subject 4

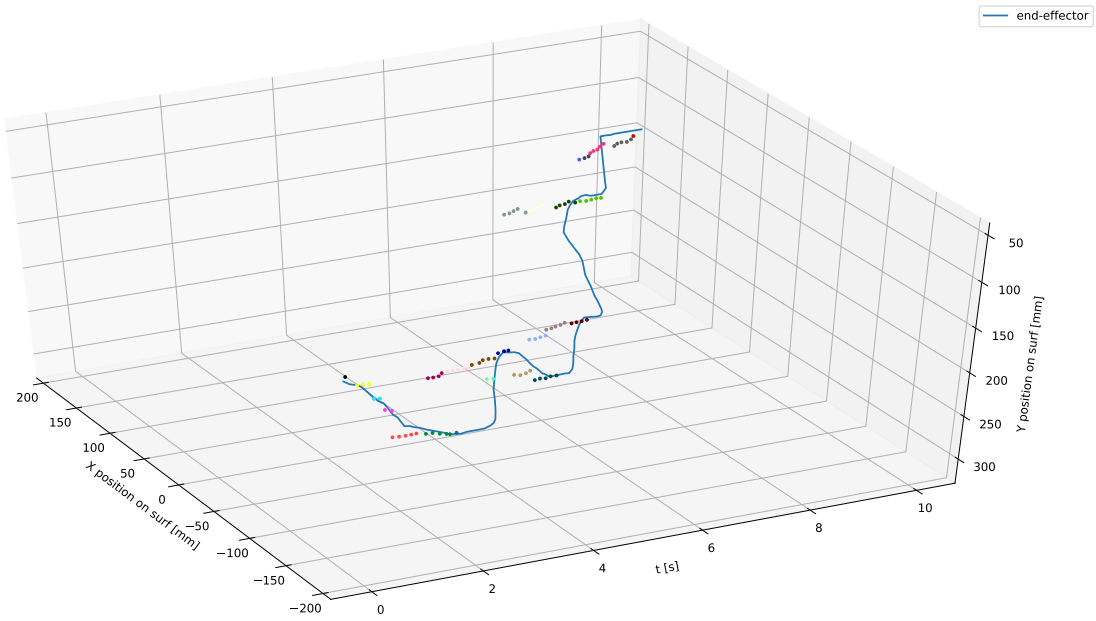

Figure 5.5: 3D plot, fixations on surface through time, subject 4

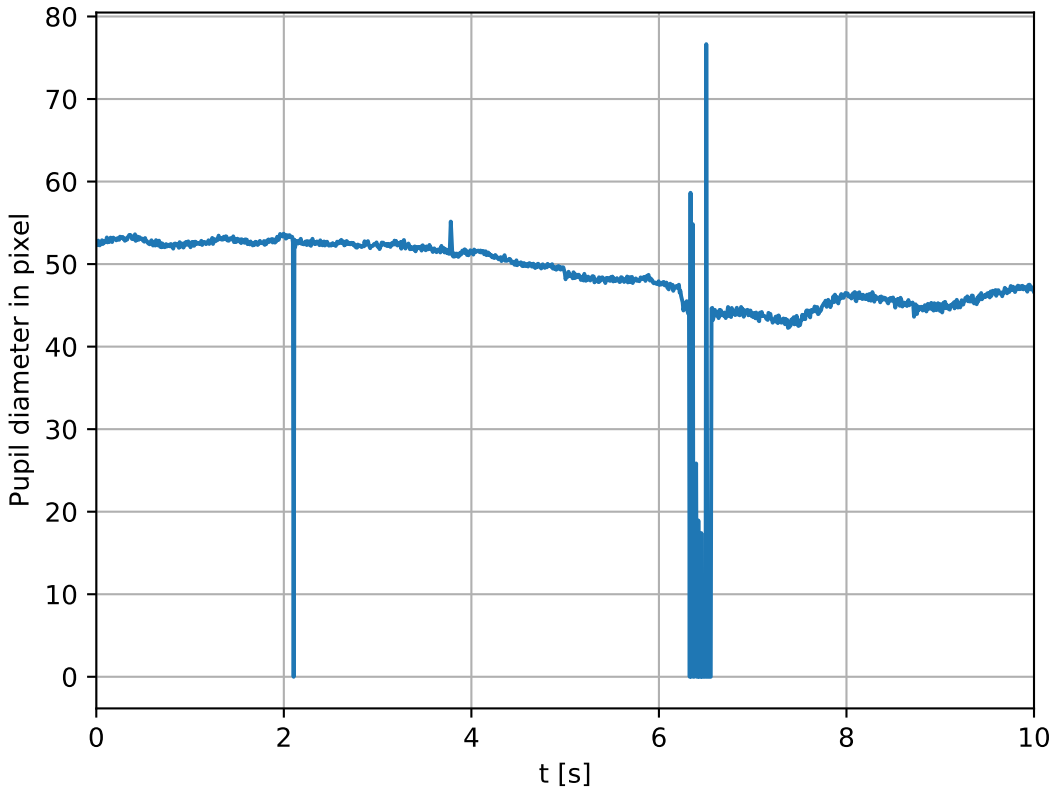

Figure 5.6: Pupil diameter during task execution, subject 4

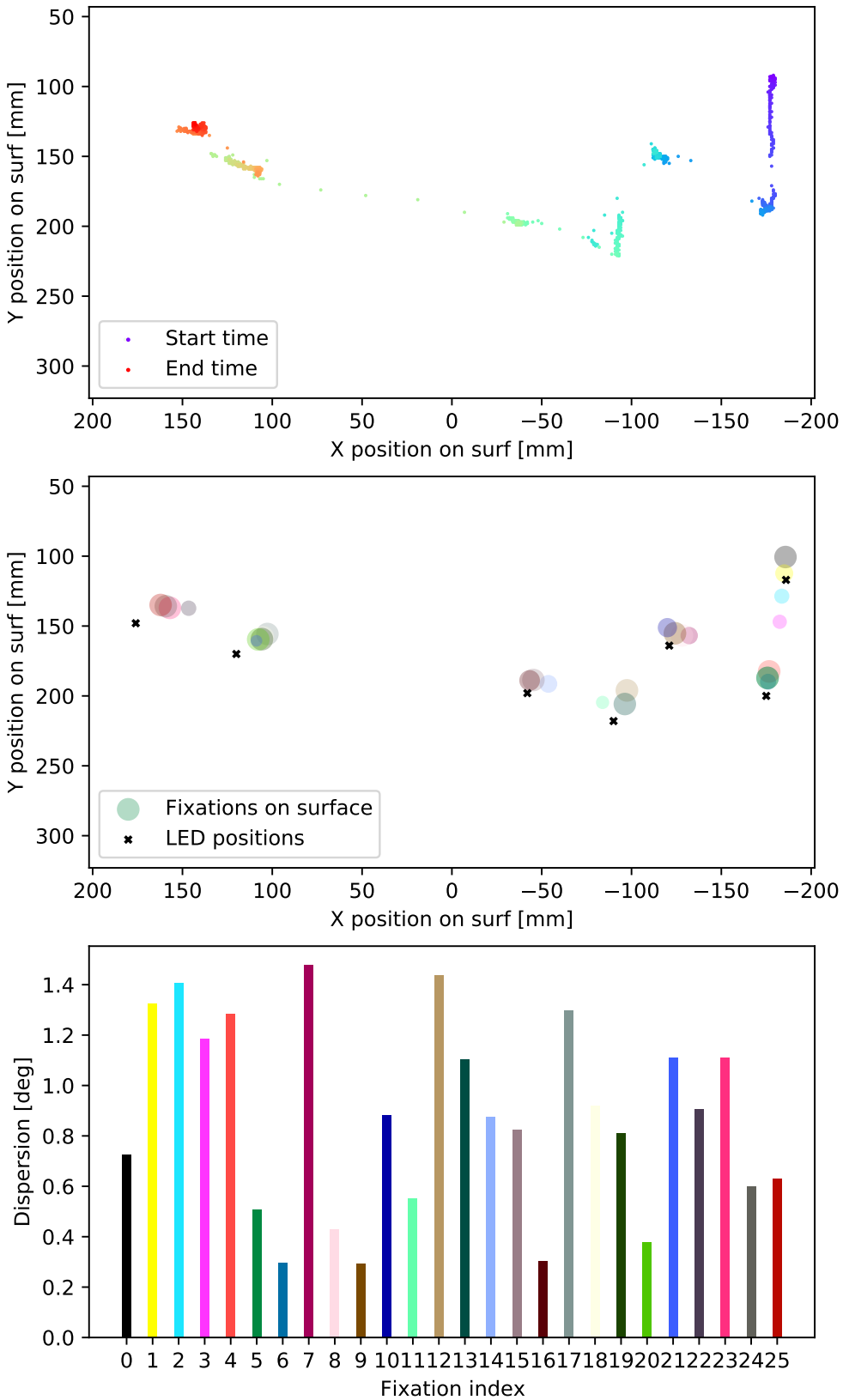

Figure 5.7: All gaze points on surface, fixations on surface and dispersion of the fixations for subject 4. The color spectrum for the top graph changes with experiment time and for the bottom two graphs each color represents a fixation.

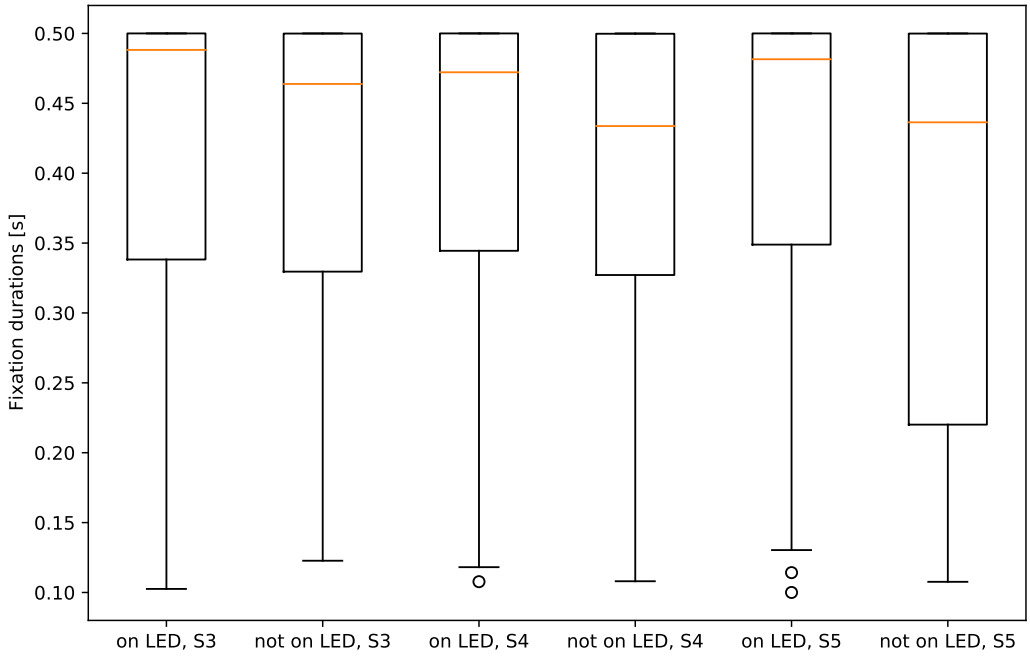

Figure 5.8: Fixation durations of all trials of all subjects

Chapter 6

Predictive algorithm design

This chapter explains the design choices made to develop the algorithm, as well as its functionalities. To start off, the goal of the algorithm is stated, followed by a discussion on how the findings of the passive robot experiment from chapter 5 can potentially be integrated in the algorithm to make predictions about how the users intend to complete the experiment task. After that, the input parameters of the algorithm, i.e. what states and observations during experiment execution does the algorithm have access to, are explained. In the next step the limitations of the algorithm are listed and finally the implementation and structure is explained.

6.1 Goal of algorithm

The overlaying goal of the algorithm is to exploit the subjects gaze behavior during task execution, to derive which LED's they are intending to reach and how they plan to sequence through them. If the algorithm can deduce the position the subject is trying to steer the end-effector to, the next step will be to assist the subject to reach that position.

6.2 Information used from passive robot data analysis relevant to algorithm

With the results obtained in chapter 5, where the subjects goal was to guide the passive (unactuated) robot end-effector through seven trial specific LEDs, the following observations were investigated for "informational use" to the algorithm:

- **Positional precision of subjects end-effector guidance:** While guiding the robot's end-effector through the LED positions, the subjects considered an LED as visited by passing them within a 10mm radius.
- **Fixations proximity to LED positions:** To decide if the subject's gaze is fixated on an LED position, a proximity region has to be chosen for each LED on the surface. A simple, but nonetheless efficient heuristic is to define a circular area with fixed radius around each LED position. The length of the radius is dependent on how accurate the subject's gaze is mapped onto the LED surface. To achieve optimal classification the radius should be as small as possible, but still large enough to include less accurate fixations on LED's. A radius of 20mm proved to yield the best results. Since fixations tend to get less accurate the closer they are to the boundaries of the surface, the length of the radius was chosen in accordance with this observation.
- **Most fixations fall onto LED positions during task execution:** Referring to fig. 5.8 of the previous chapter, approximately 4.5 as much fixations fall onto one of the trial specific LEDs compared to those that were not in any LED proximity areas.
- **Duration differences between the fixations falling onto LED positions and those not in proximity of LED positions:** While it was shown in fig. 5.8 that the median fixation duration was longer for those falling onto an LED position compared to those that were not in any of the LED's proximity areas, both categories had fixation durations ranging from 100-500ms. Because the fixations have to be processed by the algorithm during run-time, i.e. each fixation is processed individually, no apparent distinction can be made with only the duration.
- **Fixation duration and dispersion:** When considering the fixation dispersion in addition to the duration, it was shown that Pupil Lab's software classifies slow smooth pursuit eye movements as fixations. The key to identifying these false classifications is to look at the ratio between the duration and dispersion. A fixation with a short duration and high dispersion is most likely a smooth pursuit eye movement.
- **Timing differences between fixations and movement onset:** As discussed in the previous chapter 5, when subjects reach a goal LED, they tend to shift their gaze onto the next LED after a short delay of 0.1-0.2s, while movement onset of the end-effector is usually delayed between 0.3-1s.

- **Pupil diameter during task execution:** The subjects increased mental load at the start of the experiment, likely due to path planning, correlates with a wider pupil diameter. Although this information is in accordance with the theory, little value can be gained by incorporating it into the algorithm and the computational expense should be invested into more promising information extraction.

6.3 Input parameters

- **All LED positions on surface:** The locations of all LEDs on the surface are known to the algorithm. It is important to note that the five randomly chosen LEDs specific to each new experiment run are unknown to the algorithm and neither is the order in which subjects choose to pass them.
- **Current end-effector position:** The current end-effector position detected by the top mounted camera is updated with 100Hz.
- **Fixation position, duration and dispersion:** Fixation position on the LED surface and the corresponding duration and dispersion are broadcast to the algorithm.

6.4 Limitations of algorithm

- Since the servo motors of the robot arm, when activated, are locked in place at the current position, the robot has either full control of the motion (when activated) or no control (in passive configuration). This limitation did not allow for a complex shared control algorithm, where the motors could exert varying torque depending on the goal detection confidence.
- It is not possible for the robot to recognize when the user is not agreeing with the robot's chosen path to a detected goal. This is a result of the lack of force feedback from the servo motors.

6.5 Algorithm structure

The structure of the algorithm and its main functionalities are displayed in fig. 6.1.

The algorithm consists of three processes running in parallel:

- **Receive gaze data and detect subject's goal:** This process receives the user's fixations on the surface, which are sent by the Pupil Lab software. The fixation data consists of the normalized position on the surface, as well as the duration and dispersion of the fixation. To exclude potential smooth pursuit eye movements labeled as fixations, the ratio between dispersion and duration is calculated. If that ratio is above a threshold, the fixation could be a smooth pursuit eye movement and thus is not considered as a potential goal. If the dispersion-duration-ratio is below the threshold, the normalized position on the surface is then mapped to the RW coordinates of the robot action plane. If the RW position of the fixation is in the proximity area of an LED, it is added as a goal to the goal queue.

- **Detect end-effector position and calculate RW coordinates:** To obtain the current end-effector position, we first need to grab the current camera frame of the top mounted camera and detect the color marker placed on the end-effector. This is done converting the image into HSV (hue, saturation, value) space, followed by image thresholding to isolate the marker from its surrounding environment. The resulting image is then searched for connected components (i.e. shapes). To segment the marker from other detected shapes, each detected shape is checked for its area and if the size of the area is corresponding to the one of the marker. After obtaining the pixel coordinates of the marker in the image, the pixel coordinates are then mapped to the RW coordinates of the robot action plane. This is done by an affine-transformation approach with the four AprilTag markers (also detected in the image) on the corners of the surface. This works because the AprilTag markers are fixed in place and their RW positions are known, thus the affine-transformation makes use of where the color marker is in relation to the AprilTag markers. Since the color marker on top of the end-effector is offset from the LED surface, a parallax error occurs when moving away from the projected center (onto the LED surface) of the top mounted camera. To correct the parallax-error a automatic calibration routine is executed at the start of each experiment run.

- **Move robot to current goal position:** As soon as a goal is added to the goal queue by the goal detection process, the current goal is set to the first element in the queue. Since the robot's servo motors are in passive mode (no Voltage applied to the servos) until there is a current goal, the motors need to get attached at the current angles. Because the current position detected by the previous process are the x-y-coordinates of the end-effector, the current joint angles need to be calculated. This is done with the robot's inverse kinematics. After the current joint angles are obtained and the servo motors are attached, the robot begins to move towards the current goal. After reaching the current goal, the current goal is removed from the goal queue. If in the meantime a new goal has been detected and added to the queue, the current goal is set to the newly detected goal. If the goal queue is empty instead, the servo motors get detached and the process is waiting for a new goal to be added to the queue.

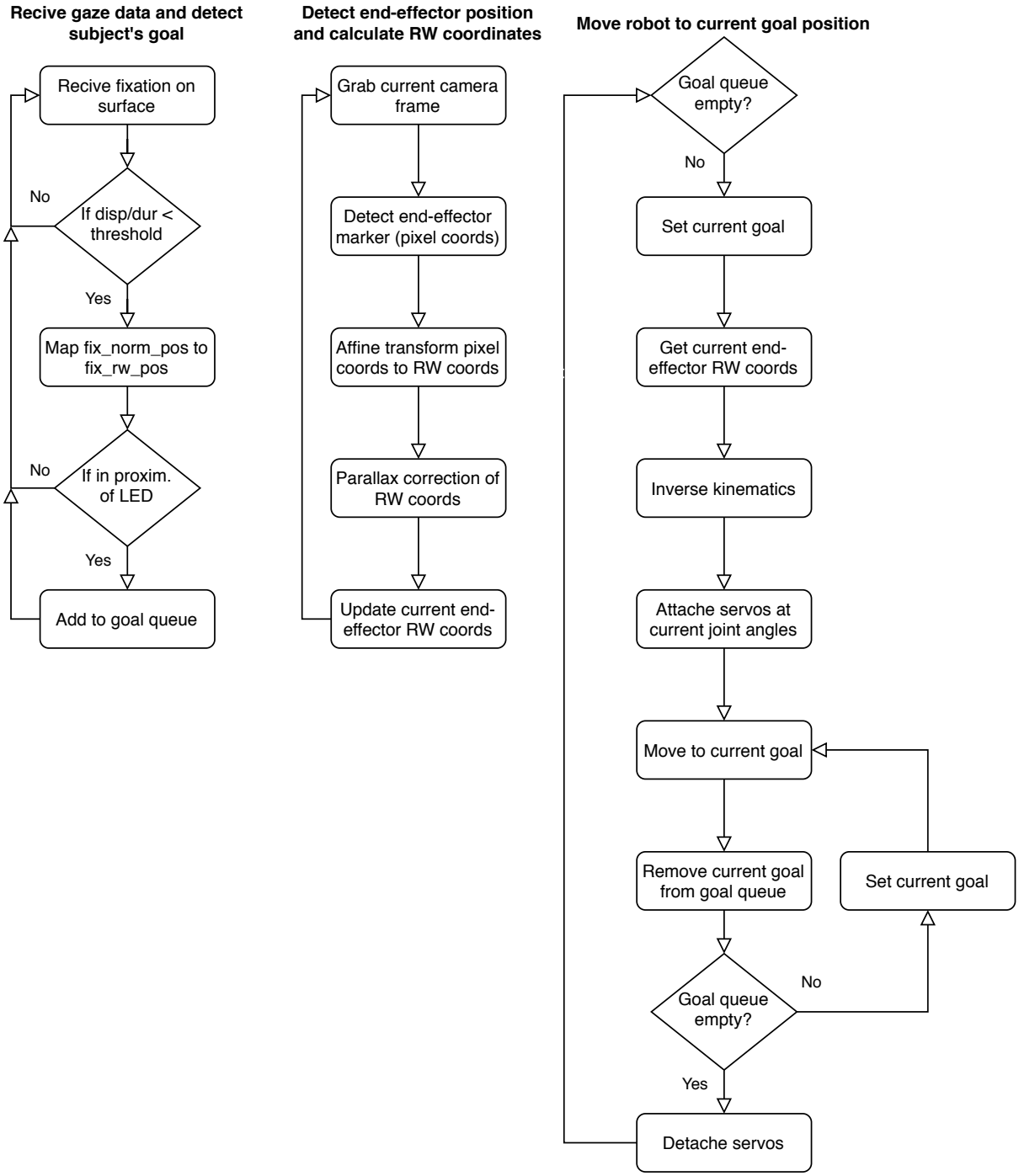

Figure 6.1: Basic structure of algorithm explaining the three main processes

Chapter 7

Results

7.1 Evaluation of Algorithm

7.1.1 Recorded data during shared control experiment

To test the algorithms performance and the effect of the active robot on the subject during co-manipulation, the RW position of the end-effector and the subjects gaze data was recorded. All recorded trials had to be made with the same subject (subject 4), since the self-isolation period of the COVID-19 virus had started. The recorded data is visualized in fig. 7.1.

Even though the first fixation lies inside the proximity area of LED 1, the algorithm does not consider it as a goal. This is because of its high dispersion, which indicates it might be the end of a smooth pursuit eye movement. Approximately 0.3s before the subject starts moving from LED 1 towards LED 2, the subjects fixates at LED 2 with low dispersion. This results in a quick response from the algorithm, identifying it as a goal LED. The robot activates almost immediately after the subject begins moving towards LED 2. Since the subject's gaze keeps fixating on LED 2 for about 0.7s after the active robot reaches it, the next goal can not yet be identified (goal queue empty) and the algorithm detaches the servo motors of the robot to hand the control back to the subject. Although the subject's gaze switches to LED 3, before moving towards it, LED 3 is detected as a goal only halfway through the motion. This is because the subject starts moving the end-effector almost at the same time as the gaze is redirected at LED 3, which requires the corresponding fixation (number 7) to be long enough to be considered a potential goal by the

algorithm. Similarly, when the subject moves the end-effector from LED 3 to 4 and from LED 4 to 5, the robot aids the subject only about 0.2s into the movement towards the goal LED. Surprisingly, the subject starts fixating on the end LED about 0.2s into the motion of moving towards it and not as with previous goals before starting the motion. Since start and end LEDs are the same in each trial, this could indicate that the subject has acquired muscle memory of the end LED position. Because the fixation occurred late, the algorithm did detect the goal late too. In conclusion, if the dispersion of the fixation is low enough, the algorithm manages to identify the goal after the subject fixates on it for at least 0.3s, which is a minimum delay for the algorithm's goal identification. How far ahead of time the robot can predict the subject's next goal is mainly dependent on how much ahead the fixation occurs before the initiation of the end-effector movement. In theory, if the subject fixates on a goal LED 0.3s before initiating the movement towards it, the robot can perform the whole movement between the LEDs. This was almost the case between LEDs 1 and 2. A problem that came up during one of the trials was that low calibration accuracy of the eye tracker resulted in a fixation lying close to an inactive LED (that was not in the sequence of the trial). As a consequence, the algorithm considered it as a goal LED. However, this error can be avoided completely when making sure the eye tracker is well calibrated before each trial.

7.1.2 Comparison between passive experiment and shared control

When comparing between the passive robot and shared control experiments no apparent difference in gaze data could be detected. However this could be due to the limited data that could be collected. In addition the subjects were not informed about the goal queue of the algorithm, which could potentially be used to increase task performance, if the subjects trust the algorithm's goal detection enough. While in the passive experiment the users in some cases were not very accurate in landing on the exact LED positions, the active robot had a higher accuracy and could make up for this.

7.2 Subjects evaluation of algorithm

The subject was asked to evaluate the algorithm on the following criteria:

- **How intuitive is the robots support during task execution, do the subjects feel the robot is supporting them during their task execution, or are they irritated by the support?** During first test runs, the subject was taken by surprise when servo motors were attached, because it was

notable and interrupted the users flow of motion. However the subject got used to this quickly, after the first two experiment runs.

- **Do the subjects think they could increase their task performance with robot assistance?** Once the motors are active and move towards the correct goal, the subject could start planning the next goal(s), which could result in less time spend on the actual goal positions. They argued however, that it usually took them a short duration to recognize if the robot is actually moving to the correct position. This delayed their planning for the next goal slightly.
- **What do they think could improve the algorithm?** The subjects opinion was, that if the robot would gradually increase its assistance (when recognizing a goal), it would feel a lot more natural and new subjects, that are not used to the abrupt activation of the motors, would be less surprised by it. This however would require to install new servo motors that allow torque control.

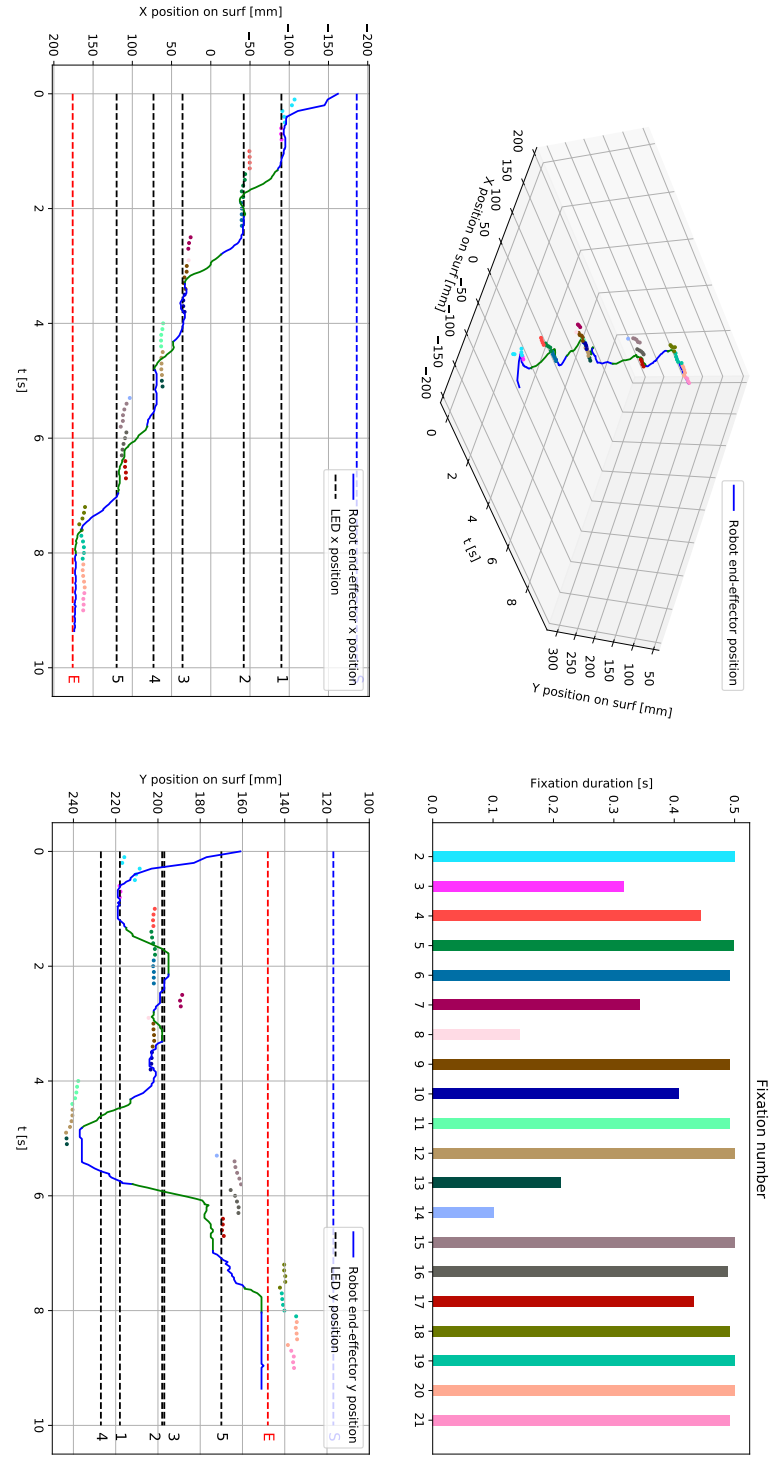

Figure 7.1: Recorded data of subject 4 during shared control, blue end-effector coordinates = passive robot and green = active robot

Chapter 8

Conclusion

8.1 Summary of results

The aim of this thesis was to investigate the potential of eye tracking technology, to help recognizing the intent of humans when working with a machine under shared control. Eye gaze has been proven to be a rich source of information when a human is planning a task. Especially when performing a physical task, the human gaze tends to fixate on key objects, before they take physical action. The chosen shared-control setting was the co-manipulation of a two degrees-of-freedom (DOF) SCARA robot. Test subjects were presented with a path planning task, where they were requested to maneuver the robot's end-effector through a sequence of LEDs on the robot action plane. The LED sequence of each run had the same start- and end-positions, while the remaining 5 were chosen randomly out of 10 in total. In a first stage, the robot arm was unactuated and the focus was to collect eye gaze data from the subjects while they were guiding the passive robot through the sequence. It was found that most gaze fixations of the subjects tend to fall onto the LEDs they had intended to connect. In fact, around 4.5 as much as the fixations not in proximity of these goal LEDs. In addition, when the subjects planned to move from one LED to the next, the subject's gaze tended to fixate on the next LED between one and two seconds before reaching the position with the robot end-effector, depending on how much distance the subject had to cover when moving from the current LED to the next. After reaching the fixated position, the gaze is shifted almost immediately (with 0.1-0.2s delay) onto the next LED, while movement onset is delayed about 0.5 seconds. This information was then used to develop an algorithm to predict which LED a subject is intending to reach. In a second data collection, where the algorithm was tested, it was shown that the algorithm was successfully recognizing the LEDs the subjects are trying to navigate the robot to and could support them in the movement. How far ahead of time the goals were recognized was dependent on how soon the

subjects gaze shifted from a reached LED to their next planned goal LED. If the subject fixates on a goal LED 0.3s before initiating the movement towards it, the robot was able to perform the whole movement between the LEDs. In most cases the algorithm initiated the support half-way through the planned motion of the subjects.

8.2 Further work

- By encouraging the subjects to perform the experiment faster, or by defining a time constraint to the experiment completion time the goal queue implemented in the algorithm could achieve increased execution speed.
- Since the algorithm could only be tested with one subject, due to recommended self isolation in connection with the COVID-19 virus, the collected data was limited. If more subjects could be recorded, better understanding between the differences of the passive and active robot experiment could be obtained.
- If the servo motors of the robotic arm could be replaced by ones that have more control possibilities, a more subtle switch between active and passive robot could be achieved. This would result in more comfort for the user, which is less irritating.
- To receive feedback from the user during manipulation of the end-effector it could prove valuable to monitor the user's reaction forces on the robotic arm. This can be achieved by installing pressure pads, piezo-sensors or torque sensors on the end-effector.

Bibliography

[1] *AprilTag markers*. https://april.eecs.umich.edu/software/apriltag/. Accessed: 2020-02-21.

[2] Aronson, Reuben M, Santini, Thiago, Kübler, Thomas C, Kasneci, Enkelejda, Srinivasa, Siddhartha, and Admoni, Henny. "Eye-hand behavior in human-robot shared manipulation". In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 2018, pp. 4–13.

[3] Dragan, Anca D and Srinivasa, Siddhartha S. "A policy-blending formalism for shared control". In: *The International Journal of Robotics Research* 32.7 (2013), pp. 790–805.

[4] Huang, De-An, Farahmand, Amir-massoud, Kitani, Kris M, and Bagnell, James Andrew. "Approximate maxent inverse optimal control and its application for mental simulation of human interactions". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[5] Huang, Rulin, Liang, Huawei, Zhao, Pan, Yu, Biao, and Geng, Xinli. "Intent-estimation-and motion-model-based collision avoidance method for autonomous vehicles in urban environments". In: *Applied Sciences* 7.5 (2017), p. 457.

[6] Kahneman, Daniel and Beatty, Jackson. "Pupil diameter and load on memory". In: *Science* 154.3756 (1966), pp. 1583–1585.

[7] Kassner, Moritz, Patera, William, and Bulling, Andreas. "Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction". In: *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*. 2014, pp. 1151–1160.

[8] Kwon, Minae, Biyik, Erdem, Talati, Aditi, Bhasin, Karan, Losey, Dylan P, and Sadigh, Dorsa. "When Humans Aren't Optimal: Robots that Collaborate with Risk-Aware Humans". In: *arXiv preprint arXiv:2001.04377* (2020).

[9] Nehaniv, Chrystopher L, Dautenhahn, Kerstin, Kubacki, Jens, Haegele, Martin, Parlitz, Christopher, and Alami, Rachid. "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction". In: *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* IEEE. 2005, pp. 371–377.

[10] Ohn-Bar, Eshed and Trivedi, Mohan Manubhai. "Looking at humans in the age of self-driving and highly automated vehicles". In: *IEEE Transactions on Intelligent Vehicles* 1.1 (2016), pp. 90–104.

[11] Pelz, Jeff, Hayhoe, Mary, and Loeber, Russ. "The coordination of eye, head, and hand movements in a natural task". In: *Experimental brain research* 139.3 (2001), pp. 266–277.

[12] *Pupil labs eye tracking device*. `https://pupil-labs.com/products/core/`. Accessed: 2020-02-02.

[13] Salvucci, Dario D and Goldberg, Joseph H. "Identifying fixations and saccades in eye-tracking protocols". In: *Proceedings of the 2000 symposium on Eye tracking research & applications*. 2000, pp. 71–78.

[14] Ziebart, Brian D, Bagnell, J Andrew, and Dey, Anind K. "The principle of maximum causal entropy for estimating interacting processes". In: *IEEE Transactions on Information Theory* 59.4 (2013), pp. 1966–1980.

List of Figures

Appendix

A.1 Pseudo code of algorithm

Algorithm 1: Recive gaze data and detect subject's goal

```
while Program running do
    if robot is calibrated then
        receive fixations;
        if fixation is on surface then
            if confidence of gaze data > confidence threshold then
                if fixation's dispersion-duration-ratio < threshold then
                    RW position of fixation = map normalized position to RW position on surface;
                    for all LED positions on surface do
                        if RW position of fixation in proximity of LED then
                            potential goal = LED position on surface;
                        end
                    end
                end
                if potential goal not current goal and not in goal queue and not in reached goals
                 then
                    add potential goal to goal queue;
                end
            end
        end
    end
end
```

Algorithm 2: Detect and calculate current RW position of end-effector

```
calculate affine transformation matrix with known RW positions of AprilTag markers;
while Program running do
    grab current camera frame;
    pixel coordinates of end-effector = detect marker of end-effector in current frame;
    if no marker detected then
        set pixel coordinates of end-effector to previous one;
    end
    RW position of end-effector = affine transform pixel coordinates of end-effector;
    if robot is calibrated then
        RW position of end-effector = parallax error correction of RW position of end-effector;
    end
end
```

Algorithm 3: Move robot to current goal position

```
while Program running do
    if robot is calibrated then
        if no current goal then
            if goal queue is not empty then
                current goal = get first goal in goal queue;
                current end-effector position = get current end-effector position;
                current joint motor angles = calculate inverse kinematics of current end-effector
                  position;
                attache servo motors at current joint motor angles;
            end
        else
            while current joint motor angles not goal joint angles do
                move motors towards goal joint angles;
            end
            add current goal to reached goals;
            if goal queue is not empty then
                current goal = get first goal in goal queue;
            else
                current goal = no current goal;
                detach servo motors;
            end
        end
    end
end
```