MSc Thesis.

07.09.2022

Gaze-contingent interfaces:

The effect on noticing times of time critical warning messages under Level 2 driving conditions while interacting with in-vehicle technology.

Rhita Addi
Delft University of Technology
Department of Cognitive Robotics
MSc Mechanical Engineering
Vehicle Engineering track
Human Factors specialization

Supervisors:
Yke Bauke Eisma
Harmen Cnossen
Jonas Glimmann
Felix Schüssel

Mercedes-Benz TUDelft

Gaze-contingent interfaces: The effect on noticing times of time critical warning messages under Level 2 driving conditions while interacting with in-vehicle technology.

Rhita Addi
Cognitive Robotics
Delft University of Technology
Delft, Netherlands
r.addi@student.tudelft.nl

***Abstract*—With the introduction of automated driving systems come benefits such as the improvement of traffic safety. However, with an increasing level of automation in vehicles also comes an increase in interaction with in-vehicle technology by drivers while they are meant to supervise the automated driving systems. Due to more interaction with in-vehicle technology and a vigilance decrement of the driver in Level 2 driving, an increase in reaction time of the driver is seen when intervention is needed by the means of a take over request. This delay in reaction by the driver opposes the benefit of the introduction of automated driving systems and causes hazardous situations. To try and circumvent the effect of vigilance decrement, this paper attempts to demonstrate the reduction of noticing time of the Hands-On-Wheel warning message for drivers of Level 2 vehicles while interacting with in-vehicle technology through the implementation of a gaze-contingent interface. The results of this experiment indicate a 79.3% lower noticing time of the Hands-On-Wheel warning message when the stimulus is placed in a gaze-contingent manner, while the participants engage in secondary tasks on the in-vehicle technology. The placement of the stimulus on the head unit when the participant is already looking at it reduces the primary task load of touching the steering wheel and causes for the stimulus to be seen quicker as compared to a static interface. However, the performance of the secondary task seems to decrease when using a gaze-contingent interface. This is due to the intrusive nature of the placement of the stimulus, which demands the driver to store information regarding the secondary task in their working memory while they attend to the primary task. Despite the decline in secondary task performance, the reduction of noticing times of time critical messages when placed in a gaze-contingent manner could be beneficial to the safety of autonomous driving functions where the driver has a vigilance task and is engaging in secondary tasks.**

***Index Terms*—Gaze-contingent, Eye tracking, Interface, Autonomous driving**

I. Introduction

For Level 2 partial driving automation, vehicles have automated lateral and longitudinal vehicle motion control as defined by the Society of Automotive Engineers. In practice, this could translate into functionalities such as adaptive cruise control (ACC) and automated lane keeping (ALK). Such systems are considered to be a driver support system where the driver is still responsible for (some) of the dynamic driving task. With active ACC and ALK, the driver does not have to give any speed or steering control input as that is automated. However, it is expected of the driver to supervise the partial driving automation system and execute general object and event detection in case the support system is unable to function. In such a case, the driver needs to be able to respond appropriately [1].

The implementation of automated driving systems such as the ACC and ALK functionalities in Level 2 driving brings along a multi-faceted set of benefits. The main benefit being the improvement of traffic safety by reducing human induced traffic incidents through the use of systems such as ACC and ALK. Other benefits consist of traffic congestion reduction, improvement in fuel economy and travel time reduction [2].

However, with an increasing level of driver support, drivers tend to shift their attention from the road to execute other non-driving related tasks (NDRT). Especially interaction with in-vehicle technology (IVT) such as radio and infotainment system interaction seem to increase in the presence of Level 2 automated driving systems (ADS) [3].

In an on-road study with activated Level 2 functionalities, video data shows that drivers are complacent and over-trust the ADS by not adhering to their responsibilities of monitoring the support system. This increased interaction with IVT is clear evidence for the inefficiency of humans as a monitory component in a dynamic driving task [4]. Especially in the case of Level 2 driving, where transitions from automated driving to manual driving occur regularly, the engagement in non-driving related tasks can be problematic.

These transitions from automated driving to manual driving are facilitated by a take over request (TOR). This process can be decomposed into three different stages [5]:

1) Noticing time: defined as the moment in time when the driver's gaze moves away from the secondary task until it fixates on the time critical warning message on the display/interface.

2) Hands-On-Wheel time: defined as the time from the fixation on the stimulus until the driver touches the steering control.
3) Intervention time: defined as the time from the first steering control touch to the first steering control input.

Once exposed to a take over request, the driver has to go through the above mentioned three stages. However, when engaging in secondary tasks under Level 2 driving, it takes significantly longer (± 1.5 sec) [6] for drivers to go through all three steps. This can result into dangerous situations with a proven higher amount of collisions in high density traffic [7].

When considering the aim of automated driving systems, it is to try and provide safety benefits. However, the use of such automated systems also force the driver into a supervisory role. When a driver is meant to supervise said automation, they are required to keep their visual attention in a situation in which not a lot changes or happens. From the vigilance studies done by Mackworth. [8], it was concluded that human operators do not perform well in such situations. The likelihood of the detection of abnormalities by human operators actually deteriorates as a function of time. As one can imagine and as has been proven by Mackworth [8], the longer an operator has to supervise a situation where in nothing noteworthy changes, the less probable it is that the operator will detect a signal when needed [8].

When translating this phenomenon onto the take over process, this could impose new safety hazards. The inattention of the operator during their vigilance task contradicts with the added benefits of using automated driving systems. This is paradox is described as the "ironies of automation" [9], which describes that a human operator with monitoring tasks over an automated system brings additional complexities with it together with the planned benefits of automation.

Considering this potential safety hazard due to inattention, it is worth wile to attempt to mitigate the ironies of automation by trying to minimize the effect of engaging in non-driving related tasks on transitions as mentioned in Eriksson and Stanton [6]. One way of alleviating the shortcomings of vigilance by drivers of vehicles equipped with ADS, is to keep the signal rate with which the driver has to pay attention artificially high, as is recommended by Bainbridge [9]. Current vehicles equipped with Level 2 automated driving systems attempt to keep the drivers' attention as a supervisor to the system through the display of an optical warning signal. This warning signal demands the driver to touch the steering wheel to ensure the driver is still aware of its Level 2 supervisory tasks according to the timeline as depicted in fig. 1.

The optical warning indicates hands and the steering control with an (optional) additional explanatory text that tells the driver to place their hands on the steering control. This is standardized according to Regulation No 79 of the Economic Commission for Europe of the United Nations [10], and is a mandatory feature for any steering control assist systems.

The optical warning, specified as the Hands-On-Wheel (HOW) message, needs to be displayed at $t_{HOW,s}$ after the driver has not been holding the steering control for up to 15 seconds with activated Level 2 ADS functionalities (S_{Auto}). If the driver has seen the HOW message, they need to react and touch the steering control to eliminate the warning message and continue in partial autonomous driving S_{Auto}. The driver has 30 seconds from $t_{HOW,s}$ onward to react, after which the HOW message will turn red and an additional acoustic warning message will be added. This instance is depicted as time instance $t_{HOW,critical}$ in scenario B for when the warning remains unnoticed (see fig. 1. The system will maintain this level of warning message for a duration of 30 seconds, represented by $S_{Critical}$. After the 30 seconds in $S_{Critical}$ without a driver response, the ADS will be deactivated in a safe manner and the vehicle will come to a halt in a safe place. This is indicated by S_{Safe} in fig. 1.

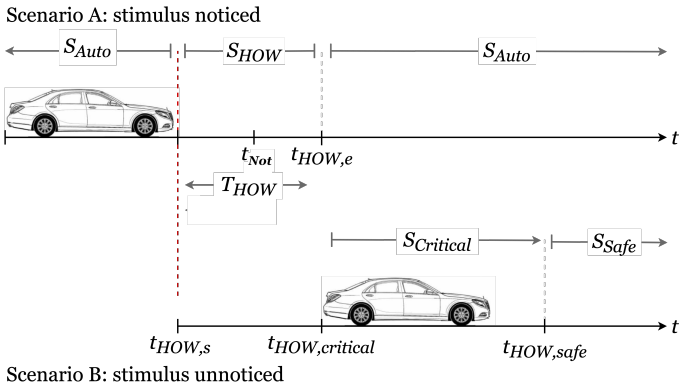


Fig. 1: Timeline of the driver optical warning signal through the various states of the automated driving system. Scenario A describes the progression of events when the stimulus is noticed by the driver. Scenario B the events after an unnoticed stimulus.

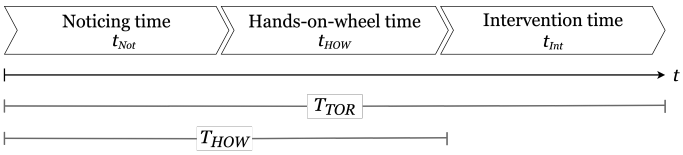


Fig. 2: Schematic display of different stages of a take over request and Hands-On-Wheel request.

As mentioned, the use of such an optical warning system like the HOW message is used to help keep the attention of drivers of Level 2 vehicles. However, it can be expected that due to the driver's distraction by the IVT in Level 2 driving, a prolonged Hands-On-Wheel time due to engagement in secondary tasks occurs. This is assumed because of the similarity between the TOR and HOW stages the driver has to undertake, as can be seen in fig. 2.

This prolonged T_{HOW} can be problematic and hazardous, as mentioned by Radlmayr et al. [7]. Therefore, it is key

to try and minimize this side effect of the implementation of Level 2 ADS functionalities. As can be seen in fig. 2, T_{HOW} comprises of two temporal components; the noticing time t_{Not} and Hands-On-Wheel time t_{HOW}. Attempting to lower the T_{HOW} of a driver in a Level 2 vehicle while interacting in-vehicle technology can be therefore be achieved by lowering either of the mentioned temporal variables.

The aim and scope of this paper lies with attempting to reduce the noticing time, t_{Not}, of the Hands-On-Wheel warning message for drivers of Level 2 vehicles while interacting with in-vehicle technology through the implementation of a gaze-contingent interface.

If the Hands-On-Wheel warning message is displayed in a gaze-contingent manner, it is hypothesized that the noticing time of the warning message will be lower in Level 2 driving while interacting with in-vehicle technology as compared to noticing times of the same warning message in a static interface under the same driving conditions.

This hypothesis is based on the experimental results achieved by Pomarjanschi et al. [11] whom explored the possible benefits of using gaze guidance in (simulated) driving in safety-critical situations. In a simulated distracted driving scenario, their participants were exposed to potentially dangerous events. These events were highlighted with (temporally transient) gaze-contingent cues when the subjects were not looking at the dangerous event to direct the driver's attention to it. This resulted in shorter reaction times and noticing times of the events when making use of the gaze-contingent cues [11].

Using the same principle of gaze-contingent placement of stimuli, this research hypothesizes to achieve the same result of shorter noticing times of time-critical events, such as the Hands-On-Wheel warning message. Secondarily, when introducing a gaze-contingent interface while interacting with in-vehicle technology, it is hypothesized that the intrusive nature of the gaze-contingent cue might also affect the participants' ability to utilize the IVT as wanted. Therefore, it is also key to monitor the performance of the participants' non-driving related task and assess whether the gaze-contingent interface has an effect on this.

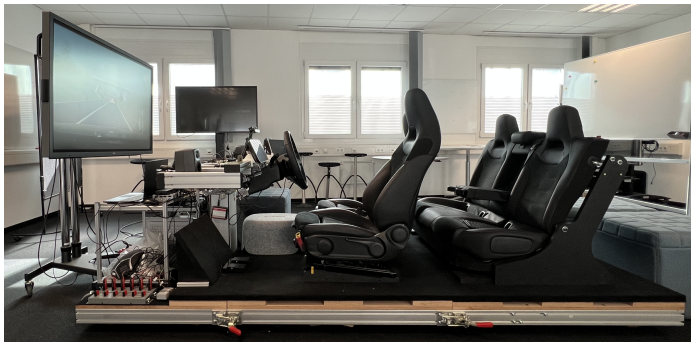
II. Method

A. Participants

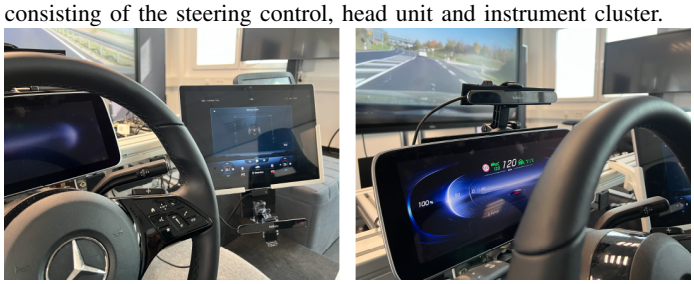
Participants of this experiment consisted of 16 Mercedes-Benz employees (8 female, 8 male) with a mean age of 28.63 years (SD = 7.43). All participants have a valid driver's license and have given informed consent to participate in this experiment.

B. Apparatus

This experiment will simulate two scenarios in which the participant is driving on a highway with activated Level 2 functions in a Mercedes-Benz vehicle. This means that the participant does not need to provide longitudinal or lateral control during the experiment, as it is simulated that Active Steering Assist (ALK) and Distronic+ (ACC) is in use in the Mercedes-Benz simulated vehicle. Both scenarios will take place in the same experimental setup that consists of various components; the seating buck, displays, eye trackers, interface, stimulus and developmental software.



(a) Seating buck with external display placed in front of the dashboard consisting of the steering control, head unit and instrument cluster.



(b) Placement of the Tobii Pro Nano eye tracker below the head unit *(left)* and above the instrument cluster *(right)*.

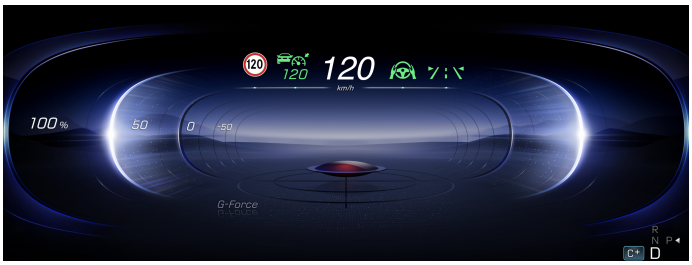
Fig. 3: Complete experimental setup.

1) The seating buck: The seating buck consists of a physical cabin model of a Mercedes-Benz vehicle. This structure is fully equipped with a driver, passenger and back seat. The front row is also equipped with a dashboard-like structure containing the displays, with on the the driver side pedals for longitudinal control and a steering-wheel in place for lateral control. The configuration can be seen in fig. 3a.

2) The displays: The displays are embedded into the seating buck to simulate the instrument cluster and the head unit. Together, the two displays make up the interface. Display specifications and placement details are disclosed in Appendix V-C. For this experiment, the passenger display has been kept out of the scope. In addition to the seating buck displays, an additional large external display has been positioned in front of the seating buck to provide a visual stimulus for simulated driving on highway roads. This display does not play a role in either the primary or secondary task of this experiment. However, it does add to the perceptual fidelity of the experiment by bringing the perception of the participant in the simulator as close as possible (within means available) to the perceivable reality [12] of a driver of a Level

2 vehicle.

3) The eye trackers: The eye trackers that are used in this experiment are two Tobii Pro Nano screen-based trackers that measure the x and y position of a person's gaze at a sample frequency of 60 Hz. One eye tracker is placed above the instrument cluster and the other below the head unit. The offset of both eye-trackers are recorded in the Tobii Pro Eye Tracker Manager for calibration purposes. The positioning of the eye trackers can be seen in fig. 3b.



(a) Mercedes-Benz NTG7 interface on the instrument cluster.

(b) Mercedes-Benz NTG7 interface on the head unit

Fig. 4: The base interface used in the experiment which facilitates the secondary tasks and displays the stimulus.

4) The interface: The interface used in this experiment uses components of the NTG7 Mercedes-Benz interface and comprise of the instrument cluster (IC) and the head unit (HU). The instrument cluster, as can be seen in fig. 4a, has the G-force disc in the centre of the interface. At the top of the interface, the speed is indicated as well as the activation of the ADS, which are the active steering assist and Distronic+ in this case. The head unit, as can be seen in fig. 4b, consists of the basic NTG7 home screen with a menu bar at the bottom. The menu bar comprises of some basic climate control functionalities and options to go into the app menu. This experiment only makes use of the app menu, where the participants can find the non-driving related tasks to be executed during the experiment as a secondary task.

These interface features make up the base of the two different interfaces that will be tested against each other in this experiment, the static NTG7 interface and the gaze-contingent NTG7 interface. Both interfaces will display the stimulus of the Hands-On-Wheel warning message according to the timeline in fig. 6b. The difference between the static and gaze-contingent interface is the placement of the stimulus.

The static interface will always display the stimulus, at $t_{HOW,s}$, in the centre of the instrument cluster. The gaze-contingent interface will always display the stimulus, at $t_{HOW,s}$ as well. However, in contrary to the static interface, the stimulus will be placed in the centre of the display the participant is looking at $t_{HOW,s}$, limited to the instrument cluster and head unit. This is done by checking the change in x and y position at timestamp $t_{HOW,s}$. If the participant is not looking at either screen, meaning that there is no change in x and y coordinates, the stimulus will be shown on the display with the latest gaze activity prior to $t_{HOW,s}$. The schematic representation of the development of the Hands-On-Wheel warning message is depicted in fig. 5.

Both interfaces have been prototyped using ProtoPie and ProtoPie Connect. The details of this prototyping process can be found in Appendix V-A.

5) The stimulus: The stimulus, known as the Hands-On-Wheel warning message, itself comprises of a pictorial message of the steering control with two hands gripping the wheel. The hands start out to be blue at $t_{HOW,s}$ and turn red 30 seconds after $t_{HOW,s}$. The hands on the steering control in the stimulus start flashing red 30 seconds after $t_{HOW,Critical}$. The auditory messages are left out of the scope for this experiment. However, it has been included in the experimental setup to enhance the fidelity and adhere to the regulations imposed by the Economic Commission for Europe of the United Nations. The auditory message is only introduced at $t_{HOW,critical}$, 30 sec after the introduction of the visual stimulus. Therefore, it is assumed that the impact of the gaze-contingent placement of the visual stimulus is not effected by the auditory message.

Furthermore, as the stimulus is depicted at $t_{HOW,s}$, a blur is eased in to mask the interface background of the stimulus. The timeline of the stimulus is summarized in fig. 6.

The stimulus is removed from either the instrument cluster or the head unit once the steering wheel is touched by the participant. This will be removed manually by the experimenter through the Wizard of Oz approach, where the experimenter fills in for a piece of (missing) technology to create the same experience for the participant of the experiment [13]. The stimulus is removed through a touch of a button on a keyboard out of sight from the participant. This is due to the lack of sensors in the seating buck that can detect touch in the steering wheel.

C. Procedure

All participants undergo the experiment twice, under two different conditions: Condition A is the control condition which tests the static NTG7 interface. Condition B tests the gaze-contingent NTG7 interface, making this a within-subject experiment. The participants take place in the seating buck in the driver's seat, where after both eye trackers are calibrated to their respective screen using the Tobii Pro Eye Tracker Manager.

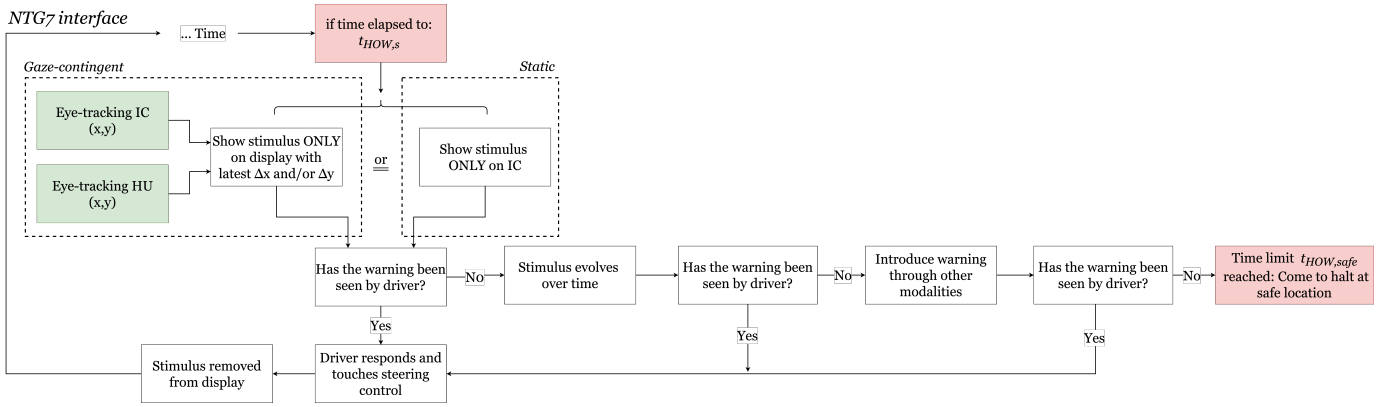
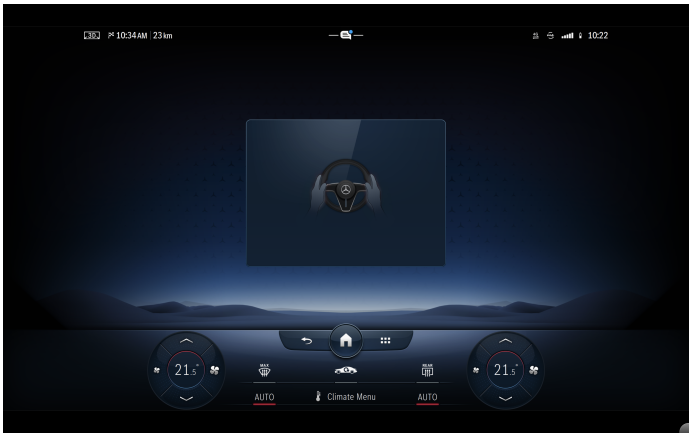
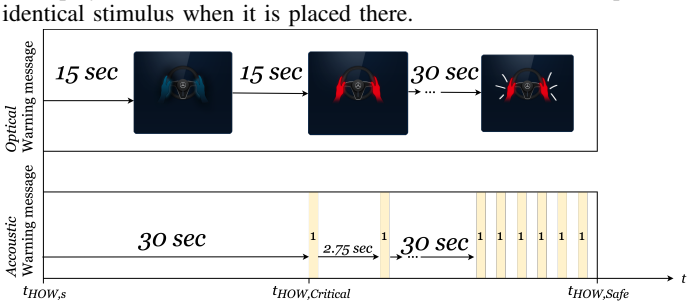


Fig. 5: Schematic representation of the development of the Hands-On-Wheel warning message, referred to as stimulus, over time. Both the gaze-contingent and the static path of the logic are displayed in the same figure to show the similarity between the two interfaces.



(a) Optical representation of the Hands-On-Wheel warning message as displayed on the head unit. The instrument cluster will depict the identical stimulus when it is placed there.



(b) Timeline of the optical and acoustic development of the Hands-On-Wheel warning message. This includes the same temporal marks as in fig. 1 on the x-axis.

Fig. 6: Overview of the stimulus presented at $t_{HOW,s}$.

The participant takes position in the driver seat 71.5 cm behind the instrument cluster. The IC has a resolution of 900 x 2400 pixels with a height of 11 cm. The eye tracker is placed 5 cm above the instrument cluster. The head unit is placed 78.6 cm in front of the participant, 29.3 to the right of the instrument cluster. This screen has a resolution of 1728 x 3088 pixels with a height of 23.2 cm. The eye tracker is placed 5 cm below the head unit. When the stimulus is triggered, it will be displayed in the centre of either display.

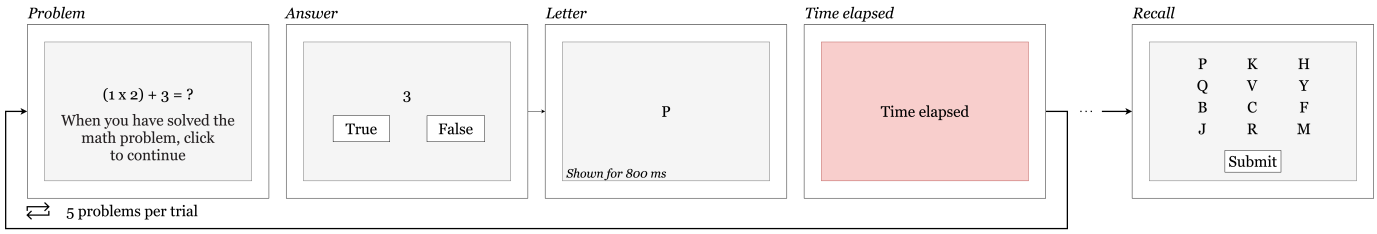
The experiment consists of a primary and a secondary task the participants have to execute. Completing all tasks in both the static and gaze-contingent interface takes 30 minutes.

The primary task of the participant is to touch the steering wheel as soon as they are exposed to the Hands-On-Wheel warning message. Once they have done this, the warning message will be removed from the interface and the participant can continue with their secondary task. The warning message repeats itself during the execution of the secondary task as elaborated upon in 6b.

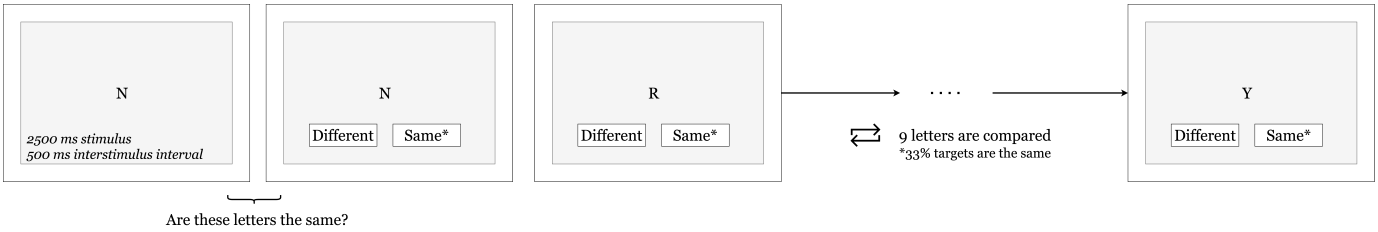
The experiment simulates Level 2 driving which means besides the primary task, the participant does not need to interact with the steering control nor the gas pedals.

The secondary tasks of the participant consists of the non-driving related tasks to simulate in-vehicle technology interaction and the mental workload that comes with attaining situation awareness when driving using partial automation [16]. According to Chen et al., using (partial) automation in driving comes with a certain mental workload. The secondary tasks have been selected accordingly to simulate said mental workload and attempt to increase behavioral fidelity of the participants. These tasks consist of the AOSPAN and 1-back task and are explained to the participant through an example trial run prior to the data collection. Next, participants execute three trials of the AOSPAN task and three trials of the 1-back task on the head unit while the external display shows a video of simulated high way driving. While performing the tasks, the stimulus is displayed according to fig. 6b after which the primary task is ought to be executed.

After the in total six consecutive trials of both secondary tasks are completed in the static NTG7 interface, the same tasks are then repeated in the gaze-contingent interface once again. The order in which conditions A and B are tested alternates between participants.



(a) Schematic representation of the mathematical questions posed to the participant with corresponding letter depiction that needs to be recalled. This is done for five consecutive mathematical problems and letters after which the recall screen is displayed [14]. This process is considered to be one trial.



(b) Schematic representation of the letters that need to be recalled by the participants. This is done for nine letters of which 33% of the letters are identical when comparing to the previously displayed letters [15]. This process is considered to be one trial.

Fig. 7: Schematic representation of the secondary tasks of which the participants will first do a practice trial of each task before the recording of the performance starts. Each participant will thereafter do three trials of each task.

1) The AOSPAN task: The Automated Operation Span Task exposes the participants to various simple mathematical problems which have to be answered. The participants are then shown a letter which they have to remember. This is repeated five times, after which the participants are required to recall all five letters that have been presented after the math problems [14]. To clarify, the schematic representation of the AOSPAN task can be seen in fig. 7a. This task will be displayed on the head unit during the experiment.

2) The 1-back task: In the 1-back task, the participant is exposed to a letter that is displayed on the head unit. The participant has to determine whether the letter that is currently being shown is the same as the letter that has been shown one letters back [15]. To clarify, the schematic representation of the 1-back task can be seen in fig. 7b. This task will be displayed on the head unit during the experiment.

D. Independent variable

The independent variable is the adaptability of the NTG7 interface that is used in the experiment. Condition A uses the static NTG7 interface which always places the stimulus on the instrument cluster. Condition B uses the gaze-contingent NTG7 interface which places the stimulus on the display where the participant is looking. The independent variables are also annotated by the dashed boxes in fig. 5.

E. Dependent variables

Mean noticing times of stimulus, $\bar{t}_{Not}$ [sec]: The main dependent variable that will measure the performance of the newly implemented gaze-contingent NTG7 interface is the noticing times of the stimulus. The noticing time is defined through means of the determined fixations on the stimuli. These fixations are obtained from the measured x and y coordinates. First, the missing coordinates due to blinking or other instances with no eye tracking measurements are restored through linearly interpolating adjacent data points. A median filter with a 100 ms interval is used on the x and y coordinates to remove noise [17]. From this, the time derivative is calculated for each participant to determine the gaze velocities $\dot{x}$ and $\dot{y}$. The gaze velocity is determined as the euclidean vector of the individual velocity components of the x and y coordinates. Using equation 1, the angular velocity is determined as the Euclidean distance of the $\dot{x}$ and $\dot{y}$ components. Where, f_s is the sampling frequency of the eye tracker and ϕ denotes the factor converting pixels to visual degrees (see eq. 2). In equation 2, h_{cm} represents the height of the display in centimeters, h_{pix} the vertical resolution of the display in pixels and D the distance to said display.

$$\dot{\theta}_i = f_s \phi \sqrt{\dot{x}_i^2 + \dot{y}_i^2} \tag{1}$$

$$\phi = \frac{4 \arctan(h_{cm}/2D)}{h_{pix}} \tag{2}$$

The gaze velocity data is then filtered to discriminate between saccades and fixations. First, any gaze velocity data above 1000 deg/sec is eliminated for being physiologically impossible for the human eye to reach [18]. Second, a saccade velocity threshold of 2000 pix/sec is assumed [17]. Any gaze velocity above this threshold is classified as a saccade. Additionally, any gaze velocity below the saccade velocity threshold is assumed to be a fixation if it adheres to a minimum fixation duration threshold of 50 ms [19]. For each of the established fixations, a median x and y position is determined

to conclude a fixation location. If the fixation is within bounds of the displayed stimulus, then the start time of the fixation can be used to assume when the stimulus has been noticed. For this specific experiment that means that the median x and y location has to lie within the bounds of $750 < x < 1650$, $118.725 < y < 781.275$ for the instrument cluster and $1094 < x < 1994$, $553.725 < y < 1216.225$ for the head unit. The noticing times of the stimulus is defined in three different manners:

1) The noticing time can be defined as the time difference between the stimulus start time, $t_{HOW,s}$, and the closest fixation start time after the stimulus, $Fix_{closest}$. This describes the moment a participant moves their gaze from elsewhere to the stimulus.
2) The noticing time can be defined as zero when the participant is already fixated at the right location when the stimulus is presented. This is when the fixation occurs before the stimulus is presented and the fixation duration exceeds the difference between the fixation start time and the stimulus start time.
3) The noticing time can be defined as *NAN* when the participant's reaction to the stimulus is recorded before a fixation takes place. It is then assumed that the participant has seen the stimulus in their peripheral vision.

One of the three above mentioned definitions establish a noticing time for each instance a stimulus is represented for each participant. The pseudo code of the above defined noticing times can be seen in fig. 8. The resulting noticing times are averaged for each participant for each condition to reach the desired metric of mean noticing time of the stimulus, $\bar{t}_{Not}$.

Noticing times, t_{Not}

```
# Find closest fixation to HOW_s
Fix_closest= find_nearest(Fix_start,HOW_s)

# Determine preliminary noticing times, t_Not
t_Not = Fix_closest- HOW_s

# Determine if participant was already fixated on stimulus
for i in range(0,len(t_Not))
   # If noticing times is negative and the duration
     of the fixation exceeds the start time of the stimulus
   if (t_Not(i) < 0 AND abs(t_Not(i))) < Fix_duration(i)
      # Noticing time assumed to be zero
      t_Not(i) == 0

# Determine if participant has noticed stimulus through
  peripheral vision
for i in range(0,len(t_Not))
   # If the end of the stimulus is earlier in time than
     the closest fixation to the screen
   if (HOW_E(i)  < Fix_closest(i)
     # Noticing time for that stimulus cannot be
       determined
      t_Not(i) == NAN
```

Fig. 8: Pseudo code: a display of how the three defined noticing times will be distinguished in the code.

Performance of AOSPAN task, P_{AOSPAN} [%]: The performance of the AOSPAN task is expressed in a percentage of correctly answered questions. This includes both the correctly answered mathematical problems and recollected letters. The order with which the letters are remembered is insignificant and does not influence the score. The score of all mathematical problems and letters of all three trials are summarized into a percentage of correct answers out of the total asked questions per participant for each condition.

Performance of 1-back task, P_{1-Back} [%]: The performance of the 1-back task is expressed in a percentage of correctly remembered letters. All letters of all three trials are summarized into a percentage of correct answers out of the total asked questions per participant for each condition..

Percentage of stimulus seen through peripheral vision, PV [%]: As mentioned in the third definition of the noticing time of the stimulus, it can be assumed that occasionally participants see the stimulus through peripheral vision. The amount of times this occurs per participant is taken as a percentage of the total amount of presented stimuli per participant for each condition.

F. Analyses

Due to a small sample size of participants, it is key to chose an appropriate statistical test that can help determine if the found differences are statistically significant. The first step in this, is to determine the nature of the data and whether it can be assumed to be normally distributed. This is done by executing the Shapiro-Wilk test with an assumed p-value of 0.05. The null-hypothesis is that the data used for the dependent variables is normally distributed.

- If $p \leq 0.05$: then the null hypothesis can be rejected and the data can be assumed to not be normally distributed.
- If $p > 0.05$: then the null hypothesis cannot be rejected and the data could be normally distributed.

TABLE I: Shapiro-Wilk test for normality of all dependent variables in each condition

Dependent variable	W	P
$\bar{t}_{Not,s}$	0.917	0.151
$\bar{t}_{Not,g}$	0.696	0.000153
$P_{AOSPAN,s}$	0.886	0.0473
$P_{AOSPAN,g}$	0.957	0.610
$P_{1-Back,s}$	0.861	0.0197
$P_{1-Back,g}$	0.696	0.000153
PV_s	0.867	0.0246
PV_g	0.849	0.0246

Four dependent variables, with each two sub data-sets of the static and gaze-contingent data, have been exposed to the Shapiro-Wilk test. For all the four main variables, at least one of both subsets can reject the null hypothesis. Meaning that it can be assumed that the data is not normally

distributed. Therefore, the Wilcoxon Signed Rank Test with a significance level of 0.05 is used to determine statistical difference between conditions, since this test is suited for non-normally distributed data sets [20].

III. Results

A. Statistics

Statistical analysis was performed using a Wilcoxon Signed Rank test with a significance level of 0.05. This test is chosen because of the non-normality of the data. After the performed statistical analysis, the null hypothesis' for the mean noticing time $\bar{t}_{Not}$ (eq. 3) and the performance of the 1-Back task P_{1-Back} (eq. 6) can be rejected. This means that these two dependent variables can be considered significantly different. The null-hypothesis' for the other dependent variables (eq. 4 and eq. 5) cannot be rejected and are therefore considered not significantly different.

Null Hypothesis H_0, t_{Not}:

$$t_{Not,S} = t_{Not,GC} \tag{3}$$

Null Hypothesis H_0, PV:

$$PV_S = PV_{GC} \tag{4}$$

Null Hypothesis H_0, P_{AOSPAN}:

$$P_{AOSPAN,S} = P_{AOSPAN,GC} \tag{5}$$

Null Hypothesis H_0, P_{1-Back}:

$$P_{1-Back,S} = P_{1-Back,GC} \tag{6}$$

TABLE II: Wilcoxon Signed Rank Test with a significance level of 0.05 is used to determine statistical difference between conditions

Dependent variable	W	P
t_{Not}	0	3.05e-05
P_{AOSPAN}	40.5	0.725
P_{1-Back}	3.5	2.13e-04
PV	37.0	0.117

B. Aggregate gaze results

Fig. 9 displays the aggregated distribution of all fixation coordinates of all participants on both displays for each condition. To achieve this, all x and y coordinates of the fixations made on the head unit and instrument cluster are concatenated together for all participants. These coordinates are plotted in a 2D histogram to create a heat map, with a specified bin resolution that corresponds to the $0.3°$ accuracy of the used Tobii Pro Nano eye trackers.

The heat map shows where on the displays the most fixations take place. Fig. 9a shows the aggregate fixation locations of all the participants completing the primary and secondary tasks using the static NTG7 interface. This clearly shows the area in yellow on the head unit which the participants fixate on the most. This area also requires the most attention from the participants during the secondary tasks as it displays the questions for the AOSPAN and 1-Back task.

The instrument cluster shows a similar result for the static NTG7 interface as seen in fig. 9a, where the fixation location is concentrated around the centre of the display. This location corresponds to the placement of the Hands-On-Wheel warning message that is always displayed on the instrument cluster in condition A.

Fig. 9b shows the aggregate fixation locations of all participants in condition B. At first glance, the heat maps of both conditions look very similar. Especially the head unit shows the same concentration of fixations around the same location where the stimuli and secondary tasks are positioned.

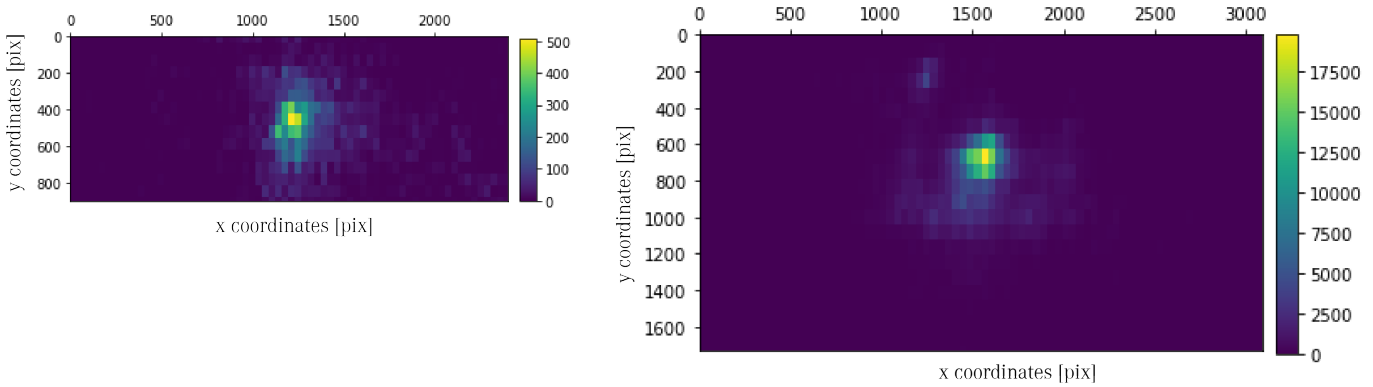
One evident difference between the aggregated fixation locations of all participants between both conditions, is the gaze data for the instrument cluster in condition B. It is apparent from fig. 9b, that the fixation locations are more spread out over all the participants on the instrument cluster. The area on which the participants fixate is more diffuse compared to the more concise area in the static NTG7 interface. This might indicate that in condition A, the instrument cluster draws the attention through the Hands-On-Wheel message which is always placed on the same location. However, in the gaze-contingent NTG7 interface, the Hands-On-Wheel warning message is placed on the display which the participants are looking at at $t_{HOW,s}$. Due to non-driving related tasks on the head unit, the stimulus is always shown on the head unit in the gaze-contingent NTG7 interface. This means that any fixations towards the instrument cluster are voluntary and do not necessarily serve a purpose towards the secondary task. Therefore the location of said fixations are more of exploratory nature and are not as limited as in condition A, where the fixations seem to concentrate around where the stimuli are placed and thus serving a purpose towards the primary task.

C. Mean noticing times of stimulus, $\bar{t}_{Not}$ [sec]

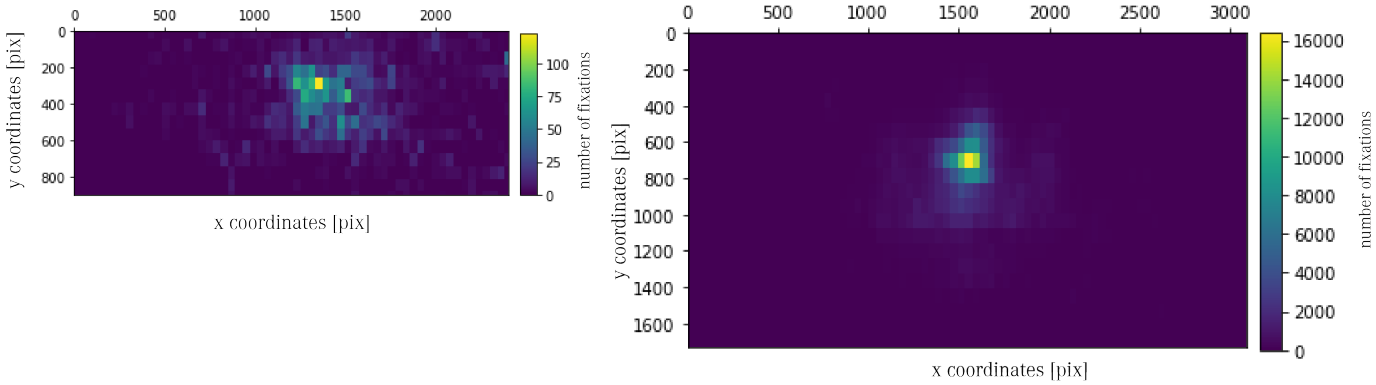
Figure 10 depicts the mean noticing times per participant $\bar{t}_{Not}$, for both the static and gaze-contingent NTG7 interface. The use of a gaze-contingent NTG7 interface while interacting with in-vehicle technology and engaging in non-driving related tasks results in statistically significant decrease in mean noticing time $\bar{t}_{Not}$ at significance level $\alpha = 0.05$, tested with a Wilcoxon Signed Rank test.

In condition A, the mean noticing times averages over all participants to 1.603 $sec.$ Condition B, averages the mean noticing times of all participants to 0.332 $sec.$ This indicates a decrease in mean noticing times $\bar{t}_{Not}$ of 79.3% from condition A to condition B.

Furthermore, the participants seem to be more consistent in when they notice the stimulus when using the gaze-contingent interface compared to condition A with the static interface. This can be seen in the spread of data points in the box plot of fig. 10 and is also expressed in the standard deviation of both conditions, where condition A has a standard deviation of 0.915 and condition B of 0.152.



(a) Condition A: heat map of all fixation locations for all participants executing the primary and secondary task of the experiment using the static NTG7 interface.



(b) Condition B: heat map of all fixation locations for all participants executing the primary and secondary task of the experiment using the gaze-contingent NTG7 interface.

Fig. 9: Heat map with on the left the instrument cluster and the head unit on the right.

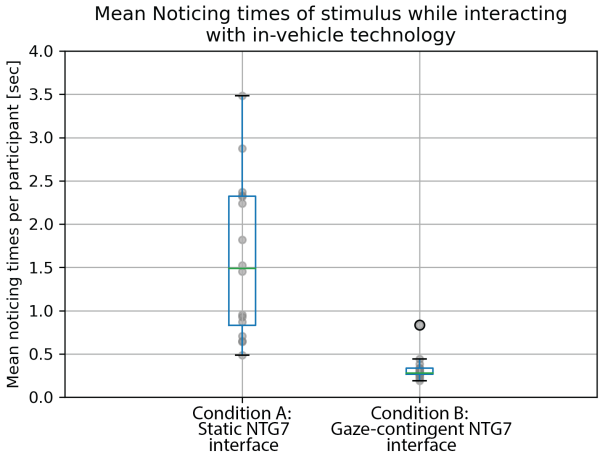


Fig. 10: Comparing the mean noticing times $\bar{t}_{Not,S}$ of the Hands-On-Wheel warning message in a static NTG7 interface to the noticing time $\bar{t}_{Not,GC}$ in the NTG7 gaze-contingent interface.

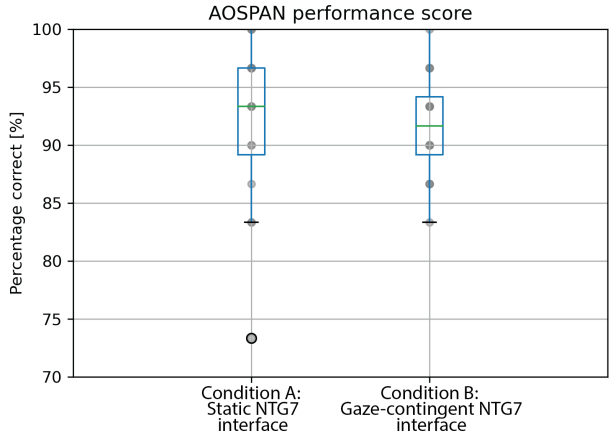
D. Performance of non-driving related tasks

Figure 11 depicts the performance of both non-driving related tasks performed in the experiment. In fig. 11a, it can be seen that average performance of the AOSPAN task is slightly higher in condition A. Using the static NTG7 interface, the participants have answered 92.1% of the asked questions correctly. With the gaze-contingent NTG7 interface, there is a slight decrease in performance with 91.7% of the asked questions answered correctly. When comparing the AOSPAN performance in both conditions, using the Wilcoxon Signed Rank test, it can be found that the difference in performance is not statistically significant.

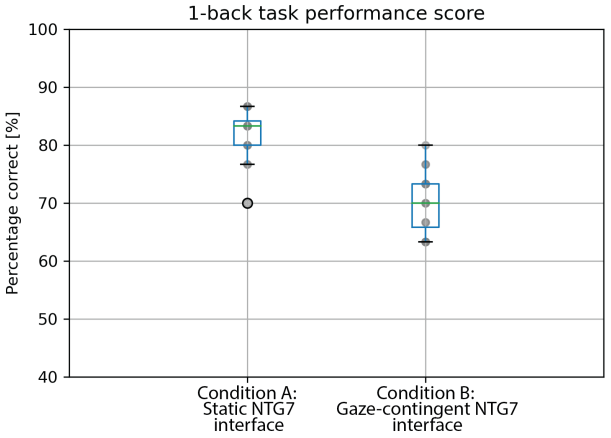
For the 1-Back task, there seems to be the same trend in performance as with the AOSPAN but with a bigger difference between the two conditions, as seen in fig. 11. In condition A, where the stimulus is presented on the instrument cluster, the participants perform better on average. This is also backed by the statistical significance found between the data sets of condition A and B using the Wilcoxon Signed Rank test at significance level 0.05. On average, the participants recollect 81.9% of the correct letters when taking the 1-back task in the static NTG7 interface. This is 11.7% higher than the recollection of letters in the gaze-contingent interface.

E. Percentage of stimulus seen through peripheral vision, PV [%]

Figure 12 shows the amount of times the stimulus is assumed to be seen through peripheral vision (as is defined in



(a) The performance of the AOSPAN task expressed in a percentage of correctly answered questions, including correctly answered mathematical problems and recollected letters.



(b) The performance of the 1-Back task expressed in a percentage of correctly recollected letters.

Fig. 11: Performance of non-driving related task on the head unit.

chapter II-E and the psuedo code of fig. 8). This is depicted for both tested conditions of the static and gaze-contingent NTG7 interface. In condition A, using the static interface, on average the participants nudge away the Hands-On-Wheel message before fixating 30.1% of the times when presented with the stimulus. In condition B, using the gaze-contingent interface, this amount was 12.4% less with only a 17.7% of the stimuli being seen through peripheral vision.

However, the Wilcoxon Signed Rank test concludes that between the data sets of both conditions lies no statistical significant difference.

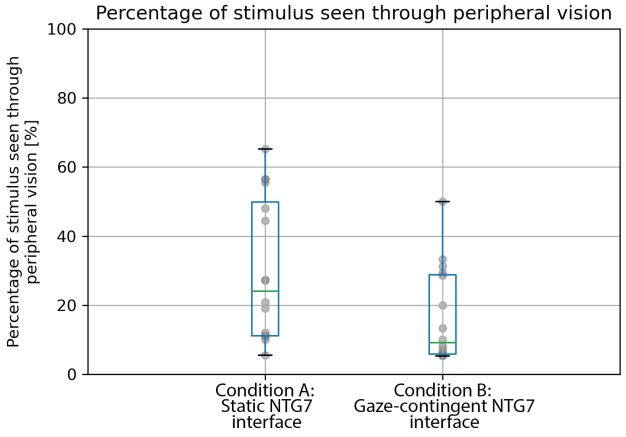


Fig. 12: Percentage of stimulus seen through peripheral vision in condition A and B.

IV. Discussion

A. Mean noticing times of stimulus, $\bar{t}_{Not}$

The aim of this research was to reduce the noticing time of Hands-On-Wheel warning message for drivers of Level 2 vehicles while interacting with in-vehicle technology through the implementation of gaze-contingent placement of time critical warning messages.

The results of the experiment show that the mean noticing time of the stimulus is 79.3% lower when placing the Hands-On-Wheel warning message on the display which the participant is currently looking at. This is the same trend as found by Pomarjanschi et al. [11], where participants' time to first fixation of a time critical event showed a decrease when using gaze-contingent cues.

Pomarjanschi et al. [11] have used gaze contingent guidance in simulated driving to guide the gaze of drivers in safety critical situations. Similar to this experiment, the drivers in Pomarjanschi et al. [11] were also subjected to a secondary (cognitive) tasks to divert the attention of the drivers from the road. Using gaze contingent guidance to call attention to potentially hazardous events resulted in shorter noticing times, safer driving behaviour and fewer collisions [11].

This trend found in the results of this experiment (and in Pomarjanschi et al. [11]) is concurrent with the theories of vigilance decrement and mainly justifies the longer noticing times for the stimulus in the static NTG7 interface. Mackworth [21] established this phenomenon in 1948 when researching human vigilance with his clock experiment, where participants had to detect stimuli in a clock presented at regular time intervals. This experiment mimicked the vigilance tasks corresponding to look-out duties in the air force and established that humans become increasingly bad at detecting stimuli during tasks where not a lot changes [21]. In the case of using static interfaces for the display of time critical warning messages, the vigilance decrement would explain the longer noticing times when the stimulus is presented in a static manner, as this corresponds with the findings in Mackworth's [21] research.

The trend of longer noticing times in the static NTG7 interface due to vigilance decrement can be justified through

the occurrence of habituation to the presented stimuli. Habituation is a mechanism that happens in the nervous system of a human, where the human becomes less responsive to a repeated stimulus [22], such as the Hands-On-Wheel warning message in the case of this experiment. What happens there is that the human brain continuously makes and adjusts a model of when a stimulus could present itself based on past events. As humans, we try to extrapolate this mental model into the future to try and predict when another incoming stimulus could present itself. If this prediction and the reality coincide, than the neural response to this stimulus is less [23]. This is what happens when the stimulus is presented in the static NTG7 interface, in which the stimulus is presented in a regular and unchanged pattern according to fig. 5. Where even the placement of the stimulus remains unchanged.

The results of this experiment indicate that this vigilance decrement is less of an issue in the case of the gaze-contingent NTG7 interface, even though the stimulus is presented according to the same timeline as in the static NTG7 interface. This can be justified by a result from the same 1948 clock experiment of Mackworth. Namely, it could be concluded from his experiment that vigilance decrement can be circumvented through an interruption between long periods of vigilance. In the clock experiment, Mackworth noticed that the participants were able to notice more stimuli when interrupted in their task through the means of a phone call [21]. Essentially, such an interruption is mimicked by the gaze-contingent placement of the Hands-On-Wheel warning message while the participants complete the secondary tasks on the head unit. The placement of the stimulus on the head unit while the participant is looking at said display reduces the resource cost [24] of the primary task of noticing the Hands-On-Wheel warning message. According to cognitive resource theory [24], reducing the work load of the primary task by gaze-contingent placement of the stimulus would minimize the effect of vigilance decrement and result in lower noticing times of the stimulus.

However, the decrease in noticing times of the stimuli for the two interfaces is multiple factors larger than as found in literature of Pomarjanschi et al. [11], where the use of gaze-contingent guidance only resulted in a 23.0% decrease in noticing times. This difference between the effect on the noticing times could be due to the nature of the gaze-contingent cue. Pomarjanschi et al. [11] make use of temporally transient gaze-contingent cue that guides the gaze from the current gaze position to the target/stimulus. This research however, uses the current gaze position at $t_{HOW,s}$ to determine the placement of the stimulus to try and minimize gaze movement of the participant. This means that guiding the gaze from any location to the stimulus, as in Pomarjanchi et al. [11], is not needed. Therefore, the corresponding time it would take to move gaze would be minimized or even eliminated.

However, to validate this reasoning, additional research needs to be done on the effect of the gaze-contingent cue design on noticing times. The research done by Loschky et al. [25] could in very broad lines insinuate something similar. Their research into spatial, resolutional, and temporal parameters affecting gaze behaviour in gaze-contingent displays concludes that spatial and resolutional parameters of a gaze-contingent cue has an effect on the gaze performance of a subject when detecting objects. Their results show that with a decreasing spatial window size and an increasing blur of the peripheral surroundings, the more likely it is for someone's gaze to be fixated to the window of high resolution [25].

When translating this concept of Loschky et al. [25] to the comparison between this research paper and that of Pomarjanchi et al. [11], one can consider the Hands-On-Wheel warning message to have a smaller spatial window and and an increasing blur of the peripheral surroundings as compared to the gaze-contingent cue Pomarjanchi et al. [11]. This could justify the big difference in decrease in mean noticing times between the two researches.

However, this reasoning is deduced to a higher abstraction level through the means of Loscky et al.'s [25] research and does not necessarily translate directly into driving behaviour. Therefore, any conclusions from this are considered to be a nudge in the right direction but still premature for any definitive justification. Therefore, it is advised to further look into the effect gaze-contingent stimulus design on mean noticing times.

Another interesting finding is that there is less variability in the mean noticing times per participant when using the gaze-contingent NTG7 interface. This could be due to the intrusive nature of this interface. As the participant is completing the non-driving related tasks, the gaze-contingent interface will always interrupt the exercise, forcing the participant to fixate on the Hands-On-Wheel warning message. However, when using the static NTG7 interface, the stimulus is always placed on the instrument cluster. This not only results in scenarios where the participant does not see the stimulus when it pops up, it also gives the participant the choice to either fixate on the stimulus on the instrument cluster right away, or to first finish their sub task (e.g. one mathematical question of the AOSPAN task) and then resume to fixate towards the instrument cluster. This multitude of options is the reason why a bigger deviation of noticing times can be justified.

B. Performance of non-driving related tasks

The performance of the non-driving related tasks from this experiment give the same decreasing trend for each condition in which they were tested. Using the static NTG7 interface, the participants performed better in both the non-driving related tasks. However, the performance of the AOSPAN did not show any significant difference in performance under both conditions, while the performance of the 1-Back was performed worse in the gaze-contingent interface.

In condition B, the participants were interrupted by the placement of the stimulus at $t_{HOW,s}$ on the same display as their NDRT. This intrusive, yet gaze-contingent, way of displaying the stimulus would create the expectation of a worse performance in NDRT, assuming that the sudden display of the stimulus would break the participants' concentration. The performance data of the AOSPAN task does show this

trend slightly but it is not a statistically significant trend. The performance of the 1-Back task however does depict this trend more clearly.

The reason why the performance of the NDRT's is lower could be due to the interruptions introduced by the stimulus in the gaze-contingent interface. Even though the gaze-contingent placement of the stimuli reduces task load for the primary task [24], these interruptions increase cognitive workload [26] for the participants by demanding that they store information in their working memory regarding the NDRT while they attend to the Hands-On-Wheel warning message. According to Drews and Musters [26], people with a low working memory capacity, are more prone to the negative effects of the interruptions and would show a tendency to perform worse in the secondary task. This is however something that needs to be established beforehand from the participants and cannot be concluded given the results of the performance of the NDRT's. However, the phenomenon discovered by Drews and Musters [26] can be consciously taken into account when further developing the gaze-contingent placement of time-critical warning messages.

C. Percentage of stimulus seen through peripheral vision

During the experiments, it has been observed that occasionally the participants react to the Hands-On-Wheel warning message without actually fixating on the stimulus. One would expect that this would occur more frequently in condition A, since the participants are mainly focused on the head unit with the execution of the secondary task. However, there seems to be no statistical difference in the proportion of times the stimulus is seen through the peripheral vision.

D. Conclusion and recommendations

The results of this experiment indicate lower noticing times of the Hands-On-Wheel warning message when the stimulus is placed in a gaze-contingent manner, while the participants engage in secondary tasks on the in-vehicle technology. The participants that use the gaze-contingent NTG7 interface for executing the primary task of putting their hands on the steering control only take $0.332\ sec$ on average to notice the stimulus. When using the static NTG7 interface, the mean noticing time of the stimuli is higher at $1.603sec$.

These results indicate that the vigilance decrement that occurs when using the static NTG7 interface can be circumvented by gaze-contingent placement of the Hands-On-Wheel warning message. The placement of the stimulus on the head unit when the participant is already looking at it reduces the primary task load and causes for the stimulus to be seen quicker as compared to the static NTG7 interface.

However, the performance of the secondary task seems to decrease when using the gaze-contingent NTG7 interface. The 1-Back task conclusively shows a decrease of 11.7% in performance in the gaze-contingent interface. The AOSPAN task shows a similar trend however this is considered to be statistically insignificant. The reason for lower performance in secondary task is due to the nature of the intrusive interruption caused by the gaze-contingent placement of the stimuli. This causes an increase in cognitive workload for the secondary task, as this interruption forces the participants to store information in their working memory as they attend to the primary task.

The current results from this research provides the opportunity to give real-time gaze-contingent feedback to the driver. However, this executed research also has some shortcomings that could be elaborated upon in further research. Firstly, the experiment can be expanded to include reaction time as dependent variable instead of solely focusing on noticing time. That way you can more realistically measure the effect of gaze-contingent feedback in time-critical situations that also require an action from the participant. As mentioned in section I, the action required when a driver is exposed to a take over request does not only consist of the noticing time but also the Hands-On-Wheel time and intervention time. To get a better understanding of what the effect is of the gaze-contingent placement of time critical warning messages on the complete take over process, the two other stages ought to be measured as well. This can be done by recording the steering control input during the experiment as well.

Furthermore, this experiment is limited to the distraction of the participant by the secondary tasks solely on the head unit. In practice, this meant that the participant's gaze was mostly on the head unit throughout the secondary task and therefore the stimulus was always placed on the head unit as well. Therefore, the gaze-contingent placement of the stimulus was not tested on the instrument cluster because the participant's gaze was never needed there to complete the secondary tasks or for any other reason. Hence, this research could benefit from the expansion of its scope by including interaction with other displays such as the instrument cluster, passenger display and heads up display such that the Hands-On-Wheel warning message is not only displayed on the head unit.

Despite the limitations of this research, the found results could still have implications on the design of the interfaces that display time critical warning messages under Level 2 driving conditions. With a general tendency of drivers to engage more with in-vehicle technology when using ADS functions [3], the use of gaze-contingent interfaces can clearly reduce the noticing time of time critical warning messages instead of the seen longer reaction times in research on take-over requests [6]. The reduction of noticing times of such messages when its placed in a gaze-contingent manner could be beneficial to the safety of autonomous driving functions where the driver has a vigilance task and is engaging in secondary tasks.

REFERENCES

[1] "SAE Levels of Driving Automation™ Refined for Clarity and International Audience." [Online]. Available: https://www.sae.org/site/blog/sae-j3016-update

[2] J. M. Anderson, N. Kalra, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola, *Autonomous vehicle technology: a guide for policymakers*. Santa Monica, CA: Rand Corporation, 2014, oCLC: ocn867840251.

[3] O. Carsten, F. C. H. Lai, Y. Barnard, A. H. Jamson, and N. Merat, "Control Task Substitution in Semiautomated Driving: Does It Matter What Aspects Are Automated?" *Human Factors*, vol. 54, no. 5, pp. 747–761, Oct. 2012, publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.1177/0018720812460246

[4] V. A. Banks, A. Eriksson, J. O'Donoghue, and N. A. Stanton, "Is partially automated driving a bad idea? Observations from an on-road study," *Applied Ergonomics*, vol. 68, pp. 138–145, Apr. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687017302594

[5] C. Gold, D. Damböck, L. Lorenz, and K. Bengler, ""Take over!" How long does it take to get the driver back into the loop?" *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 1938–1942, Sep. 2013, publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.1177/1541931213571433

[6] A. Eriksson and N. A. Stanton, "Takeover Time in Highly Automated Vehicles: Noncritical Transitions to and From Manual Control," *Human Factors*, vol. 59, no. 4, pp. 689–705, Jun. 2017.

[7] J. Radlmayr, C. Gold, L. Lorenz, M. Farid, and K. Bengler, "How Traffic Situations and Non-Driving Related Tasks Affect the Take-Over Quality in Highly Automated Driving," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 2063–2067, Sep. 2014. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1541931214581434

[8] J. F. Mackworth, "Deterioration of signal detectability during a vigilance task as a function of background event rate I," p. 2.

[9] L. Bainbridge, "Ironies of automation," *Automatica*, vol. 19, no. 6, pp. 775–779, Nov. 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109883900468

[10] P. O. o. t. E. Union, "Regulation No 79 of the Economic Commission for Europe of the United Nations (UN/ECE) — Uniform provisions concerning the approval of vehicles with regard to steering equipment [2018/1947]," Dec. 2018, publisher: Publications Office of the European Union. [Online]. Available: http://op.europa.eu/en/publication-detail/-/publication/3f7415ac-ff32-11e8-a96d-01aa75ed71a1

[11] L. Pomarjanschi, M. Dorr, and E. Barth, "Gaze guidance reduces the number of collisions with pedestrians in a driving simulator," *ACM Transactions on Interactive Intelligent Systems*, vol. 1, no. 2, pp. 8:1–8:14, Jan. 2012. [Online]. Available: https://doi.org/10.1145/2070719.2070721

[12] D. Pool, "Objective Evaluation of Flight Simulator Motion Cueing Fidelity Through a Cybernetic Approach," Ph.D. dissertation, Sep. 2012.

[13] A. Steinfeld, O. C. Jenkins, and B. Scassellati, "The oz of wizard: simulating the human for interaction research," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction - HRI '09*. La Jolla, California, USA: ACM Press, 2009, p. 101. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1514095.1514115

[14] N. Unsworth, R. P. Heitz, J. C. Schrock, and R. W. Engle, "An automated version of the operation span task," *Behavior Research Methods*, vol. 37, no. 3, pp. 498–505, Aug. 2005. [Online]. Available: http://link.springer.com/10.3758/BF03192720

[15] J. C. Felver-Gant, A. S. Bruce, M. Zimmerman, L. H. Sweet, R. P. Millman, and M. S. Aloia, "Working memory in obstructive sleep apnea: construct validity and treatment effects," *Journal of clinical sleep medicine: JCSM: official publication of the American Academy of Sleep Medicine*, vol. 3, no. 6, pp. 589–594, Oct. 2007.

[16] W. Chen, T. Sawaragi, and Y. Horiguchi, "Measurement of Driver's Mental Workload in Partial Autonomous Driving," *IFAC-PapersOnLine*, vol. 52, no. 19, pp. 347–352, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319319160

[17] Y. B. Eisma, C. D. D. Cabrall, and J. C. F. de Winter, "Visual Sampling Processes Revisited: Replicating and Extending Senders (1983) Using Modern Eye-Tracking Equipment," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 526–540, Oct. 2018, conference Name: IEEE Transactions on Human-Machine Systems.

[18] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," *Behavior Research Methods*, vol. 42, no. 1, pp. 188–204, Feb. 2010. [Online]. Available: https://doi.org/10.3758/BRM.42.1.188

[19] K. Rayner, "The 35th Sir Frederick Bartlett Lecture: Eye movements and attention in reading, scene perception, and visual search," *Quarterly Journal of Experimental Psychology*, vol. 62, no. 8, pp. 1457–1506, Aug. 2009, publisher: SAGE Publications. [Online]. Available: https://doi.org/10.1080/17470210902816461

[20] J. C. de Winter and D. Dodou, *Human Subject Research for Engineers*, ser. SpringerBriefs in Applied Sciences and Technology. Cham: Springer International Publishing, 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-56964-2

[21] N. H. Mackworth, "The Breakdown of Vigilance during Prolonged Visual Search," *Quarterly Journal of Experimental Psychology*, vol. 1, no. 1, pp. 6–21, Apr. 1948. [Online]. Available: http://journals.sagepub.com/doi/10.1080/17470214808416738

[22] "Habituation - an overview | ScienceDirect Topics." [Online]. Available: https://www.sciencedirect.com/topics/medicine-and-dentistry/habituation

[23] J. F. Mackworth, "Vigilance, arousal, and habituation," *Psychological Review*, vol. 75, no. 4, pp. 308–322, Jul. 1968.

[24] J. Flanagan and D. Nathan-Roberts, "Theories of Vigilance and the Prospect of Cognitive Restoration," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 63, no. 1, pp. 1639–1643, Nov. 2019, publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.1177/1071181319631506

[25] L. C. Loschky and G. W. McConkie, "User performance with gaze contingent multiresolutional displays," in *Proceedings of the symposium on Eye tracking research & applications - ETRA '00*. Palm Beach Gardens, Florida, United States: ACM Press, 2000, pp. 97–103. [Online]. Available: http://portal.acm.org/citation.cfm?doid=355017.355032

[26] F. A. Drews and A. Musters, "Individual differences in interrupted task performance: One size does not fit all," *International Journal of Human-Computer Studies*, vol. 79, pp. 97–105, Jul. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581915000063

[27] J. Gu, M. Meng, A. Cook, and M. G. Faulkner, "Analysis of eye tracking movements using FIR median hybrid filters," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, ser. ETRA '00. New York, NY, USA: Association for Computing Machinery, Nov. 2000, pp. 65–69. [Online]. Available: https://doi.org/10.1145/355017.355027

V. Appendix

A. Interface prototyping

The interface has been built up in different phases. The design development is done through the use of the Mercedes-Benz style guide, the interactional development is done in ProtoPie, the eye-tracking integration is done through the means of the Mercedes-Benz in-house developed GazeConnect, the communication between the interfaces of the head unit and instrument cluster is established through ProtoPie connect, the execution of the prototypes in the seating buck is done through ProtoPie Runner and the logging of all the eye-tracking data is done through the Mercedes-Benz in-house developed Data Logger. These programs will briefly be discussed in this appendix to give an idea of the development cycle of the interface. Figure 13 summarizes the development cycle.

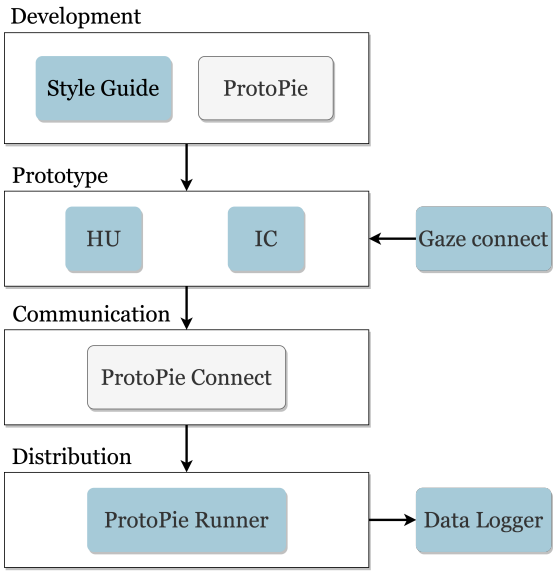


Fig. 13: Development cycle of the interface prototypes used in the experiment. The blue icons indicate Mercedes-Benz in-house developed programs and the grey icons are off the shelve prototyping programs.

The design of the interface comprises of style elements that are available in the Mercedes-Benz style guide. This guide contains all elements that are used to build the NTG7 interface, as individual PNG elements that can be altered according to Mercedes-Benz' graphic design to suit specific use. In this case that meant using the style guide to export things such as buttons for the AOSPAN/1-Back task, menu bars, and ADS indicators.

These interface elements are then imported into ProtoPie, which is an interface prototyping tool for any digital products. The imported buttons can be triggered by touch (through touch screen) or any other type of modalities. ProtoPie can then link an action to a trigger, such as the movement of an image layer, jump to a different scene, send/receive messages or even assign values to global variables. These actions can also be made conditional based on the values of variables. Figure 14 depicts a screenshot of the ProtoPie program to give an idea of how interactions are built.

For this experiment, ProtoPie was used for the screen interactions and also the gaze-contingent aspect of the interface on both the head unit and instrument cluster. That meant that each display had to have their own ProtoPie file which had to communicate with each other. This could be done through ProtoPie Connect, which connects various interfaces that run on multiple devices such that they can communicate. The communication between the devices is achieved through send and receive messages where ProtoPie Studio (ProtoPie Connect) acts a bridge between the two prototypes.

The gaze-contingent element makes use of this external communication system set up by ProtoPie connect. It revolves around the eye-tracking measurements that are communicated to the prototypes by the Mercedes-Benz in-house developed GazeConnect. The eye-tracking measurements are then locally stored in a variable in each prototype. The gaze-contingent element consists of a *DETECT* trigger, which triggers a response when it detects a variable change. In the case of the gaze-contingent element, this variable is the one that eye-tracking measurements are assigned to. If the gaze is detected on that prototype, it assigns a 1 to the variable "Gaze detected" and simultaneously sends a message to the other prototype that assigns a 0 to the variable "Gaze detected". Then, as the variable "Time" elapses, at $t_{HOW,s}$ the variable "Gaze detected" is checked. Using a conditional element in ProtoPie, the Hands-On-Wheel warning message is then triggered at $t_{HOW,s}$ on the display which has a 1 assigned to the variable "Gaze detected".

The individual head unit and instrument cluster prototypes are not only linked through ProtoPie Connect but also need to be assigned to the hardware in the seating buck. This is done through the in-house developed ProtoPie Runner software that is run from the Microsoft Surface in the seating buck. This piece of software distributes the ProtoPie files to either the head unit, instrument cluster and even passenger display.
Lastly, all the sent and received messages through ProtoPie Connect are logged with the in-house developed Data Logger. This file contains all the data necessary for the data analysis. Figure 15 shows a snippet of the logged data of one of the participants.

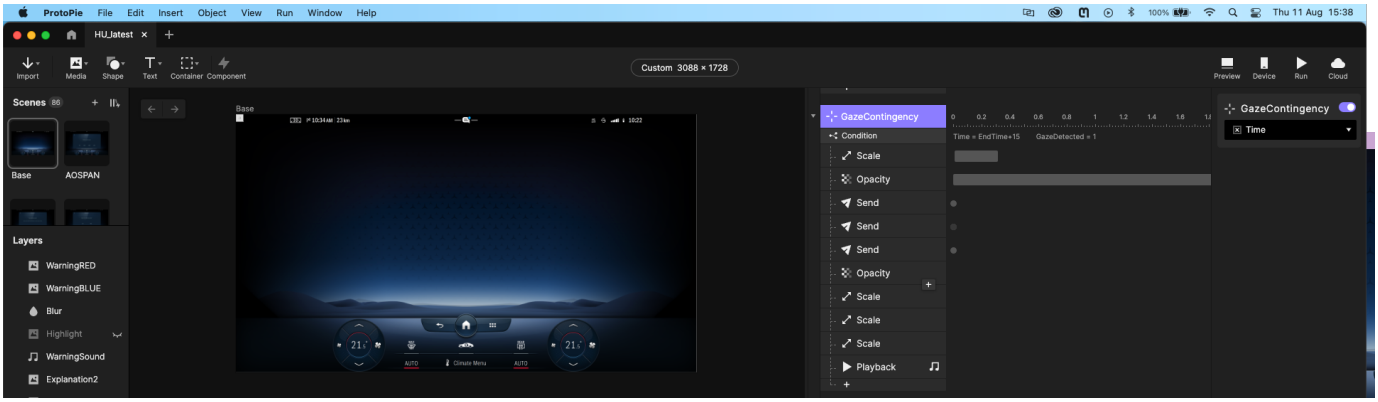


Fig. 14: ProtoPie development of the gaze-contingent element.


```
<Log time="28-02-22_08-21-51_344"><Message timestamp="28-02-22, 08:21:51.322211" messageId="startLog" /><Message
timestamp="28-02-22, 08:21:51.346217" messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22,
08:21:51.348226" messageId="GazeConnect.gazeX">1609.0523893453985</Message><Message timestamp="28-02-22,
08:21:51.348226" messageId="GazeConnect.gazeY">965.7057076449689</Message><Message timestamp="28-02-22,
08:21:51.349222" messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.350214"
messageId="GazeConnect.gazeX">1608.1621155612288</Message><Message timestamp="28-02-22, 08:21:51.350214"
messageId="GazeConnect.gazeY">966.4899422541512</Message><Message timestamp="28-02-22, 08:21:51.361214"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.363216"
messageId="GazeConnect.gazeX">1607.2784960022534</Message><Message timestamp="28-02-22, 08:21:51.364217"
messageId="GazeConnect.gazeY">967.3567442786486</Message><Message timestamp="28-02-22, 08:21:51.374219"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.379221"
messageId="GazeConnect.gazeX">1604.4783543829067</Message><Message timestamp="28-02-22, 08:21:51.381218"
messageId="GazeConnect.gazeY">969.5124450877042</Message><Message timestamp="28-02-22, 08:21:51.391225"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.397218"
messageId="GazeConnect.gazeX">1602.3858485850908</Message><Message timestamp="28-02-22, 08:21:51.399217"
messageId="GazeConnect.gazeY">971.1737087037645</Message><Message timestamp="28-02-22, 08:21:51.417214"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.419217"
messageId="GazeConnect.gazeX">1601.4557653667225</Message><Message timestamp="28-02-22, 08:21:51.422225"
messageId="GazeConnect.gazeY">971.3460540515829</Message><Message timestamp="28-02-22, 08:21:51.424217"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.429221"
messageId="GazeConnect.gazeX">1598.8031948910223</Message><Message timestamp="28-02-22, 08:21:51.434219"
messageId="GazeConnect.gazeY">970.7403011259158</Message><Message timestamp="28-02-22, 08:21:51.443252"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.448214"
messageId="GazeConnect.gazeX">1598.0269410387145</Message><Message timestamp="28-02-22, 08:21:51.451217"
messageId="GazeConnect.gazeY">970.8776641994167</Message><Message timestamp="28-02-22, 08:21:51.458218"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.464218"
messageId="GazeConnect.gazeX">1597.2828230101122</Message><Message timestamp="28-02-22, 08:21:51.469216"
messageId="GazeConnect.gazeY">971.0112653653207</Message><Message timestamp="28-02-22, 08:21:51.475219"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.480216"
messageId="GazeConnect.gazeNotAvailable" /><Message timestamp="28-02-22, 08:21:51.494216"
messageId="GazeConnect.gazeAvailable" /><Message timestamp="28-02-22, 08:21:51.496213"
messageId="GazeConnect.gazeX">1593.0413524066164</Message><Message timestamp="28-02-22, 08:21:51.497211"
messageId="GazeConnect.gazeY">973.023691655367</Message><Message timestamp="28-02-22, 08:21:51.506216"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.519218"
messageId="GazeConnect.gazeX">1591.5409395111585</Message><Message timestamp="28-02-22, 08:21:51.520211"
messageId="GazeConnect.gazeY">973.6219487620167</Message><Message timestamp="28-02-22, 08:21:51.522212"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.526210"
messageId="GazeConnect.gazeX">1591.00626385916263</Message><Message timestamp="28-02-22, 08:21:51.528212"
messageId="GazeConnect.gazeY">973.7665342518529</Message><Message timestamp="28-02-22, 08:21:51.539211"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.544217"
messageId="GazeConnect.gazeX">1588.0418788261495</Message><Message timestamp="28-02-22, 08:21:51.545214"
messageId="GazeConnect.gazeY">972.6963628045045</Message><Message timestamp="28-02-22, 08:21:51.556212"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.560212"
messageId="GazeConnect.gazeX">1586.7403041442894</Message><Message timestamp="28-02-22, 08:21:51.561211"
messageId="GazeConnect.gazeY">973.5946778503073</Message><Message timestamp="28-02-22, 08:21:51.571213"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.576214"
messageId="GazeConnect.gazeX">1586.0742675212614</Message><Message timestamp="28-02-22, 08:21:51.577210"
messageId="GazeConnect.gazeY">975.614339741899</Message><Message timestamp="28-02-22, 08:21:51.588210"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.592210"
messageId="GazeConnect.gazeX">1585.5267857413724</Message><Message timestamp="28-02-22, 08:21:51.593217"
messageId="GazeConnect.gazeY">976.006080637456134</Message><Message timestamp="28-02-22, 08:21:51.605214"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.611218"
messageId="GazeConnect.gazeX">1584.999225433754</Message><Message timestamp="28-02-22, 08:21:51.626215"
messageId="GazeConnect.gazeY">976.4522155723712</Message><Message timestamp="28-02-22, 08:21:51.628219"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.631219"
messageId="GazeConnect.gazeX">1584.465700843906</Message><Message timestamp="28-02-22, 08:21:51.636215"
messageId="GazeConnect.gazeY">977.5044947470525</Message><Message timestamp="28-02-22, 08:21:51.641230"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.648218"
messageId="GazeConnect.gazeX">1583.943593424223</Message><Message timestamp="28-02-22, 08:21:51.650215"
messageId="GazeConnect.gazeY">978.0262273571831</Message><Message timestamp="28-02-22, 08:21:51.657221"
messageId="GazeNotDetected">0</Message><Message timestamp="28-02-22, 08:21:51.662215"
```


Fig. 15: The text file that logs all the communicated messages between the ProtoPie files, including the timestamps at which these messages are sent

B. Performance of non-driving related tasks

In table III, the individual scores to the questions posed to the participants in the non-driving related tasks are displayed. Each participant has executed all the trials twice. Once in the gaze-contingent interface and once in the static interface. The scores are therefore also shown in separate rows of table III.

For the AOSPAN task, this score is out of 10 for each trial. This consists of 5 mathematical problems and 5 letters that have to be recollected. The 1-Back task score is out of 9 for each trial as the participants compare 10 letters to each other and thus answering 9 questions whether these letters are the same or not.

TABLE III: Individual performance score of the AOSPAN and 1-Back task.

		AOSPAN			1-Back		
		T1	T2	T3	T1	T2	T3
Participant 1	S	10	10	10	8	9	8
	G	10	10	10	9	9	8
Participant 2	S	10	10	10	9	9	7
	G	9	9	9	7	9	8
Participant 3	S	10	10	10	4	6	8
	G	10	10	10	7	8	8
Participant 4	S	10	10	10	6	7	6
	G	9	9	9	9	8	8
Participant 5	S	9	10	10	9	7	9
	G	10	10	10	9	9	7
Participant 6	S	8	8	8	7	9	9
	G	10	10	10	8	9	9
Participant 7	S	10	10	10	7	9	8
	G	10	10	10	8	9	9
Participant 8	S	10	10	10	9	8	9
	G	10	9	9	8	9	9
Participant 9	S	8	9	9	9	8	7
	G	9	9	9	6	8	6
Participant 10	S	10	10	10	9	9	9
	G	9	9	9	8	8	9
Participant 11	S	9	8	8	9	8	8
	G	10	9	9	9	9	9
Participant 12	S	10	10	10	8	8	7
	G	10	10	10	8	7	7
Participant 13	S	10	10	10	9	7	9
	G	9	9	9	9	8	9
Participant 14	S	9	9	10	9	8	9
	G	10	10	10	8	8	8
Participant 15	S	8	8	8	9	7	9
	G	9	8	8	7	8	8
Participant 16	S	9	9	9	9	9	7
	G	10	10	10	9	8	9

C. Display specifications and positioning in the seating buck

As mentioned in II-B, the interface comprises of the head unit and instrument cluster. The instrument cluster used in the seating buck is the same that is used in Mercedes-Benz series production cars. The head unit is a Microsoft Surface

Pro 7 which has a touch screen mimicking the screen as in the Mercedes-Benz series production cars. Table IV shows the dimensions of the used displays in centimeters and in pixels. These values are used in eq. 2, to determine the factor that converts pixels to visual degrees.

The placement of the displays are measured as the distance in centimeters from the head rest of the seat in the seating buck to the screens. The distance between the centre of the head unit and the centre of the instrument is $29.3cm$. The other dimensions can be found in table IV

TABLE IV: Display specifications of the interface.

	Instrument cluster	Head unit
Width [cm], w_{cm}	29.3	31.2
Height [cm], h_{cm}	11.0	23.2
Width [pix], w_{pix}	2400	3088
Height [pix], h_{pix}	900	1728
Distance to seat [cm], D	71.5	78.6

D. Procedure of the experiment

1) Open the Data Logger.
2) Distribute the correct ProtoPie files.
3) Open GazeConnect and make sure it is connected to the correct ProtoPie files.
4) Receive the planned participant.
5) Have the participant sign both consent forms.
6) Tell the participant to take a seat in the front left seat of the seating buck.
7) Explain what this experiment is about. Guide text is as follows: First of all, I would like to thank you for participating in this experiment. Today we want to test two different interfaces under specific conditions. What we want to simulate is level 2 driving. That means that there is automated lane keeping and adaptive cruise control. So basically that means that when you drive on the highway, you don't need to press the gas or break pedals and also not use the steering wheel. Under current law however, you are required to every once in a while touch the steering wheel. But don't worry, the interface will tell you when to do this by showing you this image. [show exemplary image of Hands-On-Wheel warning message]. In this experiment, the way we present you this specific warning message will change. In the meantime, you will have to do some small tasks on this screen here [point to Head Unit]. I will elaborate on the specifics of the task a bit later on. There are also two eye-trackers that you might have noticed. There is one above the steering wheel and one underneath the head unit. This eye-tracking data is used live to actually change the interface, but I also use this data to look at your gaze behavior and analyze that afterwards. Don't worry, it's not actually filming you. It is just tracking where you are looking on the display. So in order to get some good tracking data we need to calibrate both eye trackers. And we'll be doing that right now.
8) Calibrate both eye-trackers using the Tobii Pro eye-tracker manager.
9) Make sure that the Data Logger is running.
10) Start the correct interface according to the script order (static or gaze-contingent).
11) If starting with the gaze-contingent interface, disable the gaze-contingent element in the prototype until after the explanation of the tasks has been completed.
12) Explain both the AOSPAN and 1-back task using the trial levels.
13) Ask participants if they have understood everything. If so, we can start "driving".
14) Loop the "Driving video.mp4" on the external display.
15) Press "1" to start logging the eye-tracking data.
16) Press "E" to enable the gaze-contingent element in the ProtoPie file.
17) Participants starts level 1 of the AOSPAN task.
18) Participants starts level 2 of the AOSPAN task.
19) Participants starts level 3 of the AOSPAN task.
20) Participants starts level 1 of the 1-Back task.
21) Participants starts level 2 of the 1-Back task.
22) Participants starts level 3 of the 1-Back task.
23) Press "0" to end logging the eye-tracking data.
24) Make sure that the logging file from the Data Logger is saved.
25) Repeat steps 9-23 for the second interface according to the script order (static or gaze-contingent interface).
26) Label the saved Data Logger files according to which interface they belong.

E. Data analysis plan

This section of the appendix is dedicated to elaborate upon the data analysis plan for the eye-tracking data that has been executed to get from the measured x and y coordinates to noticing times of the Hands-On-Wheel warning message. In fig. 16, all the steps taken to get from the raw data to the noticing times are displayed graphically.

As mentioned before, the Data Logger creates a text file with all the communicated messages between the ProtoPie files. The first step of the data analysis is to extract the start time of the Hands-On-Wheel warning messages and eye-tracking data from the text file with their corresponding timestamps and append them into data frames.

From the looks of the data, it could be observed that occasionally the x and y coordinates of the same instance had slightly different timestamps. Therefore the timestamps of the x and y coordinate were averaged to the same time instance.

To filter the noise of the eye-tracking data, a median filter has been used on the x and y coordinates. A median filter of 100 ms interval has been chosen (size 6 at sampling frequency of 60 hz) to smooth out the data. This is due to the good edge preservation of median filters while it is still able to filter out noise on flat regions of the signal [27].

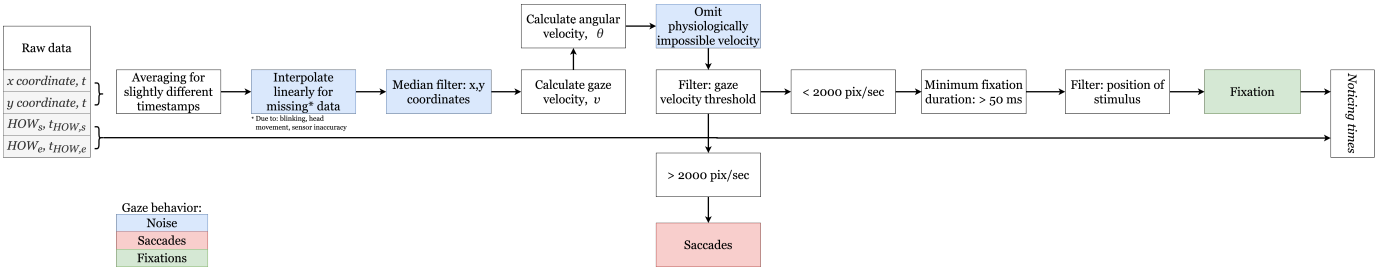


Fig. 16: Schematic representation of data filtering: from raw x and y coordinates to fixation on the stimulus.

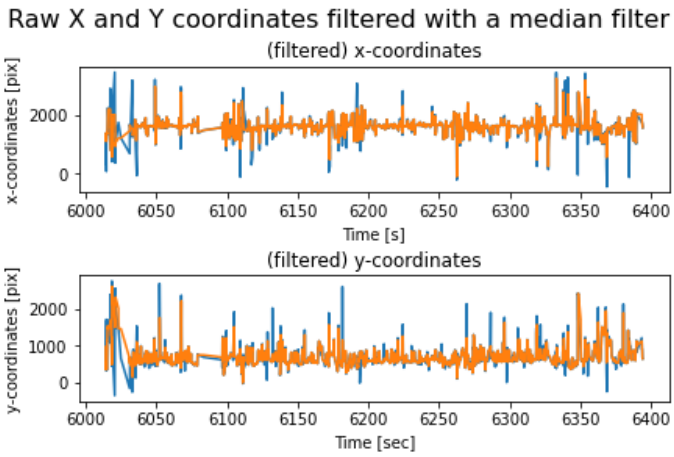


Fig. 17: The effect of the median filter on the raw x and y coordinates measured by the eye tracker, where the blue line is the raw data and orange is the filtered data of one participant.

From this filtered data, the gaze velocity is determined as the euclidean vector of the individual velocity components of the x and y coordinates. A graphical representation of this can be seen in fig. 18. (Angular) velocity and fixation duration thresholds are then applied to differentiate the saccades from the noise and fixations. From these fixations, the noticing times of the Hands-On-Wheel warning messages can be determined according to fig. 8.

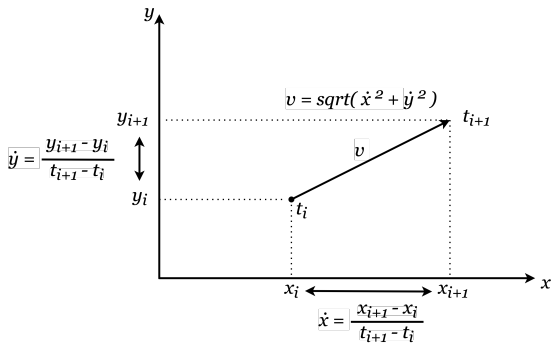


Fig. 18: Determining individual velocity of the x and y component of eye-tracking measurement, with the velocity defined as the euclidean vector of the individual velocity components.

F. Python code: data analysis of gaze-contingent NTG7 interface

```
#Rhita Addi - Msc. Vehicle Engineering - 4367391

# Importing the necessary libraries
import csv
import pandas as pd
from pandas import Timestamp
import numpy as np
from math import atan2, degrees
from scipy import ndimage, misc

import statistics as st
import matplotlib.pyplot as plt

# --- Constant variables ---
# d_hu: Distance from chair to Head Unit [cm]
# h_hu: Height of HU display in [cm]
# r_hu: Height of HU display in [pix]
# d_ic: Distance from chair to Instrument Cluster [cm]
# h_ic: Height of IC display in [cm]
# r_ic: Height of IC display in [pix]
# fs: Sampling frequency in [Hz]

d_hu = 78.6
h_hu = 23.2
r_hu = 1728
d_ic = 71.5
h_ic = 11.0
r_ic = 900

fs = 60

# Importing gaze contingent CSV file for each participant
# Participants(n); n = 1:16
raw_csv = [i for i in csv.reader(open("datasets/P16_g.csv"),delimiter=';')]

# --- Creating empty data structures ---
# hu_x, hu_y: Structures for x and y coordinates of the Head Unit (HU)
# ic_x, ic_y: Structures for x and y coordinated of the Instrument Cluster (IC)
# start: Structure for the timestamp at which HOW message starts
# e: Structure for the timestamp at which HOW message ends
# grouped_hu, grouped_ic: Structure for grouping coordinates per screen (HU/IC)
# i_hu, i_ic: Structure for indices per screen (HU/IC)

hu_x, hu_y = [], []
ic_x, ic_y  = [], []
start = []
e = []
grouped_hu, grouped_ic  = [], []
i_hu, i_ic= [], []


# Defining functions that read the data file and process it
# append() will place new items in the next available space
def process_hu_row(row):
```

```
    # HU-entries processing
    # Recognizinig GazeConnect.gaze... in the logging file
    # Adding timestamp and corresponding x and y values to hu_x/y
    # Adding no values to timestamp when no gaze is detected
    if row[0] == "GazeConnect.gazeX":
        hu_x.append((row[1],row[2]))
    if row[0] == "GazeConnect.gazeY":
        hu_y.append((row[1],row[2]))
    if row[0] == "GazeNotDetected":
        hu_x.append((row[1],None))
        hu_y.append((row[1],None))

def process_ic_row(row):
    # IC-entries processing
    # Recognizinig IC.gaze... in the logging file
    # Adding timestamp and corresponding x and y values to hu_x/y
    # Adding no values to timestamp when no gaze is detected
    if row[0] == "IC.gazeX":
        ic_x.append((row[1],row[2]))
    if row[0] == "IC.gazeY":
        ic_y.append((row[1],row[2]))
    if row[0] == "GazeNotDetected_IC":
        ic_x.append((row[1],None))
        ic_y.append((row[1],None))

def process_misc_entries(row):
    # Miscellaneous entries processing
    # Recognizing HOW_start in the logging file as HOW starting message
    # Adding corresponding timestamp to when HOW_start is found in file
    # Recognizing E in the logging file as HOW ending message
    # Adding corresponding timestamp to when E is found in file
    if row[0] == "HOW_start":
        start.append(row[1])
    if row[0] == "E":
        e.append(row[1])

def group_hu_entries():
# Grouping x, y and timestamp
    prev_no_detection = False
    for i in range(len(hu_y)):
        t1, t2 = Timestamp(hu_x[i][0].replace(',','')),Timestamp(hu_y[i][0].replace(',',''))
        x, y = None, None
        if not(hu_x[i][1] is None):
            x= float(hu_x[i][1].replace(',', '.'))
        if not(hu_y[i][1] is None):
            y = float(hu_y[i][1].replace(',', '.'))

        # Averaging slightly different timestamps for x and y coordinates
        avg = Timestamp((t1.value + t2.value) / 2.0)

        if ((x is None) or (y is None)):
            if not (prev_no_detection):
                grouped_hu.append((avg, x, y))
                prev_no_detection = True
        else:
            grouped_hu.append((avg, x, y))
```

```
            prev_no_detection = False

def group_ic_entries():
# Grouping x, y and timestamp
    prev_no_detection = False
    for i in range(len(ic_y)):
        t1, t2 = Timestamp(ic_x[i][0].replace(',','')),Timestamp(ic_y[i][0].replace(',',''))
        x, y = None, None
        if not(ic_x[i][1] is None):
            x= float(ic_x[i][1].replace(',', '.'))
        if not(ic_y[i][1] is None):
            y = float(ic_y[i][1].replace(',', '.'))

        # Averaging slightly different timestamps for x and y coordinates
        avg = Timestamp((t1.value + t2.value) / 2.0)

        if ((x is None) or (y is None)):
            if not(prev_no_detection):
                grouped_ic.append((avg, x, y))
                prev_no_detection = True
        else:
            grouped_ic.append((avg, x, y))
            prev_no_detection = False

def interpolate_hu():
# Linear interpolation between missing x and y data points
    for i in range(1,len(grouped_hu)):
        x,y = grouped_hu[i][1], grouped_hu[i][2]
        if ((x is None) or (y is None)):
            if not(i == (len(grouped_hu)-1)):
                i_x = float((grouped_hu[i-1][1] + grouped_hu[i+1][1])/2.0)
                i_y = float((grouped_hu[i - 1][2] + grouped_hu[i + 1][2]) / 2.0)
                i_t = grouped_hu[i][0]
                i_hu.append((i_t,i_x,i_y))
        else:
            i_hu.append(grouped_hu[i])

def interpolate_ic():
# Linear interpolation between missing x and y data points
    for i in range(1,len(grouped_ic)):
        x,y = grouped_ic[i][1], grouped_ic[i][2]
        if ((x is None) or (y is None)):
            if not(i == (len(grouped_ic)-1)):
                i_x = float((grouped_ic[i-1][1] + grouped_ic[i+1][1])/2.0)
                i_y = float((grouped_ic[i - 1][2] + grouped_ic[i + 1][2]) / 2.0)
                i_t = grouped_ic[i][0]
                i_ic.append((i_t,i_x,i_y))
        else:
            i_ic.append(grouped_ic[i])

def clean_data():
# Grouping the functions together that should clean all the raw data files
    for row in raw_csv[1:]:
        process_hu_row(row)
        process_ic_row(row)
        process_misc_entries(row)
```

```
    group_hu_entries()
    group_ic_entries()
    interpolate_hu()
    interpolate_ic()

# Execute all previously defined functions
clean_data()

# Converting the HOW start time from DateTimeIndex to total amount of seconds
start = pd.to_datetime(start)
Time_series = pd.Series(start)
MicroSeconds = ((Time_series.dt.hour)*360 + (Time_series.dt.minute)*60 +
                Time_series.dt.second)*1000000 + Time_series.dt.microsecond
Seconds = MicroSeconds/1000000

# Converting the HOW end time from DateTimeIndex to total amount of seconds
end = pd.to_datetime(e)
Time_series_e = pd.Series(end)
MicroSeconds_e = ((Time_series_e.dt.hour)*360 + (Time_series_e.dt.minute)*60 +
                  Time_series_e.dt.second)*1000000 + Time_series_e.dt.microsecond
Seconds_e = MicroSeconds_e/1000000

# Delete similar HOW start times
# If the difference between timestamps is less than 0.1 s assume timestamps as identical
Time = pd.DataFrame(Seconds)
Diff = Time.diff()
unique = (Diff[0] > 0.1)
indices = [i for i, x in enumerate(unique) if x]

# Omit similar timestamps and place in 'Warning' dataframe
for i in range(len(indices)):
    indices[i] = indices[i] - 1
indices = np.array(indices)
HOW_filtered = Seconds.loc[indices]
Warning = HOW_filtered.to_frame()
Warning = Warning.reset_index(drop=True)

# Converting timestamps of x and y coordinates from DateTimeIndex to total amount of seconds
# For HU data
hu = pd.DataFrame(i_hu)
hu[0] = pd.to_datetime(hu[0])
hu[0] = pd.Series(hu[0])
hu[0] = (((hu[0].dt.hour)*360 + (hu[0].dt.minute)*60 + hu[0].dt.second)*1000000
         + hu[0].dt.microsecond)/1000000

# For IC data
ic = pd.DataFrame(i_ic)
ic[0] = pd.to_datetime(ic[0])
ic[0] = pd.Series(ic[0])
ic[0] = (((ic[0].dt.hour)*360 + (ic[0].dt.minute)*60 + ic[0].dt.second)*1000000
         + ic[0].dt.microsecond)/1000000

# Median filter for x and y positions with window of 6
# Based off median filter of 100 ms with sampling frequency of 60 Hz
hu[3] = ndimage.median_filter(hu[1], size=6)
hu[4] = ndimage.median_filter(hu[2], size=6)
```

```
ic[3] = ndimage.median_filter(ic[1], size=6)
ic[4] = ndimage.median_filter(ic[2], size=6)

# Plotting raw data vs filtered data with median filter to examine effect of filter
fig, axs = plt.subplots(2, 1, constrained_layout=True)
axs[0].plot(hu[0], hu[1])
axs[0].plot(hu[0],hu[3])
axs[0].set_title('(filtered) x-coordinates')
axs[0].set_xlabel('Time [s]')
axs[0].set_ylabel('x-coordinates [pix]')
fig.suptitle('Raw X and Y coordinates filtered with a median filter', fontsize=16)

axs[1].plot(hu[0], hu[2])
axs[1].plot(hu[0],hu[4])
axs[1].set_title('(filtered) y-coordinates')
axs[1].set_xlabel('Time [sec]')
axs[1].set_ylabel('y-coordinates [pix]')

plt.show()


# Calculating velocity
# x_dot and y_dot and theta_dot calculations for HU
sample_t = 1/60
xdot_hu = hu[3].diff() / hu[0].diff()                              # [pix/sec]
ydot_hu = hu[4].diff() / hu[0].diff()                              # [pix/sec]
vel_hu = np.sqrt(np.square(xdot_hu) + np.square(ydot_hu))          # [pix/sec]

# Calculating velocity
# x_dot and y_dot and theta_dot calculations for IC
xdot_ic = ic[3].diff() / ic[0].diff()                              # [pix/sec]
ydot_ic = ic[4].diff() / ic[0].diff()                              # [pix/sec]
vel_ic = np.sqrt(np.square(xdot_ic) + np.square(ydot_ic))          # [pix/sec]

# Structuring velocities into dataframes
vel_hu = vel_hu.iloc[1:]
vel_hu = vel_hu.to_numpy()
vel_hu = pd.DataFrame(vel_hu)

vel_ic = vel_ic.iloc[1:]
vel_ic = vel_ic.to_numpy()
vel_ic = pd.DataFrame(vel_ic)

# Converting velocities into angular velocity for HU and IC
degpp_hu = degrees(atan2(.5*h_hu, d_hu)) / (.5*r_hu)               # [deg/pix]
angvel_hu = pd.DataFrame(fs*(vel_hu*degpp_hu))                     # [deg/sec]

degpp_ic = degrees(atan2(.5*h_ic, d_ic)) / (.5*r_ic)               # [deg/pix]
angvel_ic = pd.DataFrame(fs*(vel_ic*degpp_ic))                     # [deg/sec]

# Omit physically impossimble saccade
# Filter on saccade velocity threshold
# Omit angular velocity angvel > 1000 degrees/sec
sac_max_hu = (angvel_hu[0] > 1000)
i_sacmaxHU = [i for i, x in enumerate(sac_max_hu) if x]
```

```
# Setting the saccade velocity threshold at 2000 pixels/sec
sac_threshold_hu = (vel_hu[0] > 2000)
i_sacthresholdHU = [i for i, x in enumerate(sac_threshold_hu) if x]
i_filtered_hu = (np.unique(i_sacmaxHU + i_sacthresholdHU)).tolist()

# Omit physically impossimble saccade
# Filter on saccade velocity threshold
# Omit angular velocity angvel > 1000 degrees/sec
sac_max_ic = (angvel_ic[0] > 1000)
i_sacmaxIC = [i for i, x in enumerate(sac_max_ic) if x]
# Setting the saccade velocity threshold at 2000 pixels/sec
sac_threshold_ic = (vel_ic[0] > 2000)
i_sacthresholdIC = [i for i, x in enumerate(sac_threshold_ic) if x]

# The (unique) indices that need to be removed based on threshold values
i_filtered_hu = (np.unique(i_sacmaxHU + i_sacthresholdHU)).tolist()
i_filtered_ic = (np.unique(i_sacmaxIC + i_sacthresholdIC)).tolist()

#Grouping HU and IC data [Time,X,Y,X filtered, Y filtered, Velocity, Angular velocity]
hu_grouped = hu.iloc[1:(len(hu[:-1]))]
hu_grouped.columns = ['Time', 'X', 'Y','X filtered', 'Y filtered']
hu_grouped.loc[:, "Velocity"] = vel_hu
hu_grouped.loc[:, "Angular velocity"] = angvel_hu

ic_grouped = ic.iloc[1:(len(hu[:-1]))]
ic_grouped.columns = ['Time', 'X', 'Y', 'X filtered', 'Y filtered']
ic_grouped.loc[:, "Velocity"] = vel_ic
ic_grouped.loc[:, "Angular velocity"] = angvel_ic

#Splitting hu_grouped into separate fixations based on filtered saccades
fixations_hu = []
fixations_ic = []

#Add first sub-range
fixations_hu.append(hu_grouped[0:i_filtered_hu[0]-1])
fixations_ic.append(ic_grouped[0:i_filtered_ic[0]-1])

#Add middle pack of subsets
for i in range(0,len(i_filtered_hu[:-1])):
    start_hu = i_filtered_hu[i]
    end_hu = i_filtered_hu[i+1]-1
    fixations_hu.append(hu_grouped[start_hu:end_hu])

for i in range(0,len(i_filtered_ic[:-1])):
    start_ic = i_filtered_ic[i]
    end_ic = i_filtered_ic[i+1]-1
    fixations_ic.append(ic_grouped[start_ic:end_ic])

#Add last sub-range
fixations_hu.append(hu_grouped[i_filtered_hu[-1]:])
fixations_ic.append(ic_grouped[i_filtered_ic[-1]:])

# Remove empty fixations from list of dataframes
fixations_filtered_hu = [i for i in fixations_hu if not(i.empty)]
fixations_filtered_ic = [i for i in fixations_ic if not(i.empty)]
```

```
# Dataframe containing information about fixations
# [Fixation start, fixation duration, median X, median Y]
# Filter on minimum fixation duration of 50 ms than
# Filter on specific placement of HOW message
# HU: 1094 < x < 1994, 553.725 < y < 1216.225
# IC: 750 < x < 1650, 118.725 < y < 781.275
fixation_duration_hu = pd.DataFrame([(i.iloc[0,0],(i.iloc[-1,0]-
                                                   i.iloc[0,0]),
                                      st.median(i['X'].tolist()),
                                      st.median(i['Y'].tolist())) for i in
                                     fixations_filtered_hu if (i.iloc[-1,0]-
                                                               i.iloc[0,0])
                                     > 0.05 and st.median(i['X'].tolist())
                                     > 1094 and st.median(i['X'].tolist())
                                     < 1994 and st.median(i['Y'].tolist())
                                     > 553.725 and st.median(i['Y'].tolist())
                                     < 1216.225])
fixation_duration_ic = pd.DataFrame([(i.iloc[0,0],(i.iloc[-1,0]-
                                                   i.iloc[0,0]),
                                      st.median(i['X'].tolist()),
                                      st.median(i['Y'].tolist())) for i in
                                     fixations_filtered_ic if (i.iloc[-1,0]-
                                                               i.iloc[0,0])
                                     > 0.05 and st.median(i['X'].tolist())
                                     > 750 and st.median(i['X'].tolist())
                                     < 1650 and st.median(i['Y'].tolist())
                                     > 118.725 and st.median(i['Y'].tolist())
                                     < 781.275])

# Fixation starting times
fixation_ST_hu =fixation_duration_hu.iloc[:,0]

# Function to find nearest value
def find_nearest(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return array[idx]

# Function to find index of nearest value
def find_nearest_idx(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return idx

# --- Determining noticing times of HOW waring ---
# Empty structure for fixation times of HU
# Empty structure for fixation times indices of HU
# Converting HOW message start times to list
Fixation_time_hu = []
Fixation_time_hu_idx = []
Warning = Warning[0].tolist()

for i in range(0,len(Warning[:-1])):
    # Place the nearest idx of fixation start time value to the warning message in list
    Fixation_time_hu.append(find_nearest(fixation_ST_hu, Warning[i]))
    Fixation_time_hu_idx.append(find_nearest_idx(fixation_ST_hu, Warning[i]))
```

```
# Find the location of the selected nearest fixation out of fixation duration list
Fixation_duration_neg = []
for i in range(0,len(Fixation_time_hu_idx[:-1])):
    Fixation_duration_neg.append(fixation_duration_hu.iloc[Fixation_time_hu_idx[i],1])

# --- Calculating first iteration of noticing times of HOW message ---
# Empty structure for noticing times
NoticingTimes = []
zip_object = zip(Warning, Fixation_time_hu)
for Warning_i, Fixation_time_hu_i in zip_object:
    # Noticing times = Fixation start time - Warning message start time
    NoticingTimes.append((Fixation_time_hu_i - Warning_i))
NoticingTimes = pd.DataFrame(NoticingTimes)

# Determining noticing times of instances where participant was already looking
# at message
# If noticing times is negative AND fixation duration is longer than absolute
# noticing times
# Then noticing times is equal to zero

# --- HU ---
for i in range(0,len(NoticingTimes[:-1])):
    if (NoticingTimes.iloc[i,0] < 0 and abs(NoticingTimes.iloc[i,0])
        < Fixation_duration_neg[i-1]):
        NoticingTimes.iloc[i,0] = 0

# Select fixation start time nearest to warning message that only take place
# after the message
Fixation_time_upper_hu = []
for i in range(0,len(Warning[:-1])):
    # Only take into account fixations that take place after the HOW message
    fixation_ST_hu_fil = [x for x in fixation_ST_hu if x > Warning[i]]
    Fixation_time_upper_hu.append(find_nearest(fixation_ST_hu_fil, Warning[i]))

# Calculating second iteration of noticing times
# Adding upper bound noticing times
NoticingTimes_upper = []
zip_object = zip(Warning, Fixation_time_upper_hu)
for Warning_i, Fixation_time_upper_hu_i in zip_object:
    NoticingTimes_upper.append((Fixation_time_upper_hu_i - Warning_i))
NoticingTimes_upper = pd.DataFrame(NoticingTimes_upper)


# Calculating third iteration of noticing times
# If participant reacts/ends HOW message before fixation is recognized,
# then peripheral vision is assumed
EndTime_closest = []
for i in range(0,len(Warning[:-1])):
    # Only take into account end times of HOW message after HOW message
    Seconds_e_fil = [x for x in Seconds_e if x > Warning[i]]
    # Find nearest end time of HOW message
    EndTime_closest.append(find_nearest(Seconds_e_fil, Warning[i]))

# Append upperbound of noticing times if negative
for i in range(0,len(NoticingTimes)):
    if NoticingTimes.iloc[i,0] < 0:
```

```
        Fixation_time_hu[i] = Fixation_time_upper_hu[i]
        NoticingTimes.iloc[i,0] = NoticingTimes_upper.iloc[i,0]

# If noticing times is negative and the end time is later than the upper
# fixation times, then keep upper noticing times
for i in range(0,len(NoticingTimes[:-1])):
    if (NoticingTimes.iloc[i,0] < 0 and EndTime_closest[i]
        > NoticingTimes_upper.iloc[i,0]):
        NoticingTimes.iloc[i,0] = NoticingTimes_upper.iloc[i,0]

# For all noticing times, if end time of HOW message is before nearest fixation
# time than peripheral vision is assumed
# Then apply NAN to dataframe
for i in range(0,len(Warning[:-1])):
    if EndTime_closest[i] < Fixation_time_hu[i]:
        NoticingTimes.iloc[i,0] = float("NAN")

PerVision = pd.DataFrame(NoticingTimes.isna().sum())
# Export Dataframe to csv file
NoticingTimes.to_csv('NoticingTimes_16G.csv')
PerVision.to_csv('PerVision_16G.csv')
```

G. Python code: data analysis of static NTG7 interface

```
#Rhita Addi - Msc. Vehicle Engineering - 4367391


import csv
import pandas as pd
from pandas import Timestamp
import numpy as np
from datetime import timedelta
from datetime import time
import datetime as datetime
from datetime import datetime
import time
from scipy import ndimage, misc
import statistics as st
import matplotlib.pyplot as plt
from math import atan2, degrees


# --- Constant variables ---
# d_hu: Distance from chair to Head Unit [cm]
# h_hu: Height of HU display in [cm]
# r_hu: Height of HU display in [pix]
# d_ic: Distance from chair to Instrument Cluster [cm]
# h_ic: Height of IC display in [cm]
# r_ic: Height of IC display in [pix]
# fs: Sampling frequency in [Hz]
d_hu = 78.6
h_hu = 23.2
r_hu = 1728
d_ic = 71.5
h_ic = 11.0
r_ic = 900

fs = 60

#Importing CSV file
raw_csv = [i for i in csv.reader(open("datasets/P13_s.csv"),delimiter=';')]

# --- Creating empty data structures ---
# hu_x, hu_y: Structures for x and y coordinates of the Head Unit (HU)
# ic_x, ic_y: Structures for x and y coordinated of the Instrument Cluster (IC)
# start: Structure for the timestamp at which HOW message starts
# e: Structure for the timestamp at which HOW message ends
# grouped_hu, grouped_ic: Structure for grouping coordinates per screen (HU/IC)
# i_hu, i_ic: Structure for indices per screen (HU/IC)
hu_x, hu_y = [], []
ic_x, ic_y  = [], []
start = []
e = []
grouped_hu, grouped_ic  = [], []
i_hu, i_ic= [], []
EndTime = []
end = []
```

```
# Defining functions that read the data file and process it
# append() will place new items in the next available space
def process_hu_row(row):
    # HU-entries processing
    # Recognizinig GazeConnect.gaze... in the logging file
    # Adding timestamp and corresponding x and y values to hu_x/y
    # Adding no values to timestamp when no gaze is detected
    if row[0] == "GazeConnect.gazeX":
        hu_x.append((row[1],row[2]))
    if row[0] == "GazeConnect.gazeY":
        hu_y.append((row[1],row[2]))
    if row[0] == "GazeNotDetected":
        hu_x.append((row[1],None))
        hu_y.append((row[1],None))

def process_ic_row(row):
    # IC-entries processing
    # Recognizinig IC.gaze... in the logging file
    # Adding timestamp and corresponding x and y values to hu_x/y
    # Adding no values to timestamp when no gaze is detected
    if row[0] == "IC.gazeX":
        ic_x.append((row[1],row[2]))
    if row[0] == "IC.gazeY":
        ic_y.append((row[1],row[2]))
    if row[0] == "GazeNotDetected_IC":
        ic_x.append((row[1],None))
        ic_y.append((row[1],None))

def process_misc_entries(row):
    # Miscellaneous entries processing
    # Recognizing HOW_start in the logging file as HOW starting message
    # Adding corresponding timestamp to when HOW_start is found in file
    # Recognizing E in the logging file as HOW ending message
    # Adding corresponding timestamp to when E is found in file
    if row[0] == "HOW_end":
        end.append(row[1])
    if row[0] == "HOW_start":
        start.append(row[1])
    if row[0] == "E":
        e.append(row[1])
    if row[0] == "EndTime":
        EndTime.append(row[1])

def group_hu_entries():
    # Grouping x, y and timestamp
    prev_no_detection = False
    for i in range(len(hu_y)):
        t1, t2 = Timestamp(hu_x[i][0].replace(',','')),Timestamp(hu_y[i][0].replace(',',''))
        x, y = None, None
        if not(hu_x[i][1] is None):
            x= float(hu_x[i][1].replace(',', '.'))
        if not(hu_y[i][1] is None):
            y = float(hu_y[i][1].replace(',', '.'))

        # Averaging slightly different timestamps for x and y coordinates
        avg = Timestamp((t1.value + t2.value) / 2.0)
```

```
        if ((x is None) or (y is None)):
            if not (prev_no_detection):
                grouped_hu.append((avg, x, y))
                prev_no_detection = True
        else:
            grouped_hu.append((avg, x, y))
            prev_no_detection = False

def group_ic_entries():
    # Grouping x, y and timestamp
    prev_no_detection = False
    for i in range(len(ic_y)):
        t1, t2 = Timestamp(ic_x[i][0].replace(',','')),Timestamp(ic_y[i][0].replace(',',''))
        x, y = None, None
        if not(ic_x[i][1] is None):
            x= float(ic_x[i][1].replace(',', '.'))
        if not(ic_y[i][1] is None):
            y = float(ic_y[i][1].replace(',', '.'))

        # Averaging slightly different timestamps for x and y coordinates
        avg = Timestamp((t1.value + t2.value) / 2.0)

        if ((x is None) or (y is None)):
            if not(prev_no_detection):
                grouped_ic.append((avg, x, y))
                prev_no_detection = True
        else:
            grouped_ic.append((avg, x, y))
            prev_no_detection = False

def interpolate_hu():
    # Linear interpolation between missing x and y data points
    for i in range(1,len(grouped_hu)):
        x,y = grouped_hu[i][1], grouped_hu[i][2]
        if ((x is None) or (y is None)):
            if not(i == (len(grouped_hu)-1)):
                i_x = float((grouped_hu[i-1][1] + grouped_hu[i+1][1])/2.0)
                i_y = float((grouped_hu[i - 1][2] + grouped_hu[i + 1][2]) / 2.0)
                i_t = grouped_hu[i][0]
                i_hu.append((i_t,i_x,i_y))
        else:
            i_hu.append(grouped_hu[i])

def interpolate_ic():
    # Linear interpolation between missing x and y data points
    for i in range(1,len(grouped_ic)):
        x,y = grouped_ic[i][1], grouped_ic[i][2]
        if ((x is None) or (y is None)):
            if not(i == (len(grouped_ic)-1)):
                i_x = float((grouped_ic[i-1][1] + grouped_ic[i+1][1])/2.0)
                i_y = float((grouped_ic[i - 1][2] + grouped_ic[i + 1][2]) / 2.0)
                i_t = grouped_ic[i][0]
                i_ic.append((i_t,i_x,i_y))
        else:
            i_ic.append(grouped_ic[i])
```

```
def clean_data():
    # Grouping the functions together that should clean all the raw data files
    for row in raw_csv[1:]:
        process_hu_row(row)
        process_ic_row(row)
        process_misc_entries(row)
    group_hu_entries()
    group_ic_entries()
    interpolate_hu()
    interpolate_ic()

# Execute all previously defined functions
clean_data()

# Converting the HOW start time from DateTimeIndex to total amount of seconds
start = pd.DataFrame(EndTime)
start.columns = ["Time"]
start["Time"] = pd.to_datetime(start["Time"]) + pd.DateOffset(seconds=15)
Time_series = pd.Series(start["Time"])
MicroSeconds = ((Time_series.dt.hour)*360 + (Time_series.dt.minute)*60 +
                Time_series.dt.second)*1000000 + Time_series.dt.microsecond
Seconds = MicroSeconds/1000000

# Converting the HOW end time from DateTimeIndex to total amount of seconds
end = pd.to_datetime(e)
Time_series_e = pd.Series(end)
MicroSeconds_e = ((Time_series_e.dt.hour)*360 + (Time_series_e.dt.minute)*60 +
                  Time_series_e.dt.second)*1000000 + Time_series_e.dt.microsecond
Seconds_e = MicroSeconds_e/1000000

# Delete similar HOW start times
# If the difference between timestamps is less than 0.1 s assume
# timestamps as identical
Time = pd.DataFrame(Seconds)
Diff = Time.diff()
unique = (Diff["Time"] > 0.1)
indices = [i for i, x in enumerate(unique) if x]

# Omit similar timestamps and place in 'Warning' dataframe
for i in range(len(indices)):
    indices[i] = indices[i] - 1
indices = np.array(indices)
HOW_filtered = Seconds.iloc[indices]
Warning = HOW_filtered.to_frame()
Warning = Warning.reset_index(drop=True)

# Converting timestamps of x and y coordinates from DateTimeIndex to total
# amount of seconds
# For HU data
hu = pd.DataFrame(i_hu)
hu[0] = pd.to_datetime(hu[0])
hu[0] = pd.Series(hu[0])
hu[0] = (((hu[0].dt.hour)*360 + (hu[0].dt.minute)*60 + hu[0].dt.second)*1000000
        + hu[0].dt.microsecond)/1000000
```

```
# For IC data
ic = pd.DataFrame(i_ic)
ic[0] = pd.to_datetime(ic[0])
ic[0] = pd.Series(ic[0])
ic[0] = (((ic[0].dt.hour)*360 + (ic[0].dt.minute)*60 + ic[0].dt.second)*1000000
         + ic[0].dt.microsecond)/1000000

# Median filter for x and y positions with window of 6
# Based off median filter of 100 ms with sampling frequency of 60 Hz
hu[3] = ndimage.median_filter(hu[1], size=6)
hu[4] = ndimage.median_filter(hu[2], size=6)
ic[3] = ndimage.median_filter(ic[1], size=6)
ic[4] = ndimage.median_filter(ic[2], size=6)

# Plotting raw data vs filtered data with median filter to examine effect of filter
fig, axs = plt.subplots(2, 1, constrained_layout=True)
axs[0].plot(ic[0], ic[1])
axs[0].plot(ic[0],ic[3])
axs[0].set_title('(filtered) x-coordinates')
axs[0].set_xlabel('Time [s]')
axs[0].set_ylabel('x-coordinates [pix]')
fig.suptitle('Raw X and Y coordinates filtered with a median filter', fontsize=16)

axs[1].plot(ic[0], ic[2])
axs[1].plot(ic[0],ic[4])
axs[1].set_title('(filtered) y-coordinates')
axs[1].set_xlabel('Time [sec]')
axs[1].set_ylabel('y-coordinates [pix]')

plt.show()


# Calculating velocity
# x_dot and y_dot and theta_dot calculations for HU
sample_t = 1/60
xdot_hu = hu[3].diff() / hu[0].diff()                          # [pix/sec]
ydot_hu = hu[4].diff() / hu[0].diff()                          # [pix/sec]
vel_hu = np.sqrt(np.square(xdot_hu) + np.square(ydot_hu))      # [pix/sec]

# Calculating velocity
# x_dot and y_dot and theta_dot calculations for IC
xdot_ic = ic[3].diff() / ic[0].diff()                          # [pix/sec]
ydot_ic = ic[4].diff() / ic[0].diff()                          # [pix/sec]
vel_ic = np.sqrt(np.square(xdot_ic) + np.square(ydot_ic))      # [pix/sec]

# Structuring velocities into dataframes
vel_hu = vel_hu.iloc[1:]
vel_hu = vel_hu.to_numpy()
vel_hu = pd.DataFrame(vel_hu)

vel_ic = vel_ic.iloc[1:]
vel_ic = vel_ic.to_numpy()
vel_ic = pd.DataFrame(vel_ic)

# Converting velocities into angular velocity for HU and IC
degpp_hu = degrees(atan2(.5*h_hu, d_hu)) / (.5*r_hu)           # [deg/pix]
```

```
angvel_hu = pd.DataFrame(fs*(vel_hu*degpp_hu))                # [deg/sec]

degpp_ic = degrees(atan2(.5*h_ic, d_ic)) / (.5*r_ic)          # [deg/pix]
angvel_ic = pd.DataFrame(fs*(vel_ic*degpp_ic))                # [deg/sec]

# Omit physically impossimble saccade
# Filter on saccade velocity threshold
# Omit angular velocity angvel > 1000 degrees/sec
sac_max_hu = (angvel_hu[0] > 1000)
i_sacmaxHU = [i for i, x in enumerate(sac_max_hu) if x]
# Setting the saccade velocity threshold at 2000 pixels/sec
sac_threshold_hu = (vel_hu[0] > 2000)
i_sacthresholdHU = [i for i, x in enumerate(sac_threshold_hu) if x]
i_filtered_hu = (np.unique(i_sacmaxHU + i_sacthresholdHU)).tolist()

# Omit physically impossimble saccade
# Filter on saccade velocity threshold
# Omit angular velocity angvel > 1000 degrees/sec
sac_max_ic = (angvel_ic[0] > 1000)
i_sacmaxIC = [i for i, x in enumerate(sac_max_ic) if x]
# Setting the saccade velocity threshold at 2000 pixels/sec
sac_threshold_ic = (vel_ic[0] > 2000)
i_sacthresholdIC = [i for i, x in enumerate(sac_threshold_ic) if x]


# The (unique) indices that need to be removed based on threshold values
i_filtered_hu = (np.unique(i_sacmaxHU + i_sacthresholdHU)).tolist()
i_filtered_ic = (np.unique(i_sacmaxIC + i_sacthresholdIC)).tolist()
#Grouping HU and IC data [Time,X,Y,X filtered, Y filtered, Velocity, Angular velocity]
hu_grouped = hu.iloc[1:(len(hu)-1)]
hu_grouped.columns = ['Time', 'X', 'Y','X filtered', 'Y filtered']
hu_grouped.loc[:, "Velocity"] = vel_hu
hu_grouped.loc[:, "Angular velocity"] = angvel_hu

ic_grouped = ic.iloc[1:(len(hu)-1)]
ic_grouped.columns = ['Time', 'X', 'Y', 'X filtered', 'Y filtered']
ic_grouped.loc[:, "Velocity"] = vel_ic
ic_grouped.loc[:, "Angular velocity"] = angvel_ic

#Splitting hu_grouped into separate fixations based on filtered saccades
fixations_hu = []
fixations_ic = []

#Add first sub-range
fixations_hu.append(hu_grouped[0:i_filtered_hu[0]-1])
fixations_ic.append(ic_grouped[0:i_filtered_ic[0]-1])

#Add middle pack of subsets
for i in range(0,len(i_filtered_hu[:-1])):
    start_hu = i_filtered_hu[i]
    end_hu = i_filtered_hu[i+1]-1
    fixations_hu.append(hu_grouped[start_hu:end_hu])

for i in range(0,len(i_filtered_ic[:-1])):
    start_ic = i_filtered_ic[i]
    end_ic = i_filtered_ic[i+1]-1
```

```
    fixations_ic.append(ic_grouped[start_ic:end_ic])

#Add last sub-range
fixations_hu.append(hu_grouped[i_filtered_hu[-1]:])
fixations_ic.append(ic_grouped[i_filtered_ic[-1]:])

# Remove empty fixations from list of dataframes
fixations_filtered_hu = [i for i in fixations_hu if not(i.empty)]
fixations_filtered_ic = [i for i in fixations_ic if not(i.empty)]

# Dataframe containing information about fixations
# [Fixation start, fixation duration, median X, median Y]
# Filter on minimum fixation duration of 50 ms than
# Filter on specific placement of HOW message
# HU: 1094 < x < 1994, 553.725 < y < 1216.225
# IC: 750 < x < 1650, 118.725 < y < 781.275
fixation_duration_hu = pd.DataFrame([(i.iloc[0,0],(i.iloc[-1,0]-i.iloc[0,0]),
                                      st.median(i['X'].tolist()),
                                      st.median(i['Y'].tolist())) for i in
                                     fixations_filtered_hu if (i.iloc[-1,0]-
                                                               i.iloc[0,0])
                                     > 0.05 and st.median(i['X'].tolist())
                                     > 1094 and st.median(i['X'].tolist())
                                     < 1994 and st.median(i['Y'].tolist())
                                     > 553.725 and st.median(i['Y'].tolist())
                                     < 1216.225])
fixation_duration_ic = pd.DataFrame([(i.iloc[0,0],(i.iloc[-1,0]-i.iloc[0,0]),
                                      st.median(i['X'].tolist()),
                                      st.median(i['Y'].tolist())) for i in
                                     fixations_filtered_ic if (i.iloc[-1,0]-
                                                               i.iloc[0,0])
                                     > 0.05 and st.median(i['X'].tolist())
                                     > 750 and st.median(i['X'].tolist())
                                     < 1650 and st.median(i['Y'].tolist())
                                     > 118.725 and st.median(i['Y'].tolist())
                                     < 781.275])

# Fixation starting times
fixation_ST_hu =fixation_duration_hu.iloc[:,0]
fixation_ST_ic =fixation_duration_ic.iloc[:,0]

# Function to find nearest value
def find_nearest(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return array[idx]

# Function to find index of nearest value
def find_nearest_idx(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return idx


# --- Determining noticing times of HOW waring ---
# Empty structure for fixation times of HU
```

```
# Empty structure for fixation times indices of HU
# Converting HOW message start times to list
Fixation_time_hu = []
Fixation_time_hu_idx = []
Fixation_time_ic = []
Fixation_time_ic_idx = []
Warning = Warning["Time"].tolist()


for i in range(0,len(Warning[:-1])):
# Place the nearest fixation start time value to the warning message in list
    Fixation_time_hu.append(find_nearest(fixation_ST_hu, Warning[i]))
# Place the nearest idx of fixation start time value to the warning message in list
    Fixation_time_hu_idx.append(find_nearest_idx(fixation_ST_hu, Warning[i]))
# Place the nearest fixation start time value to the warning message in list
    Fixation_time_ic.append(find_nearest(fixation_ST_ic, Warning[i]))
# Place the nearest idx of fixation start time value to the warning message in list
    Fixation_time_ic_idx.append(find_nearest_idx(fixation_ST_ic, Warning[i]))

# Find the location of the selected nearest fixation out of fixation duration list IC
Fixation_duration_neg_ic = []
for i in range(0,len(Fixation_time_ic_idx)):
    Fixation_duration_neg_ic.append(fixation_duration_ic.
                                    iloc[Fixation_time_ic_idx[i],1])

# Empty structure for noticing times
NoticingTimes_ic = []
zip_object = zip(Warning, Fixation_time_ic)
for Warning_i, Fixation_time_ic_i in zip_object:
    # Noticing times = Fixation start time - Warning message start time
    NoticingTimes_ic.append((Fixation_time_ic_i - Warning_i))
NoticingTimes_ic = pd.DataFrame(NoticingTimes_ic)

# Determining noticing times of instances where participant was already
# looking at message
# If noticing times is negative AND fixation duration is longer than
# absolute noticing times
# Then noticing times is equal to zero
# IC
for i in range(0,len(NoticingTimes_ic[:-1])):
    if (NoticingTimes_ic.iloc[i,0] < 0 and abs(NoticingTimes_ic.iloc[i,0])
        < Fixation_duration_neg_ic[i]):
        NoticingTimes_ic.iloc[i,0] = 0

# Select fixation start time nearest to warning message that only take
# place after the message
Fixation_time_upper_ic = []
for i in range(0,len(Warning[:-1])):
    # Only take into account fixations that take place after the HOW message
    fixation_ST_ic_fil = [x for x in fixation_ST_ic if x > Warning[i]]
    if len(fixation_ST_ic_fil) != 0:
        Fixation_time_upper_ic.append(find_nearest(fixation_ST_ic_fil, Warning[i]))
    elif len(fixation_ST_ic_fil) == 0:
        Fixation_time_upper_ic.append(float("NAN"))

# Calculating second iteration of noticing times
```

```
# Adding upper bound noticing times
NoticingTimes_upper_ic = []
zip_object = zip(Warning, Fixation_time_upper_ic)
for Warning_i, Fixation_time_upper_ic_i in zip_object:
    NoticingTimes_upper_ic.append((Fixation_time_upper_ic_i - Warning_i))
NoticingTimes_upper_ic = pd.DataFrame(NoticingTimes_upper_ic)


# Calculating third iteration of noticing times
# If participant reacts/ends HOW message before fixation is recognized,
# then peripheral vision is assumed
EndTime_closest_ic = []
for i in range(0,len(Warning[:-1])):
    # Only take into account end times of HOW message after HOW message
    Seconds_e_fil_ic = [x for x in Seconds_e if x > Warning[i]]
    # Find nearest end time of HOW message
    EndTime_closest_ic.append(find_nearest(Seconds_e_fil_ic, Warning[i]))
# Append upperbound of noticing times if negative
for i in range(0,len(NoticingTimes_ic)):
    if NoticingTimes_ic.iloc[i,0] < 0:
        Fixation_time_ic[i] = Fixation_time_upper_ic[i]
        NoticingTimes_ic.iloc[i,0] = NoticingTimes_upper_ic.iloc[i,0]

# If noticing times is negative and the end time is later than the upper
# fixation times, then keep upper noticing times
for i in range(0,len(NoticingTimes_ic[:-1])):
    if (NoticingTimes_ic.iloc[i,0] < 0 and EndTime_closest_ic[i]
        > NoticingTimes_upper_ic.iloc[i,0]):
        NoticingTimes_ic.iloc[i,0] = NoticingTimes_upper_ic.iloc[i,0]

# For all noticing times, if end time of HOW message is before nearest
# fixation time than peripheral vision is assumed
# Then apply NAN to dataframe
for i in range(0,len(Warning[:-1])):
    if EndTime_closest_ic[i] < Fixation_time_ic[i]:
        NoticingTimes_ic.iloc[i,0] = float("NAN")

# The amount of times stimulus is seen through peripheral vision
PerVision = pd.DataFrame(NoticingTimes_ic.isna().sum())

# Export Dataframes to csv file
NoticingTimes_ic.to_csv('NoticingTimes_13S.csv')
PerVision.to_csv('PerVision_13S.csv')
```

Consent Form

PRINCIPAL INVESTIGATORS: RHITA ADDI
Mercedes Benz AG

Purpose of the evaluation: In this study we aim to evaluate the effectiveness of adaptive interfaces.

Procedure:

1. You are asked to sign the consent form.
2. Interviewer explains the procedure.
3. You will interact with the interface in the seating buck.
4. Upon your consent, eye-tracking data will be recorded.
5. You start the study following the instructions given by the investigator.
6. You will complete several non-driving related tasks on the interface.

Benefits: The results of this study will be useful for us to evaluate the possibilities of implementing eye-trackers in UI solutions.

Alternatives to Participation: Participation in this evaluation is voluntary. You are free to withdraw or discontinue participation.

Cost and Compensation: Participation in this evaluation will involve no cost to you.

Confidentiality: All information collected during the session will be kept strictly confidential. You will be identified through identification numbers. No publications or reports from this project will include identifying information on any participant. If you agree to join this study, please sign your name below.

_____ Male Female Other
Participant's Name Participant's Gender

_____ _____ _____
Participant's Age Participant's Signature Date

_____
Principal Investigator

Fig. 19: Consent form the participants signed to agree with the participation in the experiment.

Declaration of Consent for Recordings and Publication of Images and/or Audio/Video of Legal Adults

Person recorded:

Last name, first name:

The **subject matter** of this declaration covers photo images made by Daimler AG or its service partners.

I consent to allow photos and videos that were made of me or that I provided to be used by Daimler AG and its affiliates for the following purpose without compensation and for an indefinite period: **Employer Branding**

For distribution and public exhibition

On the intranet[1] / on the Internet (including social media)[2]/ in internal/external, digital media (offline, e.g. presentations) / for sharing with the press and other media outlets,

in order to **communicate about Daimler as an employer.**

In the event that the recordings are used for other purposes, I will be asked to give separate consent.

To the extent that my photo suggests my ethnic background, religion or health (e.g. skin tone, headdress, glasses), my consent also pertains to this information.

☐ My name may be specified in connection with the recordings.

☐ My name shall not be specified in connection with the recordings.

The legal basis for processing of personal data is your consent pursuant to Article 6 (1f) of the General Data Protection Regulation (GDPR). This consent is voluntary. This declaration is governed exclusively by German law.

The **controller** for the purpose of data processing is Daimler AG, which is obligated to comply with the requirements of the GDPR. The controller can be reached at Mercedesstrasse 120, 70327 Stuttgart, e-mail: dialog@daimler.com".

The controller may transmit the recordings to service providers that assist it with the creation and distribution of the recordings, e.g. media companies or IT service providers.

You may contact tristan.fluechter@daimler.com to request the personal information stored about you. Under certain conditions, you can also request the rectification or erasure of your data. You may also be entitled to a right to restrict the processing of your personal data as well as a right to have the personal data provided by you disclosed in a structured, commonly used and machine-readable format.

You have the right to lodge a complaint with the data protection officer or a data protection supervisory authority if you feel that the processing of your personal information violates the GDPR or other laws (Art. 77 of the GDPR). You can reach our **data protection officer** at: Chief Officer for Corporate Data Protection, Daimler AG, 70546 Stuttgart, dataprotection@daimler.com.

The company limits the **storage** of your data to the necessary period. For this reason, we regularly delete your personal information as follows: 6 years after publication.

_____ _____

Place, date **Signature** of the person recorded

I agree to having my data processed as described above.

I have been informed that I may **revoke** my consent at any time. The revocation of your consent does not affect the legality of the data processing that was carried out based on your consent prior to its revocation. Send your revocation to: Tristan.fluechter@daimler.com.

[1]The **data recipients** are Daimler Group companies and other people associated with Daimler's intranet. The recording will thus also be **transmitted outside the EEA.** This transmission takes place based on the appropriate data protection guarantees, specifically the Daimler Data Protection Policy: https://www.daimler.com/datenschutz/.

[2] I acknowledge that information **on the Internet is accessible around the world** and that further use of these photos by third parties cannot be ruled out as a result. Recordings can be found using search engines and may be linked to other information. They can be copied and further distributed. This may mean that even after being erased on the original page, information published on the internet can still be found elsewhere.

Fig. 20: Consent form the participants signed to agree with any recordings during the experiment.