



Technische Universität München  
Lehrstuhl für Ergonomie

## Modeling Driver Distraction

Michael Christian Florian Krause

Vollständiger Abdruck der von der Fakultät für Maschinenwesen  
der Technischen Universität München  
zur Erlangung des akademischen Grades eines  
Doktor-Ingenieurs (Dr.-Ing.) genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Manfred Hajek  
Prüfer der Dissertation: 1. Prof. Dr. Klaus Bengler  
2. Prof. Dr. Martin Baumann

Die Dissertation wurde am 09.01.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 20.06.2017 angenommen.



## Abstract

The assessment and empirical testing of the potential for interfaces to distract drivers is a time-consuming and costly issue in the automobile industry. This topic is addressed and supported by different guidelines and standards. For human factors engineering, it would be beneficial to obtain an approximate idea concerning the performance of a task in driver distraction testing before undertaking the experiments. This could improve suitable interaction design at an early stage e.g., during (paper) prototyping.

In this thesis, a prediction model is implemented (open source) and evaluated.

The approach is based on measuring subtasks and storing their results in a database. From the subtask database, complete tasks can be assembled. The subtasks were measured from 24 subjects. A separate prediction is calculated for each subject based on synthesized subtasks (virtual experiment). From these 24 values (distribution), characteristic values such as the 85<sup>th</sup> percentile can be derived.

After discussing the properties of delays, System Response Times are incorporated into the prediction model and are used in an evaluation experiment to test the model. It is demonstrated that System Response Times can have an impact on distraction metrics. These delays can (mathematically) lower Single Glance Durations.

Typical driver distraction metrics are reviewed and enhanced (e.g., for lateral driving performance and Single Glance Durations). The prediction model incorporates 13 metrics:

- Total Time on Task (TTT static; non-driving)
- Total Time on Task while driving
- Glance – Total Glance Time (task related)
- Glance – Single Glance Duration (task related)
- Glance – Number of Glances (task related)
- Glance – Total Eyes-Off-Road Time
- Glance – Single Glance Duration (eyes-off-road)
- Glance – Number of Glances (eyes-off-road)
- Occlusion – Total Shutter Open Time (TSOT)
- Occlusion – R-Metric (TSOT/TTT)
- Tactile Detection Response Task (TDRT) – Deterioration in Reaction Time (%)
- Driving – Deterioration in Lateral Drift (%)
- Driving – Deterioration in Longitudinal Drift of Headway (%)

An evaluation experiment with 24 subjects revealed that most of these predictions could be a helpful support. When excluding the unreliably predictable Deterioration in Longitudinal Drift of Headway, the average percentage error of predictions to measurements was 16%, with a mean coefficient of determination  $R^2 = .614$ .

## Zusammenfassung

Um das Fahrerablenkungspotential von Interfaces zu erfassen, werden in der Automobilindustrie (zeit- und kostenintensive) empirische Tests durchgeführt. Diese Vorgänge werden empfohlen und unterstützt durch regionale Richtlinien und internationale Standards. Für Ergonomen wäre es vorteilhaft bereits in einem frühen Stadium, zum Beispiel während der Konzeptfindung, eine grobe Vorstellung von möglichen späteren Testergebnissen zu erhalten.

In der Arbeit wird ein (quelloffenes) Prädiktionsmodell erstellt und evaluiert. Der Ansatz nutzt vermessene und gespeicherte Subtasks aus denen zur Prädiktion Aufgabenabläufe zusammengestellt werden können. Die abgespeicherten Subtasks stammen von 24 Probanden, für die jeweils durch das Zusammensetzen eine Vorhersage erstellt wird (virtuelles Experiment). Aus der Verteilung der 24 Werte können dann Kennzahlen wie das 85. Perzentil abgeleitet werden. Für die Umsetzung wurden verbreitete Metriken näher betrachtet und teilweise erweitert oder verbessert; beispielsweise betreffend die laterale Fahrzeugführung und die Einzelblickdauern.

Nach der Diskussion und Klassifikation von Verzögerungen werden Systemantwortzeiten in das Modell einbezogen und in einem Evaluationsexperiment eingesetzt. Die Ergebnisse zeigen, dass Systemantwortzeiten Einzelblickdauern (rechnerisch) reduzieren können.

Das Modell umfasst 13 Metriken:

- Total Time on Task (TTT static; non-driving)
- Total Time on Task while driving
- Glance – Total Glance Time (task related)
- Glance – Single Glance Duration (task related)
- Glance – Number of Glances (task related)
- Glance – Total Eyes-Off-Road Time
- Glance – Single Glance Duration (eyes-off-road)
- Glance – Number of Glances (eyes-off-road)
- Occlusion – Total Shutter Open Time (TSOT)
- Occlusion – R-Metric (TSOT/TTT)
- Tactile Detection Response Task (TDRT) – Deterioration in Reaction Time (%)
- Driving – Deterioration in Lateral Drift (%)
- Driving – Deterioration in Longitudinal Drift of Headway (%)

Die Ergebnisse eines Evaluationsexperiments mit 24 Probanden lassen darauf schließen, dass das Modell bei der Abschätzung und Vorbereitung von Fahrerablenkungstests hilfreich sein kann.

Nach Ausschluss der unzuverlässig prädizierbaren Deterioration in Longitudinal Drift of Headway, liegt der mittlere prozentuale Fehler der Prädiktionen bei 16%, mit einem durchschnittlichen Determinationskoeffizienten von  $R^2 = .614$ .

# Contents

<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>ix</b>
<b>Glossary</b>	<b>xi</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Fundamentals</b>	<b>6</b>
2.1. Driver Distraction Guidelines . . . . .	7
2.2. Delays in System Response . . . . .	9
2.3. Driver Performance Metrics . . . . .	20
2.4. Task Analysis and Modeling . . . . .	28
2.5. Own Previous Work and Motivation . . . . .	38
<b>3. Building the Model</b>	<b>41</b>
3.1. Hardware Setup . . . . .	42
3.2. Application and Subtasks . . . . .	46
3.3. Test Subjects and Procedure . . . . .	53
3.4. Postprocessing and Problems . . . . .	55
3.5. Prediction Model – Calculation Methods . . . . .	59
3.6. Descriptive Results . . . . .	61
3.6.1. Comparison to Former Experiment (Age) . . . . .	61
3.6.2. Glance Metrics With and Without TDRT Measurement . . . . .	64
3.6.3. Glance Metrics During Delays . . . . .	65
<b>4. Evaluation Experiment</b>	<b>71</b>
4.1. Hardware Setup . . . . .	72
4.2. Tasks . . . . .	74
4.2.1. Task 1, Touchscreen – ‘Config’ . . . . .	75
4.2.2. Task 2, Touchscreen – ‘Radio Tuning’ . . . . .	77
4.2.3. Task 3, Touchscreen – ‘Phone Normal’ . . . . .	78
4.2.4. Task 4, Touchscreen – ‘Phone Delay’ . . . . .	78
4.2.5. Task 5, Touchscreen – ‘Phone Blanking’ . . . . .	79
4.2.6. Task 6, Touchscreen – ‘Spell’ . . . . .	81
4.2.7. Task 7, Rotary Knob – ‘Contacts’ . . . . .	82
4.2.8. Task 8, Rotary Knob – ‘Spell’ . . . . .	83
4.2.9. Task 9, Rotary Knob – ‘Phone’ . . . . .	84
4.2.10. Task 10, Rotary Knob – ‘Config’ . . . . .	85
4.2.11. Acclimatization Tasks . . . . .	85
4.3. Test Subjects and Procedure . . . . .	86

4.4.	Hypotheses and Questions . . . . .	89
4.5.	Postprocessing and Problems . . . . .	91
4.6.	Results and Discussion . . . . .	95
4.6.1.	Pass/Fail Overview . . . . .	95
4.6.2.	Issue 1 – Predictive Quality of the Model . . . . .	97
4.6.3.	Hypotheses 2 – Effects on Single Glance Durations . . . . .	101
4.6.4.	Issue 3 – Metrics With and Without TDRT . . . . .	103
4.6.5.	Hypotheses 4 – Age Effects . . . . .	106
4.6.6.	Issue 5 – Training/Accommodation Effects . . . . .	108
<b>5.</b>	<b>Conclusion</b>	<b>111</b>
	<b>Bibliography</b>	<b>115</b>
<b>A.</b>	<b>Appendix – Prediction Tool Manual</b>	<b>124</b>
<b>B.</b>	<b>Appendix – Instructions</b>	<b>131</b>
<b>C.</b>	<b>Appendix – App Parameters</b>	<b>134</b>
<b>D.</b>	<b>Appendix – Evaluation Results – Extended Data</b>	<b>142</b>
D.1.	Total Task Time Unoccluded . . . . .	143
D.2.	Total Shutter Open Time . . . . .	144
D.3.	Occlusion R-ratio . . . . .	145
D.4.	Total Task Time While Driving . . . . .	146
D.5.	Total Glance Time to IVIS . . . . .	147
D.6.	Number of Glances to IVIS . . . . .	148
D.7.	Single Glance Duration to IVIS . . . . .	149
D.8.	Total Eyes-Off-Road Time . . . . .	150
D.9.	Number of Glances, Eyes-Off-Road . . . . .	151
D.10.	Single Glance Duration, Eyes-Off-Road . . . . .	152
D.11.	DRT Deterioration . . . . .	153
D.12.	DLP Deterioration . . . . .	154
D.13.	DFH Deterioration . . . . .	155
D.14.	85 <sup>th</sup> Percentile Predictions and Bootstrapped Results . . . . .	156
D.14.1.	Total Shutter Open Time – 85 <sup>th</sup> Percentile . . . . .	156
D.14.2.	Total Glance Time – 85 <sup>th</sup> Percentile . . . . .	158
D.14.3.	Single Glance Duration to IVIS – 85 <sup>th</sup> Percentile . . . . .	159
D.14.4.	Total Eyes-Off-Road Time – 85 <sup>th</sup> Percentile . . . . .	160
D.14.5.	Single Glance Duration, Eyes-Off-Road – 85 <sup>th</sup> Percentile . . . . .	161
D.14.6.	DLP and DFH Bootstrap Indicator . . . . .	162

# List of Figures

2.1.	Delay levels (cf. 2008/653/EC, 2008, Principle 4.3.4.7.); illustration (cf. Kaaresoja and Brewster, 2010, Figure 2 and Figure 3) . . . . .	10
2.2.	SDLP values calculated with different data lengths (unfiltered, high-pass filtered with 0.1 Hz and 0.5 Hz) from Östlund et al. (2005, p. 39; Figure 7)	21
2.3.	Lane position trajectories . . . . .	22
2.4.	Spectral densities of lane positions . . . . .	23
2.5.	Spectral densities of following headway . . . . .	24
2.6.	Lateral (Lane Position; LP) and longitudinal (Following Headway; FH) metrics between ego-car and leading vehicle . . . . .	25
2.7.	Distribution of mean occlusion task times of two age groups from Kang et al. (2013, p. 20) . . . . .	31
2.8.	Task estimate dialog box (1) from Kurokawa (1990, p. 284; Figure 89) . . .	33
2.9.	Task analytic procedure dialog box (1) from Kurokawa (1990, p. 298; Figure 93) . . . . .	34
2.10.	Task analytic procedure dialog box (2) from Kurokawa (1990, p. 299; Figure 94) . . . . .	34
2.11.	Split glance problem (cf. Krause et al., 2015b) . . . . .	39
3.1.	Laboratory setup subtask experiment . . . . .	42
3.2.	Laboratory setup subtask experiment . . . . .	43
3.3.	Network Connections . . . . .	44
3.4.	App config/start screen . . . . .	46
3.5.	Example of a workflow for one subtask block . . . . .	46
3.6.	Subtask – Delay visualizations . . . . .	48
3.7.	Subtask – Number input . . . . .	49
3.8.	Subtask – List selection . . . . .	49
3.9.	Subtask – +/- Number input . . . . .	50
3.10.	Subtask – Slider . . . . .	51
3.11.	Subtask – Text input . . . . .	51
3.12.	Mileage . . . . .	53
3.13.	Areas of Interest in D-Lab . . . . .	55
3.14.	Histogram of TDRT reaction times . . . . .	57
3.15.	Availability of test subjects for a subtask TDRT metric . . . . .	60
3.16.	Glance metrics during delays . . . . .	69
4.1.	Laboratory setup for the evaluation experiment (panorama) . . . . .	72
4.2.	Laboratory setup for the evaluation experiment . . . . .	72
4.3.	Task flow – Task 1, Touchscreen – ‘Config’ . . . . .	75
4.4.	Task flow – Task 2, Touchscreen – ‘Radio Tuning’ . . . . .	77
4.5.	Task flow – Task 3, Touchscreen – ‘Phone Normal’ . . . . .	78
4.6.	Task flow – Task 4, Touchscreen – ‘Phone Delay’ . . . . .	78

4.7. Task flow – Task 5, Touchscreen – ‘Phone Blanking’ . . . . .	79
4.8. Display Blanking Algorithm . . . . .	80
4.9. Task flow – Task 6 Touchscreen – ‘Spell’ . . . . .	81
4.10. Task flow – Task 7 Rotary Knob – ‘Contacts’ . . . . .	82
4.11. Task flow – Task 8 Rotary Knob – ‘Spell’ . . . . .	83
4.12. Task flow – Task 9 Rotary Knob – ‘Phone’ . . . . .	84
4.13. Task flow – Task 10 Rotary Knob – ‘Config’ . . . . .	85
4.14. Mileage . . . . .	86
4.15. Experimental Procedure . . . . .	87
4.16. Areas of Interest in D-Lab . . . . .	91
4.17. Calculation of TSOT from $TTT_{occluded}$ . . . . .	92
4.18. Histogram of TDRT reaction times . . . . .	93
4.19. Subjective ratings for the interactions with the phone tasks . . . . .	101
4.20. Mean Total Glance Time – With/without TDRT method . . . . .	103
4.21. Mean Single Glance Duration – With/without TDRT method . . . . .	104
4.22. Mean Drift in Lane Position – With/without TDRT method . . . . .	104
4.23. Radio Tuning, Point in Time (early, late) . . . . .	108
A.1. Online tool . . . . .	124
A.2. Config window . . . . .	125
A.3. Composed task window . . . . .	125
A.4. Add subtask. Subtask selection window . . . . .	126
A.5. Change subtask . . . . .	126
A.6. Subtask description . . . . .	127
A.7. Glance visualization . . . . .	127
A.8. Subtask distribution . . . . .	128
A.9. Result visualization . . . . .	129
B.1. Instructions – General . . . . .	131
B.2. Instructions – Driving Task (I) . . . . .	132
B.3. Instructions – Driving Task (II) . . . . .	132
B.4. Instructions – Occlusion . . . . .	133
B.5. Instructions – Detection Response Task . . . . .	133
D.1. Evaluation results – Boxplot – Total Task Time unoccluded . . . . .	143
D.2. Evaluation results – Boxplot – Total Shutter Open Time . . . . .	144
D.3. Evaluation results – Boxplot – Occlusion R-ratio . . . . .	145
D.4. Evaluation results – Boxplot – Total Task Time while driving . . . . .	146
D.5. Evaluation results – Boxplot – Total Glance Time to IVIS . . . . .	147
D.6. Evaluation results – Boxplot – Number of Glances to IVIS . . . . .	148
D.7. Evaluation results – Boxplot – Single Glance Duration to IVIS . . . . .	149
D.8. Evaluation results – Boxplot – Total Eyes-Off-Road Time . . . . .	150
D.9. Evaluation results – Boxplot – Number of Glances, eyes-off-road . . . . .	151
D.10. Evaluation results – Boxplot – Single Glance Duration, eyes-off-road . . . . .	152
D.11. Evaluation results – Boxplot – DRT deterioration . . . . .	153
D.12. Evaluation results – Boxplot – DLP deterioration . . . . .	154
D.13. Evaluation results – Boxplot – DFH deterioration . . . . .	155



# List of Tables

1.1.	Timescales for driving safety impairments . . . . .	2
2.1.	Criteria of guidelines . . . . .	7
2.2.	Acceptable System Response Times from MIL-STD-1472G (2012, p. 24) . .	14
3.1.	Comparison of subtasks from this experiment to former experiments. Total Glance Time . . . . .	62
3.2.	Comparison of subtasks from this experiment to former experiments. Single Glance Duration . . . . .	62
3.3.	Comparison of subtasks from this experiment to former experiments. Total Shutter Open Time . . . . .	62
4.1.	Criteria – Measurement Pass/Fail Overview . . . . .	95
4.2.	Evaluation overview . . . . .	97
D.1.	Evaluation results Total Task Time unoccluded . . . . .	143
D.2.	Evaluation results Total Shutter Open Time . . . . .	144
D.3.	Evaluation results Occlusion R-ratio . . . . .	145
D.4.	Evaluation results Total Task Time while driving . . . . .	146
D.5.	Evaluation results Total Glance Time to IVIS . . . . .	147
D.6.	Evaluation results Number of Glances to IVIS . . . . .	148
D.7.	Evaluation results Single Glance Duration to IVIS . . . . .	149
D.8.	Evaluation results Total Eyes-Off-Road Time . . . . .	150
D.9.	Evaluation results Number of Glances, eyes-off-road . . . . .	151
D.10.	Evaluation results Single Glance Duration, eyes-off-road . . . . .	152
D.11.	Evaluation results DRT deterioration . . . . .	153
D.12.	Evaluation results DLP deterioration . . . . .	154
D.13.	Evaluation results DFH deterioration . . . . .	155
D.14.	Evaluation results P85 TSOT . . . . .	156
D.15.	Evaluation results Bootstrapping TSOT AAM 15 s limit . . . . .	157
D.16.	Evaluation results Bootstrapping TSOT NHTSA 12 s limit . . . . .	157
D.17.	Evaluation results P85 Total Glance Time . . . . .	158
D.18.	Evaluation results Bootstrapping TGT AAM 20 s limit . . . . .	158
D.19.	Evaluation results P85 Single Glance Duration to IVIS . . . . .	159
D.20.	Evaluation results Bootstrapping SGD to IVIS AAM 2 s limit . . . . .	159
D.21.	Evaluation results P85 Total Eyes-Off-Road Time . . . . .	160
D.22.	Evaluation results Bootstrapping TEORT NHTSA 12 s limit . . . . .	160
D.23.	Evaluation results P85 SGD eyes-off-road . . . . .	161
D.24.	Evaluation results Bootstrapping SGD eyes-off-road NHTSA 2 s limit . .	161
D.25.	Evaluation results Bootstrapping DLP . . . . .	162
D.26.	Evaluation results Bootstrapping DFH . . . . .	162

# Acknowledgements

It is worth noting that this thesis is not funded by any grant. Thankfully, Christina Krutzenbichler and Andreas Janiak wrote their theses at the institute, the Car Connectivity Consortium allowed data comparisons with a driver distraction project, and the Institute of Ergonomics (TUM) provided scientific infrastructure and enabled enlightening discussions.

In the last six years, I learned a lot from students, colleagues and project partners when they were writing their theses or working with me on projects. I also appreciate the insights provided in the final presentations and seminar talks of so many students and research associates at the institute. Within this rather large and bright group, I am particularly grateful to four persons, Prof. Klaus Bengler, Antonia Conti, Moritz Späth and Thomas Moll, who accompanied and guided me in diverse driver distraction projects over several years.

Thanks to Prof. Baumann, for the consent and effort to join the thesis committee.

I would like to thank my family, not only for the last years, but for also supporting me for 3.5 decades.

# Glossary

AAM	Alliance of Automobile Manufacturers
AMP	Accelerated Mobile Pages
ANR	Application Not Responding
AOI	Area of Interest
API	Application Programming Interface
APK	Android application package
ARV	Average Rectified Value
CDN	Content Delivery Network
CI	Confidence Interval
COG	Center Of Gravity
DC	Direct Current
DFH	Drift of Following Headway
DFT	Discrete Fourier Transform
DFT	Driver Focus-Telematics Working Group, AAM
DLP	Drift in Lane Position
DRT	Detection Response Task
DV	Dependent Variable
ESoP	European Statement of Principles on Human-Machine Interface for IVIS
EOR	Eyes-Off-Road
FH	Following Headway [seconds]
FHWA	Federal Highway Administration
fps	Frames Per Second
GOMS	Goals, Operators, Methods, Selections rules
HDRT	Head-mounted Detection Response Task
HMI	Human-Machine Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identification
IV	Independent Variable
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
IVIS	In-Vehicle Information and Communication System
JS	JavaScript
JSON	JavaScript Object Notation
KLM	Keystroke-Level Model
LANEX	Lane Exceedence
LC	Liquid Crystal
LCD	Liquid Crystal Display
LCT	Lane Change Test

LP	Lane Position
M	Mean
MAE	Mean Absolute Error
MANOVA	Multivariate Analysis Of Variance
MAPE	Mean Absolute Percentage Error
MHP	Model Human Processor
MIL-STD	United States Military Standard
MLP	Mean Lane Position
MTM	Methods-Time Measurement
NHTSA	National Highway Traffic Safety Administration
NOG	Number of Glances
OEM	Original Equipment Manufacturer
OTG	USB On-The-Go
p.p.	Percentage Point
P85	85 <sup>th</sup> Percentile
ppi	Pixels Per Inch
Q1	First Quartile
Q3	Third Quartile
RDRT	Remote Detection Response Task
RMS	Root Mean Square
RMSE	Root Mean Square Error
SAE	Society of Automotive Engineers
SD	Standard Deviation
SE	Standard Error
SDFH	Standard Deviation of Following Headway
SDLP	Standard Deviation of Lane Position
SGD	Single Glance Duration
SRT	System Response Time
SRD	System Response Delay
TCP	Transmission Control Protocol
TDRT	Tactile Detection Response Task
TEORT	Total Eyes-Off-Road Time
TGT	Total Glance Time
TSOT	Total Shutter Open Time
TTT	Total Task Time
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language

---

# 1. Introduction

A real-life insight into the topic of driver distraction was provided by the naturalistic driving study (NDS) within the American second Strategic Highway Research Program (SHRP 2). The SHRP 2 NDS used “[...] video, kinematic, and audio data [...]”, “[...] from more than 3,500 drivers across a 3-y period.”, “[...] capturing more than 35 million miles [...]”, “[...] comprising 905 injurious and property damage crash events [...]” (Dingus et al., 2016). Dingus et al. (2016) reports a prevalence of 3.53% of time for the use of in-vehicle devices and 6.4% for hand-held cell phones. Thus about 10% of the time, drivers are operating electronic devices; with a risk odds ratio of approximately 2.5–3.6.

In recent discussions regarding driver distraction, the automated car is often mentioned. Some argue that driver distraction problems will be solved by automated cars. Distraction can also be an issue when the automated car wants to return control to a distracted driver. The advocates for automation argue that a fully autonomous car solves the problem. The forecasts of the time horizons for automated and autonomous techniques are diverse. Even if an autonomous car could be constructed today, the internal processes of car manufacturers and administrations might add a decade until it could be purchased. When looking at the state reached during the Eureka PROMETHEUS Project (1987–1995) the progress of today’s autonomous cars, two decades later, can be put into a more reasonable time frame. While a tremendous amount of attention and money is provided to related projects, this allocation can hamper the research of problems better solved now.

Therefore, this thesis is in the field of manual-drive interaction modeling.

In Germany and Europe, the decreasing trend of fatal crashes has stopped and in Germany, over the last two years (2014, 2015), a slight increase in traffic deaths has been observed.<sup>1,2</sup> The increasing use of electronic devices is often mentioned as a plausible cause. Another factor has been also reported:<sup>3</sup> (Legal) medical treatments. The influence of pharmaceuticals on driving performance is ignored by many users. While interaction with an electronic device is on a short time scale and can be stopped, drugs can impair for hours. A possible classification of safety reducing factors on different time scales is shown in Table 1.1.

What is missing in the table is the frequency of use; while an IVIS task is typically a matter of mere seconds, it is possible to repeat or link them. Though electronic devices currently receive a lot of (media) attention, it is nevertheless worth mentioning some other

---

<sup>1</sup>FAZ 07/12/2016 <http://www.faz.net/aktuell/gesellschaft/ungluecke/mehr-verkehrstote-und-mehr-unfaelle-in-2015-als-2014-14337123.html> (accessed 08/07/2016)

<sup>2</sup>Zeit 03/28/2016 Erstmals seit 15 Jahren mehr Verkehrstote in der EU <http://www.zeit.de/mobilitaet/2016-03/verkehrstote-eu-strassenverkehr> (accessed 08/07/2016)

<sup>3</sup>SZ 06/07/2015 <http://www.sueddeutsche.de/auto/beunruhigende-unfallstatistik-unfallursache-raetselhaft-1.2504375> (accessed 08/07/2016)

Timescale	Impairment Examples
years	inappropriate education, insufficient discipline or time planning, speeding, weak points in infrastructure, wrong seating position, etc.
weeks/months	dangerously tuned car, wrong tires, bald tires, etc.
hours	medication, intoxication, exhaustion, haste, etc.
minutes	eating, talking, smoking, temporary speeding, incorrect mental model when it starts to rain, freeze or first snowfall etc.
seconds	IVIS tasks, grab or search an object, etc.

Table 1.1.: Timescales for driving safety impairments

contributing factors in Table 1.1 to illustrate a more comprehensive perspective of driving safety.

A lingering danger persists if a novice driver’s lessons never taught him/her that crossing cars can be hidden behind the A pillar (cf. Remlinger, 2013). The knowledge and teaching regarding driver assistance systems can also be improved in German driving schools (cf. Maier, 2013). Given the importance of long-term driver education, it is unfortunate that German television discontinued its famous TV show, ‘Der 7. Sinn’ (1966–2005). The weekly, three-minute-long educational film clips were broadcast for 39 years and received 45 international awards.<sup>4,5</sup> The lives saved by these clips are probably countless.

Sometimes the infrastructure itself can encourage dangerous situations. An example is a street in Hamburg which became famous for inducing unintended accelerations by more than ten drivers, who typically crashed into shop windows.<sup>67</sup> Fatal accident foci are usually tracked and mitigated by German road administrations (e.g., over a three-year duration on a pin map).

Over the last several years, trees have attracted some media attention.<sup>8</sup> On rural roads, 886 people were killed in collisions with trees (2006). On all German streets collisions with trees resulted in 1034 traffic deaths (2006).<sup>9</sup> Therefore, some German states planned

<sup>4</sup>Welt 05/03/2010 <http://www.welt.de/fernsehen/article7446004/Rueckkehr-des-TV-Ratgebers-Der-7-Sinn-gefordert.html> (accessed 08/07/2016)

<sup>5</sup>Wolfsburger Allgemeine Zeitung 02/24/2016 <http://www.derwesten.de/auto/experten-fordern-von-ard-rueckkehr-von-sendung-der-7-sinn-id11591812.html> (accessed 08/07/2016)

<sup>6</sup>Hamburger Abendblatt 02/20/2015 <http://www.abendblatt.de/hamburg/altona/article137654463/Wieder-Waitzstrasse-Seniorin-rast-mit-Auto-in-Bankgebaeude.html> (accessed 08/07/2016)

<sup>7</sup>Hamburger Abendblatt 03/21/2016 <http://www.abendblatt.de/hamburg/elbvororte/article207246183/Die-Waitzstrasse-bleibt-ein-gefaehrliches-Pflaster.html> (accessed 08/07/2016)

<sup>8</sup>Welt 02/24/2014 <http://www.welt.de/politik/deutschland/article125143927/Deutschlands-schoenste-Alleen-vor-der-Abholzung.html> (accessed 08/07/2016)

<sup>9</sup>Werner Köppel, Bonn 2008 7. Deutscher Verkehrsexpertentag der GUVU, Empfehlungen zum Schutz vor Unfällen mit Aufprall auf Bäume (ESAB 2006) <http://www.landsberg.bund-naturschutz.de/fileadmin/kreisgruppen/landsberg/Dokumente/Baumf%C3%A4llungen%20Alleen/ESAP2006.pdf> (accessed 08/07/2016)

---

to cut down all trees within a range recommended by guidelines on road safety.<sup>10,11,12</sup>

The law regarding distraction by phones in Germany appear quite confusing. In the first step, it is determined how an accident happened. If an accident happened and the driver was distracted (e.g., by a mobile device) the driver may be held responsible for some claims, e.g., by the insurance company. If no accident happened, fines can still be imposed: German §23(1)(a) *StVO*<sup>13</sup> mentions that it is forbidden to grab or hold a car phone (probably outdated) or mobile phone when the vehicle is moving or the engine is running. This is very specific as it is only applicable to car phones and mobile phones; satnavs, cameras, tablets, notebooks, walkie-talkies, calculators, voice recorders, music players, etc. are not included. In this sense, §23(1)(a) *StVO* seems inadequate and arbitrary. A driver that (accidentally and uselessly) operated a (short distance) home cordless phone<sup>14</sup> is beyond the scope of §23(1)(a) *StVO*. Another driver operated a hand-held mobile phone in front of a red traffic light in a start-and-stop car. The judgment agreed that that this could be allowed according to §23(1)(a) *StVO* due to the fact the engine was off.<sup>15</sup> A driver with an older car without start-and-stop would probably be fined in the same situation. In addition, when a driver places a phone in a dashboard cradle and enters a phone number or SMS, he/she also seems to escape being fined if no accident happens; despite potentially detrimental driver distraction (cf. Dingus et al., 2016, p. 2639, Fig. 2). A recent decision<sup>16</sup> allowed a driver to hold a bluetooth-coupled phone in a specific case (forgotten to put it down), renders the law even more confusing. The German minister of transport seems to be aware of this and wants to widen the scope of the law.<sup>17</sup> Avenoso (2012) provides a short overview of the varying overall distracted-driving regulations of some European countries.

There seems a clear cross-cultural understanding of basic forbidden actions, e.g., shop lifting. The indistinct topic of driver distraction could be an indication that it should be more an issue of engineering and driver education than arbitrary law enforcement. Drivers and situations are highly diverse. An interaction and situation that could be difficult for one driver might be responsibly managed by an experienced driver. Mobile phones while driving can be also used for beneficial purposes, for instance, a traffic light application on a smartphone has been extensively tested and optimized for use on arterial roads and has displayed some potential to voluntarily reduce speeding (Krause et al., 2014b).

---

<sup>10</sup>Richtlinien für passiven Schutz an Straßen durch Fahrzeug-Rückhaltesysteme (RPS),2009

<sup>11</sup>Zeit 08/09/2016 Der Baum als Feind <http://www.zeit.de/mobilitaet/2016-07/alleebaeume-autolobby-strassenbau-regeln> (accessed 09/24/2017)

<sup>12</sup>Uwe Ellmers (BaSt), Mehr Verkehrssicherheit trotz Bäumen am Straßenrand, 21. DVR Forum [http://www.dvr.de/download2/p4176/4176\\_3.pdf](http://www.dvr.de/download2/p4176/4176_3.pdf) (accessed 08/07/2016)

<sup>13</sup><https://dejure.org/gesetze/StVO/23.html> (accessed 08/07/2016)

<sup>14</sup><http://blog.burhoff.de/2009/11/olg-koeln-handyverbot-gilt-nicht-fuer-festnetz-mobilteil/> (accessed 08/07/2016)

<sup>15</sup><https://dejure.org/dienste/vernetzung/rechtsprechung?Gericht=OLG%20Hamm&Datum=09.09.2014&Aktenzeichen=1%20RBs%201/14> (accessed 08/07/2016)

<sup>16</sup>OLG Stuttgart, Beschl. v. 25.04.2016 - 4 Ss 212/16 [http://www.burhoff.de/asp\\_weitere\\_beschluesse/inhalte/3479.htm](http://www.burhoff.de/asp_weitere_beschluesse/inhalte/3479.htm) (accessed 08/07/2016)

<sup>17</sup>WAZ 08/13/2016 <http://www.derwesten.de/politik/dobrindt-will-das-handyverbot-am-steuer-ausweiten-id12094049.html> (accessed 09/24/2017)

---

When considering deaths in Germany, from 2005 to 2009, no airplane passengers were killed, compared to an annual average of 2,524 passengers and drivers in light vehicles during the same period.<sup>18</sup> In 2014, 3581 traffic deaths were reported in Germany.<sup>19</sup> Driving safety has evolved significantly over the previous decades and the reduction of traffic deaths now demonstrates a kind of ceiling effect. Further steps should therefore be expected to be rather small and probably expensive.

It is perhaps worthwhile to consider statistics. Approximately 10,000 people per year commit suicide in Germany (2013).<sup>20</sup> It is surprising, that with about 44 million light vehicles in Germany (2014)<sup>21</sup> and 5.5 million legal weapons (owned by 1.5 million people)<sup>22</sup> this ratio is inverted for the types of suicides (2013): 84 suicidal car accidents and 795 suicides by three different classes of weapons.<sup>20</sup> Despite easy access, vehicles seem either neglected by suicides or the classification of car suicides by investigators is biased toward ‘accidents’. If this assumption of bias has a reasonable foundation, the traffic statistics could be questionable or at least not directly useful in assessing traffic safety for non-suicidal road users.

A deception that could influence the property damage crash statistics is intentional accidents with the intent to defraud. The insurance companies estimate annual damage of up to 2 billion Euro in Germany.<sup>23</sup> Intentional car crashers select difficult situations and deceive other drivers into accidents to obtain money. The impending dash cams might be able to counteract such actions.<sup>24</sup> These usually non-severe events are also hidden in accident statistics. Non-fatal incidents are sometimes used in human factors analyses.

Overall, the potential influences on traffic accident statistics are endless, e.g.: the weather<sup>25</sup> or the population of wild animals and related deer crossings<sup>26</sup>. An undisputed

---

<sup>18</sup>Ingeborg Vorndran, Unfallstatistik - Verkehrsmittel im Risikovergleich [https://www.destatis.de/DE/Publikationen/WirtschaftStatistik/Verkehr/Unfallstatistik122010.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/Publikationen/WirtschaftStatistik/Verkehr/Unfallstatistik122010.pdf?__blob=publicationFile) (accessed 08/07/2016)

<sup>19</sup>Gesamtunfallgeschehen – Unfalltote und Unfallverletzte 2014 in Deutschland <http://www.baua.de/de/Informationen-fuer-die-Praxis/Statistiken/Unfaelle/Gesamtunfallgeschehen/Gesamtunfallgeschehen.html> (accessed 08/07/2016)

<sup>20</sup>Anzahl der Sterbefälle durch Suizid in Deutschland nach Art der Methode in den Jahren 2012 bis 2014 <http://de.statista.com/statistik/daten/studie/585/umfrage/selbstmordmethoden-in-deutschland-2006/> (accessed 08/07/2016)

<sup>21</sup><https://www.destatis.de/DE/ZahlenFakten/Wirtschaftsbereiche/TransportVerkehr/UnternehmenInfrastrukturFahrzeugbestand/Tabellen/Fahrzeugbestand.html> (accessed 08/07/2016)

<sup>22</sup>Zeit 01/16/2014 Waffenland Deutschland <http://www.zeit.de/2014/04/waffen-deutschland> (accessed 08/07/2016)

<sup>23</sup>Gesamtverband der Deutschen Versicherungswirtschaft <http://www.gdv.de/versicherungsbetrug/autobumser/> (accessed 08/07/2016)

<sup>24</sup>Versicherungsmagazin 06/02/2016 <http://www.versicherungsmagazin.de/Aktuell/Nachrichten/195/23141/Dashcam-Schadenaufklaerung-durch-Fremde-legitim.html> (accessed 08/07/2016)

<sup>25</sup>Welt 08/22/2016 <http://www.welt.de/motor/news/article157795348/Unfallstatistik-1-Halbjahr-2016.html> (accessed 09/24/2017)

<sup>26</sup>Mittelbayerische 04/04/2016 <http://www.mittelbayerische.de/region/schwandorf/gemeinden/burglengenfeld/die-wildunfaelle-nehmen-deutlich-zu-22389-art1362182.html> (accessed 08/07/2016)



---

key factor in accidents and traffic deaths is still speed.<sup>27</sup> Speed is directly linked to the severity of injuries.<sup>27</sup>

Car manufacturers and after-market suppliers typically want to provide customers some (non-driving related) functionality while driving. National and international guidelines, standards and voluntary commitments limit these potentially distracting tasks or indicate positive implementations. These countermeasures are incorporated into the development cycle as driver distraction testing. Some of these tests require a working prototype and significant effort (e.g., test laboratory, test subjects, data acquisition and analysis). If a new task fails, it can fail in a late stage of the development. The options could be to abandon the new functionality, lock it while driving or rework and repeat the testing. Nevertheless, these functions are designed for use while driving. Therefore, these special engineered solutions should be preferred over probably untested general purpose apps. However, even the (untested) navigation apps on smartphones could be better suited than the road-books and maps found on the co-driver's seat for many years.

The thesis attempts to find a way to predict the outcome (i.e. the distraction metrics) of a hypothetical task when a human factors specialist approximately knows the interaction steps. For this modeling, the measured values of several subtasks are gathered in a database then the potential of combining these subtasks into a complete task is evaluated. The findings are also used to illustrate how the current guidelines and standards may be improved.

The complete lockout of tasks while driving is perhaps comparable to the discussion of the ban on comfortable standby circuits in household equipment. Over time, the standby circuits were improved from initially consuming several watts to  $< 0.5\text{ W}$  according to EC 1275/2008<sup>28</sup>, nowadays. To block all non-driving-related tasks would also impede convenience. Another approach could be to engineer necessary tasks in a suitable way. To ban tasks in IVIS is theoretical; drivers could easily use their smartphone apps instead.

In brief, this thesis in the field of interaction modeling attempts to support laboratory driver distraction testing through inexpensive measures based on prediction models to mitigate (secondary task related) short time impairments and reduce the number of experiments.

The structure of the thesis:

In Chapter 2, *Fundamentals* specific for this thesis are covered. Chapter 3 *Building the Model*, describes the experiment which built the prediction model. Chapter 4, *Evaluation Experiment*, evaluates the experimental model. The final Chapter 5 *Conclusion* summarizes the outcomes and presents possible implications.

---

<sup>27</sup>WHO Fact sheet, Road safety – Speed [http://www.who.int/violence\\_injury\\_prevention/publications/road\\_traffic/world\\_report/speed\\_en.pdf](http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/speed_en.pdf) (accessed 08/07/2016)

<sup>28</sup>EC 1275/2008 <http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX%3A32008R1275> (accessed 08/07/2016)

---

## 2. Fundamentals

The fundamentals chapter focuses on basics specific to this thesis. The chapter has the following structure:

Section 2.1 provides a brief introduction to *Driver Distraction Guidelines*. These regional guidelines propose measurement methods, metrics and criteria in the assessment of driver distraction.

A literature review and discussion of the properties of delays can be found in Section 2.2, *Delays in System Response*. For the distraction modeling it is assumed (and later demonstrated), that system response delays can have a crucial influence on driver distraction metrics. The section clarifies the often-mixed *control activation feedback* and *dialog level system response*.

Section 2.3 covers *Driver Performance Metrics* and explains the approach used for this thesis. Due to the short time scale of the subtasks that are used for modeling and the additive capability, the drift in the lateral position (lateral velocity) and the rate of change in the time headway (drift in following headway) are the two metrics selected to assess driving performance. Both are related to a baseline driving performance to obtain a performance deterioration percentage.

Section 2.4, *Task Analysis and Modeling*, reviews preexisting task analysis and modeling methods.

In *Own Previous Work and Motivation* (Section 2.5), a reference to a recent industry cooperation and related experiments at the Institute of Ergonomics (TUM) is detailed. This section also holds the motivation and technical key points (requirements) for this work and the prediction model. This leads to the next chapter (Chapter 3 *Building the Model*).

## 2.1. Driver Distraction Guidelines

The main documents which address driver distraction are guidelines. This thesis aims and relies to some extent on these documents; the reader therefore needs at least some rudimentary understanding of these regional recommendations. An attempt to introduce (app) developers to this specific field of ‘suitability while driving’ was provided by Krause and Bengler (2015).

The relevant guideline for Europe is the ‘European Statement of Principles’ (ESoP) 2008/653/EC (2008) and related ISO standards. The intention of the document is to help developers rather than force them to comply with restrictive criteria. Therefore, different interfaces can be developed for a task and the best interface identified.

The American guidelines take another approach. They provide criteria and test methods: Driver Focus-Telematics Working Group (2006); NHTSA (2014); SAE J2364 (2004). It could be enough to develop one interface, as long it is below defined thresholds. American documents hold criteria that directly or implicitly limit the task duration. This is another difference from the European understanding, that the task length is not one of the most important parameters. The handling of continuous tasks (e.g., navigation) is another differentiator. While the Driver Focus-Telematics Working Group (2006) provides a procedure (assessment of driving performance) that could be applicable to the assessment of these continuous tasks, NHTSA (2014) is intended only for ‘testable tasks’ (which have a clear start and end).

Document	Occlusion Total Shutter Open Time [s]	Total Glance Time [s]	Single Glance Duration [s]
AAM/DFT	15	20	2
ESoP	–	–	(1.5)
JAMA	7.5	8	–
NHTSA	12	12	2

Table 2.1.: Criteria of guidelines

Table 2.1 presents an overview of criteria from different guidelines. While it is tempting to compare the different rows (guidelines), this is not easily possible. The metrics in the guidelines address different measurements and calculations, for instance, the glance times from NHTSA (2014) address eyes-off-the-road glances, while the Driver Focus-Telematics Working Group (2006) uses glances toward a task display. Other differences could be special task trainings and subject selection (e.g., JAMA, 2004) and the calculation of metrics (percentiles). The uncommon 1.5 s dwell time for the ESoP stems from the referenced ISO 15005:2002. The often-mentioned ‘2-seconds-rule’ in fact are three rules: One rule can be found in Driver Focus-Telematics Working Group (2006) and two rules are provided in NHTSA (2014). These are based on different metrics and calculations.

The previously mentioned guidelines are more complex than shown here. Especially for developers, it is essential to recognize the ‘principles’ of these guidelines. A principle

could be for example, that while interacting with IVIS, at least one hand must be on the steering wheel or that the contrast of characters is sufficient. A comparison of the guidelines can be found in Heinrich (2013).

At the end of November 2016, while this document was being completed, the NHTSA released a proposal with requests for comments (NHTSA, 2016). For task acceptance testing this ‘Phase 2’ document references the Phase 1 document (NHTSA, 2014). Therefore, the NHTSA criteria (Table 2.1) and discussions within this thesis, are still relevant for Phase 2.

When discussing system delays in Section 2.2, also the TRL checklists are mentioned (Stevens et al., 1999; Stevens and Cynk, 2011). To reach even further into the origins of driver distraction history, Carsten and Nilsson (2001) is a recommended read that includes some background information.

This thesis has a focus on the prediction of occlusion and glance metrics. Therefore, the description of the prediction model and the final discussion refers to the guidelines: Driver Focus-Telematics Working Group (2006) and NHTSA (2014).

## 2.2. Delays in System Response

The term delay is widely used, however, to describe an important characteristic in human-machine interaction, the word ‘delay’ alone is not specific enough.

A memorable definition of delays can be found in the ESoP (2008/653/EC, 2008).

From ESoP (2008/653/EC, 2008):

### 4.3.4.7. Interaction with displays and controls principle VII

*The system’s response (e.g. feedback, confirmation) following driver input should be timely and clearly perceptible.*

*Explanation: The system’s response applies at two levels:*

- *the control activation feedback level, e.g. button displacement, auditory beep,*
- *the dialogue level, which is the system’s response to the driver’s input, e.g. recommended route.*

*The system’s response is timely if it is perceived as quite instantaneous. For control activation feedback, timing should be from the moment at which the system recognizes each driver input. For the dialogue level response (which may be either the requested information or an indication that processing is underway), the timing should be from the end of the driver’s input. [...]*

This idea differentiates between *control activation feedback* and *dialog level system response*, which is interpreted and illustrated in Figure 2.1. This separation can be seen as a condensed and simplified concept of the 17 ‘topics’ from Miller (1968). A drawback is the statement “[...] *timing should be from the moment at which the system recognises each driver input.*”. A system with a low sampling or detection rate of user actions would benefit from its own inability.

The input philosophy (on-release or on-press activation) has obvious implications for the example. Figure 2.1 assumes a widespread on-release paradigm, that allows correction or gesture recognition before an action is triggered. In the example, the user touches the screen over a virtual button. The system recognizes the user action and after a technical feedback lag the button is colored to give instantaneous control activation feedback (first level). The technical feedback lag can consist of: the time needed to sample and preprocess some physical data by the touchscreen hardware (digitizer) and driver, forwarding the data to the operating system, event handling by the application and drawing into a frame buffer and transmitting the frame to a screen.

The user then lifts a finger to trigger an on-release event. After another feedback lag period, the system decolors the button (first-level feedback). Because the action triggers a long calculation, a message informs the user about the current state of the calculation (dialog/second-level feedback). When the calculation is finished, a green tick (second-level feedback) shows the users the end and success of the operation; e.g., the calculated navigation route. A system may even allow the user to cancel a long-lasting operation.

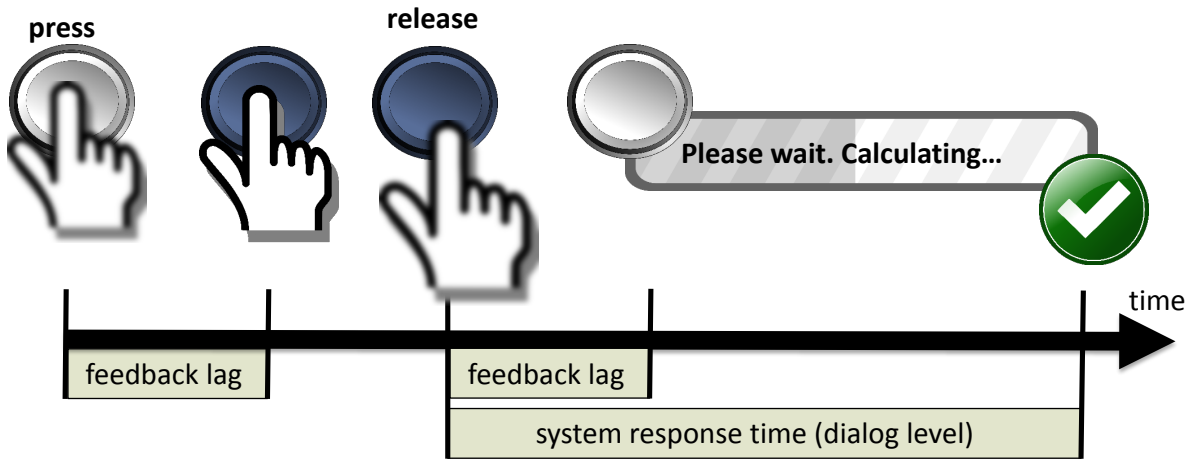


Figure 2.1.: Delay levels (cf. 2008/653/EC, 2008, Principle 4.3.4.7.); illustration (cf. Kaaresoja and Brewster, 2010, Figure 2 and Figure 3)

This interaction concept can be generalized and is known from other situations: If one writes a letter to an agency, the agency can quickly respond that the request has been received. The final answer to the question can take a while. The same is true for everyday conversations, when someone gets a question and has to think about the answer. A first-level expression (e.g., nodding) can signalize that the question has been received. If an extended thought is required, some more gestures and feedback may be needed. At least four different status information needs of a user are involved:

- the interaction partner is ready for interactions (current appearance, end of former interactions)
- acknowledgment that an interaction fragment has been received (first-level feedback; could be supported by second-level onset)
- a request is currently processed (optional second-level feedback)
- and finally a dialog result is available (second-level feedback)

The first-level (control) feedback typically is a combined, ‘crisp’, single-stage event (e.g., highlight a button, play a click), while the second level (dialog) can support the first-level feedback with a simultaneous onset and may smoothly evolve (fade in a dialog, animate progress indicator, show final result). Handling discussions about delays with the two-level concept in mind might solve some problems (e.g., the often-discussed long-press gesture). An indication that an event is a second-level feedback is obviously that a first-level feedback occurred before. The first level is often on a short timescale. Indicating words could be: feedback, lag, latency, propagation, transport delay. The second level is often connected to words like: idle, wait, response time.

The first- and second-level feedbacks are third party or external confirmations (e.g., from a computer). When someone operates a button, touches a screen or talks, s/he also has a self-induced, natural feedback, for example, when feeling the haptic click of a mechanical button, noticing the touch and release of the touchscreen glass surface or

hearing his/her own voice when talking to speech recognition. In Figure 2.1 this can be interpreted as a fundamental ‘zero level’ feedback and should be the reference for the time taken when specifying delays. In human-computer interaction, this physiological self perception (zero level) should work hand in hand with the first-level feedback.

Nielsen (1993) discusses three timescales:

[...]

**0.1 second:** Limit for users feeling that they are **directly manipulating** objects in the UI. For example, this is the limit from the time the user selects a column in a table until that column should highlight or otherwise give feedback that it’s selected. Ideally, this would also be the response time for sorting the column — if so, users would feel that *they* are sorting the table. (As opposed to feeling that they are *ordering* the computer to do the sorting for them.)

**1 second:** Limit for users feeling that they are **freely navigating** the command space without having to unduly wait for the computer. A delay of 0.2-1.0 seconds does mean that users notice the delay and thus feel the computer is "working" on the command, as opposed to having the command be a direct effect of the users’ actions. Example: If sorting a table according to the selected column can’t be done in 0.1 seconds, it certainly has to be done in 1 second, or users will feel that the UI is sluggish and will lose the sense of "flow" in performing their task. For delays of more than 1 second, indicate to the user that the computer is working on the problem, for example by changing the shape of the cursor.

**10 seconds:** Limit for users **keeping their attention** on the task. Anything slower than 10 seconds needs a percent-done indicator as well as a clearly sign-posted way for the user to interrupt the operation. Assume that users will need to reorient themselves when they return to the UI after a delay of more than 10 seconds. Delays of longer than 10 seconds are only acceptable during natural breaks in the user’s work, for example when switching tasks.

[...]

The first (0.1 s) and second (1 s) limit from Nielsen could be mapped to the two interaction levels: control level (manipulation) and dialog level (navigation). The third threshold (10 s) could be a relevant upper limit for (second-level) delays in IVIS interactions; as attention is crucial while driving. When a secondary task further increases workload, due to additional reorientation caused by long delays, it might be deemed unsuitable for use while driving.

In an ESoP draft (2005) the two-level statement mentioned before was further specified by a time limit, which was later removed. (ESoP draft, 2005, p. 28, Principle 4.7):

*The system’s response is timely if it is perceived as quite instantaneous, i.e. within a time of 250 ms. For control activation feedback timing should be from the moment at which the system recognises each driver input. For the dialogue level response (which may be either the requested information, or an indication that processing is underway) the timing should be from the end of the driver’s input.*

*When the system’s processing time requires longer than 250 ms, some signal should be displayed after 250 ms to inform the driver that the system has recognised the input and*

*is preparing the requested response.*

This could allow the interpretation that the 250 ms should apply at both levels.

The two-level statement mentioned before has been also used in the AAM Principle 3.5 (Driver Focus-Telematics Working Group, 2006, p. 72); but the wording (*‘quite instantaneous’*) has been modified to: *“The system’s response is timely if it is clearly perceived as reacting as expected”*. Also a slightly different sentence for the criteria is used: *[...] Criterion/Criteria: The maximum system response time for a system input should not exceed 250 msec. If system response time is expected to exceed 2 seconds, a message should be displayed indicating that the system is responding [...]*  
*The 250 msec provision is adopted to be consistent with ISO 15005. [...]*

For an average reader, these criteria merge the two-level concept into one. The previously cited ISO 15005 seems similarly unaware of two levels. This whispering down the lane resulted in a shortened adaption into NHTSA (2012):

*[...] V.10 Response Time. A device’s response (e.g., feedback, confirmation) following driver input should be timely and clearly perceptible. The maximum device response time to a device input should not exceed 0.25 second. If device response time exceeds 0.25 second, a clearly perceptible indication should be given indicating that the device is responding. [...]*

According to NHTSA (2013, p. 223): *“With this recommendation, NHTSA intended to match the recommendations of the Alliance Guidelines Principle 3.5 and ISO 15005: 2002.”* At a first glance, the statement above (V.10) seems similar to ESoP draft (2005, Principle 4.7) and item C10 in the TRL checklist (Stevens and Cynk, 2011, p. 46). Nevertheless, as can be seen by the additional checklist item C9 and the wording *“Following control activation feedback [...]*”, the TRL checklist operates with two levels, assumes they are sequential and applies a 250 ms recommendation to the second level:

*C9 Is control activation feedback adequate and appropriate? [...]*

*C10 Following control activation feedback, is the required information provided within an appropriate timescale?*

*The IVIS response (e.g. feedback, confirmation) following driver input should be timely and clearly perceptible; if there is a time lag visual distraction may increase or the driver may try and activate the control again.*

*When the system’s processing time requires longer than 250 ms, some signal should be displayed within 250 ms to inform the driver that the system has recognised the input and is preparing the requested response. [...]*

In a former version of the checklist (Stevens et al., 1999), the related items were C7 and F5.1; both recommending 250 ms. Tracing back the wording, it is likely that TRL is the source of *“[...] response (e.g. feedback, confirmation) following driver input should be timely and clearly perceptible [...]*”.

The comments and answers (NHTSA, 2013, pp. 221–224) to the NHTSA proposal do not use the two-level concept and indicate some confusion: *“[...] NHTSA again care-*



fully reviewed this principle and researched the Alliance's rationale for this criterion." This resulted in the final principal of NHTSA (2013); disregarding the salutary two-level concept:

[...] *K. Device Response Time.*

1. *A device's response (e.g., feedback, confirmation) following driver input should be timely and clearly perceptible.*
2. *As a "best practice," the maximum device response time to a device input should not exceed 0.25 seconds. The measurement of this time should begin starting at the completion of the driver's control input.*
3. *If a device's response time exceeds 2.00 seconds, a clearly perceptible indication should be given indicating that the device is responding. Again, the measurement of this time should begin starting at the completion of the driver's control input.[...]*

A slight difference between the NHTSA guideline statement and the Alliance Guidelines (AAM/DFT), is the wording "[...] exceeds 2.00 seconds [...]" (NHTSA, 2013) compared to "[...] expected to exceed 2 seconds [...]" (Driver Focus-Telematics Working Group, 2006). It is assumed that this difference was unintentional, but it can provoke some thoughts: Expectations about System Response Times can be made during the implementation and, e.g., hard-coded by a programmer. Also, the system itself may make expectations (e.g., based on download speed) and react dynamically. These expectations may lead to assumptions that a delay is longer than 2 seconds, before 2 seconds are already over (by knowledge or prediction). Another solution could be an implementation that supervises its own program flow and, when a two-second delay is exceeded, an indication is enabled (guarding).

For both (AAM and NHTSA guidelines) it is unclear if the message should be shown directly (e.g., within 250 ms) or after 2 s. The TRL checklist would be clear ([...] *some signal should be displayed within 250 ms [...]*). If an indication on dialog level is given directly (and not after 2 s) it might support the first-level feedback and be easier to implement. On the other hand, the 2 s or 1 s (MIL-STD-1472G, 2012, 5.1.2.1.4.h, p. 23) may can be used to suppress superfluous second-level indications (cf. p. 221 Mercedes-Benz NHTSA, 2013; Nielsen, 1993); assuming that appropriate first-level feedback is already provided in another way. The source for the recommended 2 s is unclear. One source could be the informational annex of DIN EN ISO 9241-1 (1997) (2 s response time limit for menu interactions).

Because the long-press gesture (e.g., to save a radio station) is mentioned in AAM and NHTSA discussions, some thoughts: The user needs feedback that the key is depressed (first level), for instance, on a touchscreen by hover coloring and an initial beep. The user needs first-level feedback again when the system recognizes the long-press gesture, e.g., a beep with a different pitch. This is common practice and could be explained with the two-level concept. Therefore, it is unclear why this example complicates discussions and needs an explicit exemption from the AAM Principle 3.5 (Driver Focus-Telematics Working Group, 2006, p.74). Whether a long-press gesture is suitable for an IVIS is not part of this thesis.

Miller (1968) mentioned the point in time of a delay during a task: “*The rule is that more extended delays may be made in a conversation or transaction after a closure than in the process of obtaining a closure.*”. Closure means the termination of a subtask. Kohlisch and Kuhmann (1997) further differentiate between *intra-task* and *inter-task* delays: “[...] a user may be forced to keep a provisional result in memory during an *intra-task* SRT [...]”

An extensive and free of charge resource for human factors engineering is MIL-STD-1472G (2012). In 5.12.1.4 (p. 277) the standard specifies round-trip times (delays) for virtual environments regarding simulator sickness. The round-trip time for a system shall not exceed 100 ms (preferably 75 ms). The update for head-mounted displays due to head movement shall not exceed 16 ms. The latency limit for unmanned aerial vehicles (5.12.3.2.4, p. 284) shall not exceed 100 ms. For unmanned ground vehicles the teleoperation round-trip shall not exceed 250 ms for the vehicle control and 100 ms for the weapon systems (5.12.3.3.4, p. 285). The general response time criteria for displays (5.1.2.1.4.d, pp. 23–24) differentiate between real-time systems and non-real-time systems and provide a table with 13 acceptable response times for different interactions (see Table 2.2). A two-level concept is not mentioned, but would split the table into control activation (first level) feedback of 0.1–0.2 s and dialog level feedback (second level) of 0.5–10 s. When compared to Table XXII in MIL-STD-1472F (1999, p. 196) the error feedback (0.2 s) could be a misprint (MIL-STD-1472F (1999): 2.0 s). While MIL-STD-1472F (1999) and MIL-STD-1472G (2012) do not provide references, there could be a connection to the suggested values from Miller (1968).

System Interpretation	Response Time Definition	Time(seconds)
Key response	Key depression until positive response, e.g., "click"	0.1
Key print	Key depression until appearance of character	0.2
Page turn	End of request until first few lines are visible	1.0
Page scan	End of request until text begins to scroll	0.5
XY entry	From selection of field until visual verification	0.2
Pointing	From input of point to display point	0.2
Sketching	From input of point to display of line	0.2
Local update	Change to image using local data base, e.g., new menu list from display buffer	0.5
Host update	Change where data is at host in readily accessible form, e.g., a scale change of existing image	2.0
File update	Image update requires an access to a host file	10
Inquiry (simple)	From command until display of a commonly used message	2.0
Inquiry (complex)	Response message requires seldom used calculations in graphic form	10
Error feedback	From entry of input until error message appears	0.2

Table 2.2.: Acceptable System Response Times from MIL-STD-1472G (2012, Table V, p. 24)

5.1.2.1.4.h (p. 23) states that if a delay is longer than 1 s, the user must be informed and for delays exceeding 10 s, a count-down is required. More generally, this is also mentioned in 5.1.3.4.b (p. 41). In 5.1.3.3.3.f (p. 37), it is specified for joysticks that the delay between control movement and display shall be not greater than 0.1 s. In 5.1.3.5.1.d (p. 44) two response-time related concepts are specified and explained (response-time induced keyboard lockout and keyboard restoration).

It must be mentioned that most automobile infotainment tasks are discrete by definition to achieve interruptibility and therefore consist of time-discrete interactions (e.g., single button presses when entering a phone number). While some of the previously mentioned (transport/round-trip) delays are specifications for continuous interactions (e.g., moving in a virtual environment or remotely operating a vehicle). Continuous tasks are typical

in the fields of tele-robotics, remote-operated driving or camera-monitor-mirrors. These and related fields are out of the scope of this thesis. The values above are mentioned to approach an initial understanding of technical feasibility and requirements. These data reveal that a time limit for first-level feedback should be 100 ms (cf. Miller, 1968). In Kaaresoja and Brewster (2010), it can be seen that even a power-restricted embedded system (i.e., a mobile phone) approached this requirement for discrete interactions years ago (Nokia 5800, released at the end of 2008). This delay recommendation (100 ms) targets visual/manual interfaces. The recommendation ITU G.114 (2003) includes (modeled) ratings of user acceptance regarding delay in speech transmission which may be useful for speech interfaces.

Some experiments and real-life examples regarding delays are mentioned and reviewed with the two-level concept in mind:

Rassl (2004) implemented a surrogate phone interface to enter a phone number with a rotary knob. During an experiment in real traffic, the visual feedback was delayed in four conditions by 0.1 s, 0.2 s, 2 s and 3 s. According to the description, the subjects were trained without delay and blindsided in the experiment by the different delays. The two short delays and two long delays were grouped in analysis. The total task on time was more than doubled for the long delays (31 s to 73 s), also the total glance time (17 s to 35 s); there was no significant difference in the mean Single Glance Duration ( $p = 0.34$ ). According to the data sheet<sup>1</sup> the reported rotary encoder had a detent torque of 15 mNm (and a 52 mm-diameter cap), this provided haptic feedback (zero-level feedback). When interpreting the setup in the context of the ESoP feedback levels, Rassl implemented a first-level delay. The visual channel (screen) was continuously delayed. Continuous first-level delays of 2 s and 3 s are nevertheless rare. A signal from the earth to the moon would need about 1.3 s (i.e., round-trip 2.6 s). Modern communication protocols sometimes gather data in a buffer to, for example, reduce data redundancy (compression) or enhance transmission characteristics (interleaving), which can cause different delays.

Utesch and Vollrath (2010) implemented a surrogate IVIS menu with delays (System Response Time) and tested it with the LCT method. In the study, the delay length was manipulated (0 s, 0.5 s, 1 s) and the delay type (constant, variable) as well as an additionally acoustic click after the delay (*‘which indicates input readiness’*) were included as parameters. In the variable condition, the delays were randomly varied in the range of  $\pm 50\%$ . The delay was inserted when users jumped from (hierarchical) menu level to menu level, but not when navigating within a menu level layer. For system operation, the arrow keys of a hardware keyboard were used. The subjects were not instructed beforehand about delays occurring. No main effect of delay length on driving performance was found. Constant delays led to better driving performance. The subjects found the delays generally annoying and some the acoustic feedback also. The ESoP level concept is not addressed in the paper. It can be assumed that the hardware keyboard provided a characteristic mechanical feedback (zero level). The delay when navigating from menu level to menu level would be a typical situation for a second-level delay (dialog level); when appropriate first-level feedback would be given before. From the description (System Response Time),

---

<sup>1</sup>Alps Datasheet 2004, 8-directional Switch and Encoder with a Center Push RKJXT Series, <http://de.onlinecomponents.com/datasheet/rkjxt1e12001.aspx?p=10114295> (accessed 04/17/2016)

it seems that first- and second-level feedback was mapped into one delay. The times (0.5 s and 1 s) are long for a first-level delay. Subjects had to keep the announced task goal (menu item) in mind, therefore it can be further specified as intra-task delay.

Constant delays were also mentioned by Miller (1968). Miller provides an example: organists can compensate for the constant operational delay from a key press until a tone comes out of the pipes and travels to the ears. In another example, the processing time of a hypothetical employee's badge-reader would benefit from a fixed length of time in Miller's *Topic 5*, regarding usability. In the view of the ESoP levels, the organist would compensate for a constant first-level feedback and the workers would get used to a constant second-level delay when presenting the badge, which would allow behavioral automatism. Eagleman (2009) reports an artifact of the calibration of the human brain to delays: When a human is adapted to an (artificially injected) short delay between a self-actuated action and a sensation, removing the injected delay can create an illusion that the sensation happened before the action. The *motor-sensory recalibration* experiments (for typical 100 ms delay) are described in Stetson et al. (2006): For longer injected delays (250 ms, 500 ms, 1000 ms) the adaption effect decays.

Anderson et al. (2011) differentiate between *initial latency* and *continuous latency* and tested different durations from 80 ms to 780 ms regarding subjective ratings. For some systems, the initial delay is needed, e.g., to recognize gestures. The rating dropped with delay length. The continuous delay was only slightly more annoying than the initial latency alone. When classified by the ESoP levels, the experiment principally addresses first-level feedback. Also noteworthy is the accurate notation of the delay in the study (80 ms). Even when the experimenter wants a 0 s delay (physical impossible), there are always the (baseline) delays of the systems used (see also Stetson et al., 2006).

Lee et al. (2016) included an experimental condition with a delay: “[...] showed the result of each entry only after a delay of 500-1200ms, which was drawn from a uniform. However, participants were able to type multiple letters ahead.” The delay was randomized for every keystroke, the virtual keyboard provided some first-level feedback (highlighting)<sup>2</sup>. If one assumes that the display of a typed letter is typically part of a first-level feedback, the artificial delay condition splits this apart and the display of letters is shifted to a second-level feedback. Perhaps this is irritating for test subjects. The study focused on glance strategies during error recovery. The system with delay led people to more often choose the strategy with an additional glance toward the road, during error recovery. In the discussion, this is condensed to: “[...] immediate feedback makes drivers visually focus longer on the task.” On the other hand, it is not discussed how the driving metrics are influenced by this type of delay. The figures in the paper hold indications for a deterioration in delay conditions. It would be reasonable, if it is more challenging to handle two lagging systems (the car and the randomly delayed IVIS).

---

<sup>2</sup>specified and clarified on 04/25/2016 by communication with J. Y. Lee via [https://www.researchgate.net/publication/295854663\\_Error\\_Recovery\\_in\\_Multitasking\\_While\\_Driving](https://www.researchgate.net/publication/295854663_Error_Recovery_in_Multitasking_While_Driving) (Comments)

Lee et al. (2015) also used a system with a delay in one experimental condition. The delay was further specified by online communication<sup>3</sup>: 800 ms–2500 ms. The study focused on the glance behavior at a task boundary (pressing the next button between screen reading). The button provided a first-level feedback (highlighting). The results indicated that when the delay was inserted after the button press, the behavior of keeping the eyes on the IVIS, shortly after the press, was diminished. The duration and appearance would fit a second-level feedback delay.

A special method to enrich first-level feedback or bridge between the first and second levels to enhance the user experience, could be animations (cf. Bengler and Broy, 2008). An animation can transfer valuable information (regarding the developer’s intention) to build or encourage a specific mental model. In the example of Bengler and Broy (2008), animations of 0 ms, 300 ms and 1500 ms are tested in an occlusion experiment. An animation shows that a configuration menu is on the back of a navigation map, which can be rotated (animation) by a hardware button. With a 0 ms animation, this has the appearance of just showing a configuration screen. The 300 ms animation was preferred by almost all test subjects and no statistical deterioration can be found regarding the task times during occlusion. The 1500 ms annoyed the users and revealed a deterioration in task times. Animations are twofold: while visual entertainment is forbidden (cf. driver distraction guidelines), an animation may have merit in enhancing guidance and learnability (Bengler and Broy, 2008).

Measures to limit or mitigate the negative user experience caused by delays are hard-wired into the Android mobile operating system. Android monitors if an application responds to a user input within 5 s and, if not, generates an Application Not Responding (ANR) dialog (Google, 2016b). The dialog allows the user to terminate a frozen program. These ANR events are sent to the developer. Therefore, the developer is aware of the problem and can work on it. Since Android 3.x (Honeycomb, API level 11), the system will not permit a developer to open a network connection in the main thread (Google, 2016c). The main thread handles the user interface. This forces developers to implement appropriate threading and keep activation control feedback (first level) and, e.g., downloads (second level) separated. A long-lasting download can not render the user interface unresponsive.

In addition, recent technological progress has addressed data transmission delays: Google’s Accelerated Mobile Pages (AMP) Project<sup>4</sup> helps fast rendering web-pages and therefore shortens HMI delays, e.g., when browsing the web. Additionally, HTTP/2<sup>5</sup> and the related SPDY<sup>6</sup> can speed up data transmissions. The increasingly decentralized Content Delivery Networks (CDN) should also help to provide data quickly.

---

<sup>3</sup>specified and clarified on 04/25/2016 by communication with J. Y. Lee via [https://www.researchgate.net/publication/281294809\\_Secondary\\_Task\\_Boundaries\\_Influence\\_Drivers'\\_Glance\\_Durations](https://www.researchgate.net/publication/281294809_Secondary_Task_Boundaries_Influence_Drivers'_Glance_Durations) (Comments)

<sup>4</sup>Accelerated Mobile Pages Project 2016, <https://www.ampproject.org/> (accessed 04/24/2016)

<sup>5</sup>IETF HTTP Working Group 2016, HTTP/2, <https://http2.github.io/> (accessed 04/24/2016)

<sup>6</sup>Google 2015, SPDY, <https://developers.google.com/speed/spdy/> (accessed 04/24/2016)

With Android Auto there is a framework for developing and using (specialized) apps in a connected car and smartphone setup. These apps must fulfill at least 26 quality criteria (Google, 2016a). Three of these quality criteria explicitly address delays:

[...]

*App-specific buttons respond to user actions with no more than a two-second delay.*

[...]

*App launches in no more than 10 seconds.*

*App loads content in no more than 10 seconds.*

As mentioned above, the AAM guideline (Driver Focus-Telematics Working Group, 2006) address delays directly with length criteria. In the guideline, delays are also indirectly mentioned on pp. 41–43 in the discussion of ‘check glances’:

(p. 41) “[...]new technologies might produce many very short ‘check’ glances, which, individually, are not likely to be a problem. For example, a system request with a long response time might prompt the driver to use several very short (e.g., 300 ms in duration) glances to see if the response has arrived and is displayed. Thus, limiting the number of glances when short check glances are included appears overly conservative in such an instance. Instead, a limit on total glance time to task-related controls and displays is offered.”

(p. 43) “[...]While the system is busy retrieving the information as indicated, for example, by an hour glass symbol the driver will typically perform very short ‘check glances’ of less than 300 ms in duration, typical of the glances used to check instrumentation.”

(p. 43, footnote) “[...] to address the concern that there may be many, rather than just one or two, such check glances, multiple check glances not intervened by a control action are considered part of the visual demand of the function or feature and should be included as part of the calculation pending further research.”

This indicates that the authors were aware that (second-level) delays can probably influence the measurement of glance metrics. They assumed (very short) 300 ms glance durations, which would tremendously decrease single glance metrics when combined with typical 1–2 s glances. ‘pending further research’ is an indication that experimental data was missing.

This lack of experimental data was also documented in ISO 16673 (2007). The annex makes informative suggestions concerning how System Response Delays (SRD) can be handled in an occlusion experiment. After some assumptions regarding check glances, the standard states: “It should be noted, however, that to date little research is available on the effects of SRD on driver visual demand. It is not known, for example, to what extent visual demand varies with the length of an SRD. It is not known to what extent drivers use the SRD periods to look at the road (vs. glance at the device or system). Further, the mode and content of indicators used to inform drivers that an SRD is active or terminated may have differing effects on visual demand and eye glance behaviour. [...] Thus, empirical research is limited, on SRDs as well as on SRD-state indicators, and on the effects these have on visual demand. [...] Users should understand that when an SRD is involved in a task, it may be most appropriate to set aside occlusion-based methods and instead apply direct measurement of eye glances.”

The annex then makes assumptions and recommends how the influence of SRDs on oc-

clusion metrics may be subtracted.

To summarize properties that can characterize a delay: Response delays can be on a first level (control activation feedback) or on a second level (dialog, system status). They can independently appear on the visual, auditory or haptic feedback channels (cf. Kaaresoja and Brewster, 2010). During the delay, further input can be possible or the system is stalled. Perhaps the delay (dialog level) is cancelable. Second-level delays can be visualized with indetermined signals (e.g., static splash screen, circle animations, barber poles), determined signals (e.g., percentages, progress bars) or none (e.g., the system appears frozen). The delay time could be a technical requirement or artificially injected (e.g., for experiments and engineering). During an intra-task delay, the user may have to keep data in mind (cognitive effort), while an inter-task delay without cognitive effort in some situations may help to regenerate (cf. Kohlisch and Kuhmann, 1997). A delay itself can be initial (e.g., due to gesture recognition to switch into zoom-mode) and/or continuous (cf. Anderson et al., 2011). The task characteristic could be discrete (e.g., entering numbers on a number pad) or continuous (e.g., teleoperation of a car). The point in time of a delay can be predictable for the user (e.g., a splash screen on startup) or unpredictable (e.g., the operating system or thread stalls). Furthermore, the duration of a delay can be predictable (e.g., constant) or unpredictable. The duration of an unpredictable delay can be also near zero (e.g., some data was locally cached before) or can (inscrutably for some users) switch between values (e.g., cached/not cached or fast WiFi/slow GSM connection). Unpredictable durations (sometimes zero) will likely also mask the predictability of the point in time.

A common understanding of System Response Time (SRT) is offered by Kohlisch and Kuhmann (1997): “[...] is defined as the time elapsed from entering a command until its completion. During SRT, new user commands are not accepted because the computer is busy.” Therefore, SRTs are not synonymous with *delays*. SRTs are a special subtype of delays: Typically a second-level delay which cannot be canceled and often stalls the input of the system/user interface; some wrongly designed systems will even stall the output (no ongoing user indicator).

## 2.3. Driver Performance Metrics

It is always advisable to use established metrics. When one deviates from this principle, it needs justification concerning the reasons why the common methods are unsuitable. This is provided in the following section regarding the way driving performance metrics are handled in this thesis.

Two common driving metrics to assess lateral and longitudinal driving performance in a constant car-following task are:

- Standard Deviation of Lane Position *SDLP* (cf. Knappe, 2009; SAE J 2944, 2013; Östlund et al., 2005; DIN EN ISO 17287, 2003)
- and the Standard Deviation of the Following Headway *SDFH* (cf. Driver Focus-Telematics Working Group, 2006)

The (Following) Headway is defined for this thesis as the tip-to-tail distance divided by the speed of the following vehicle. This is in accordance with Driver Focus-Telematics Working Group (2006, p. 45), which indicates with ‘inter-vehicle range’ and ‘range-rate’ that the distance measurement from a radar is probably used. In this thesis, the recorded speed of the simulated vehicle is used in the headway calculation (including the small lateral component); accelerations are not incorporated into this calculation. The unit of following headway is *seconds*. In SAE J 2944 (2013) the term ‘Time Gap’ is proposed and (Time) Headway is used in a slightly different way (tip-to-tip; when do two vehicles pass the same landmark). Due to the calculation of a standard deviation, the difference (constant length offset of the leading car) is not essential in this thesis.

The assessment of task performance by calculation of a Root Mean Square Error (RMSE) can be seen as a special case of calculating a Standard Deviation (SD). The RMSE has a long tradition in human factors engineering of evaluating the performance in tracking tasks (mean power of an error signal).

$$RMSE_{discrete} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\text{lanePosition}_i - \text{predefinedLanePosition})^2}$$

$$MLP_{discrete} = \frac{1}{N} \sum_{i=1}^N \text{lanePosition}_i$$

$$SDLP_{discrete} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\text{lanePosition}_i - MLP)^2}$$

The SD has the benefit over the RMSE that it automatically adapts, to some extent, to the individual subject behavior. For example, if a driver has a tendency to drive closer to the right lane marking during the car-following task, the SD will assess the deviations from this individual strategy. A RMSE calculation with the default assumption that all drivers would or should drive in the middle of the lane will give a slightly different result. When an individual behavior/strategy (e.g., driving in the middle of the lane) is the same as the assumption for the RMSE, SD and RMSE calculations become identical. Therefore, a potential offset, i.e. Mean Lane Position (MLP), is inherently calculated into the RMSE calculation. Standard Deviation and RMSE can become problematic when drivers



adapt or vary strategies to situations during longer analysis periods (cf. Knappe, 2009, p. 49). Due to this low frequency components SD calculations can be duration dependent (see Figure 2.2); i.e., despite the implicit normalization (averaging by number of samples), the same task would display a higher variance if performed longer (cf. Östlund et al., 2005, p. 36, p. 39). The comparison of SDLP for tasks with different lengths is therefore questionable. The Modified Lateral Position Variation (MSDLP) in Östlund et al. (2005) attempts to counteract such effects by high-pass filtering (e.g., 0.1 Hz) as can be seen in Figure 2.2. This can be also transferred to longitudinal metrics (cf. Östlund et al., 2005, p. 36).

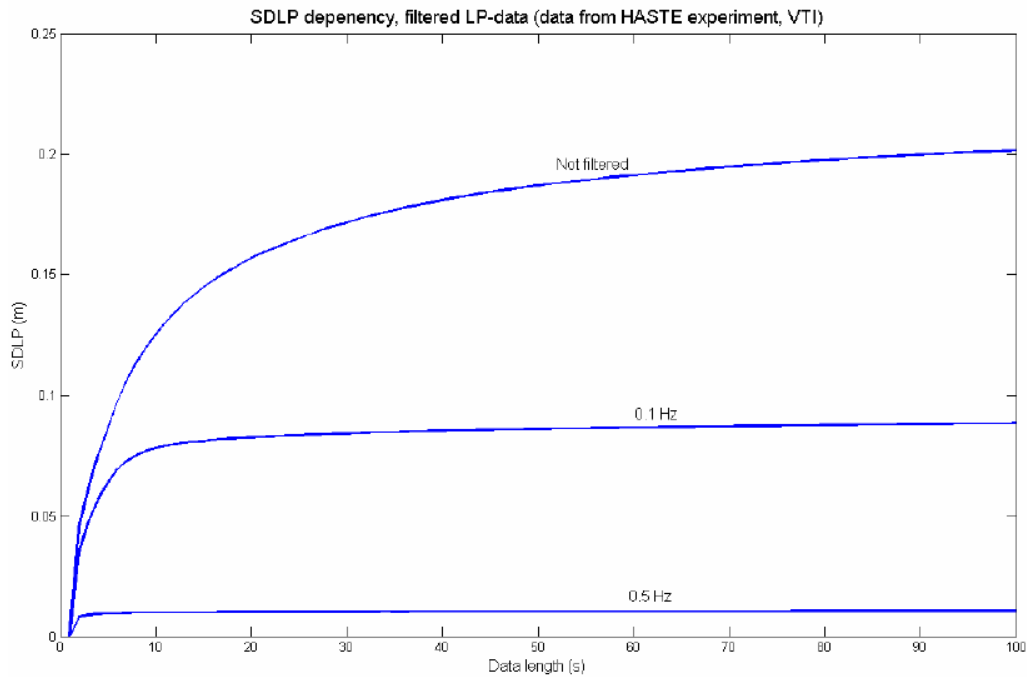


Figure 2.2.: SDLP values calculated with different data lengths (unfiltered, high-pass filtered with 0.1 Hz and 0.5 Hz) from Östlund et al. (2005, p. 39; Figure 7)

Figure 2.3 presents the lane position data of 24 persons reassessed from an experiment reported in Krause et al. (2015a). For each person, the figure includes one baseline drive and three trials while tuning radio frequencies on different devices. Therefore,  $4 \times 24 = 96$  trajectories. Each trial started from standstill and evolved into the the car-following task. After an initial 30s (approximately 500m), the measurement data were analyzed. At this point in time, the test subjects started radio tuning (in non-baseline trials). If a trajectory crosses the vertical blue lines it indicates lane exceedances (LANEX) according to the AAM definition. This should illustrate the dynamic and what happens (lateral) during a simulator experiment. The figure displays the tendency of the subjects to drive more on the right side with more LANEX on this side (in this experimental setup, without rumble strips and in this specific driving simulator mockup).

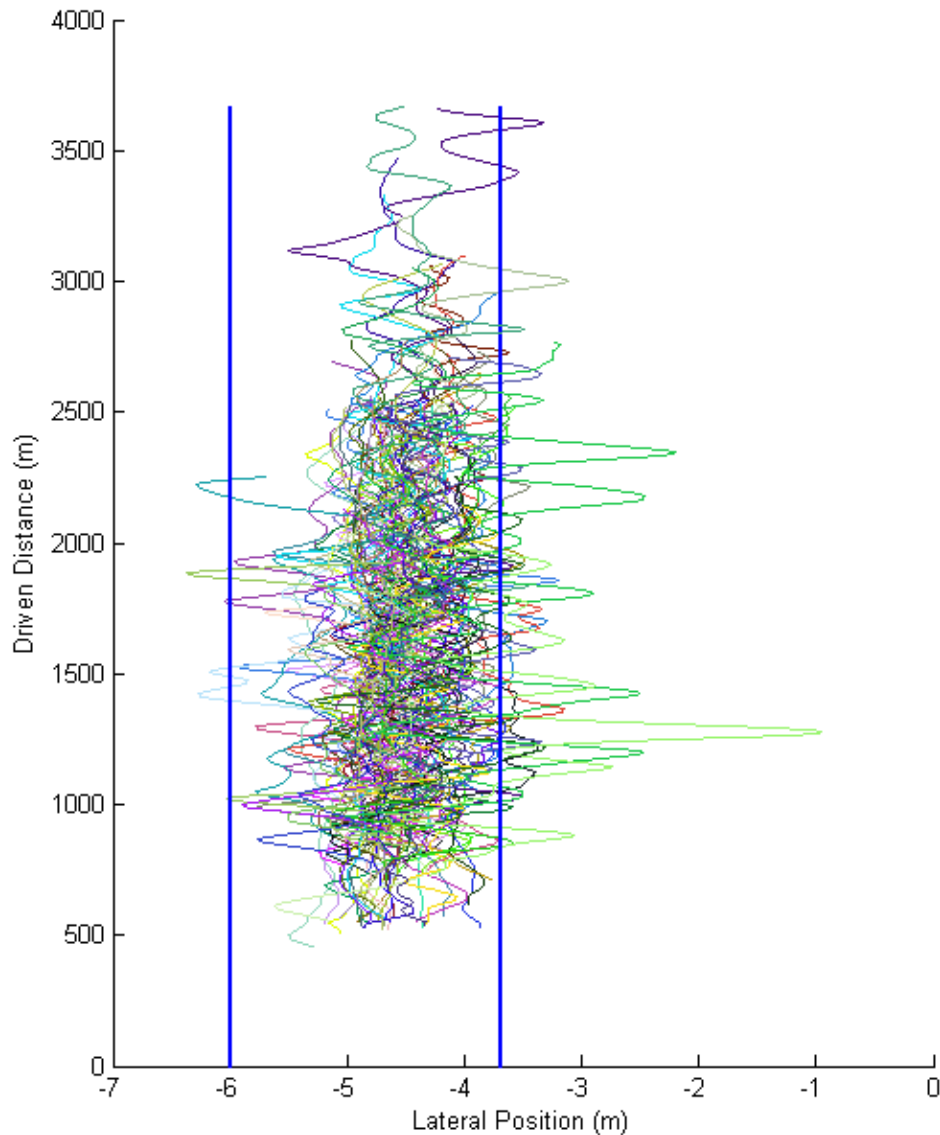


Figure 2.3.: 96 lane positions trajectories from an AAM car-following experiment with 24 people in four conditions. The trajectories are the center of gravity (COG) of the simulated vehicle. The simulation uses the right shoulder as reference (x-axis '0'). The test track had a break-down lane with 3000 mm width and 3750 mm lane width. The two blue vertical lines represent LANEX limits defined in Driver Focus-Telematics Working Group (2006, p.44) converted for the COG offset (half car width 832.5 mm). The heading/angle of the car is neglected. The lane markings widths (left 150 mm; right 300 mm) are positioned half/half on adjacent lanes, which broadens the lane by 75 mm and 150 mm on the sides

The calculation of a standard deviation typically implies that there are deviations around a mean value. The purpose of this work is to assess the performance in (short) subtasks and use these to assemble the performance of a (longer) task. Therefore, the question arises what duration is suitable for calculating SDLP and SDFH? This question also arose during the experiments of Conti et al. (2015), with assessment of single button presses of about 0.7 s.

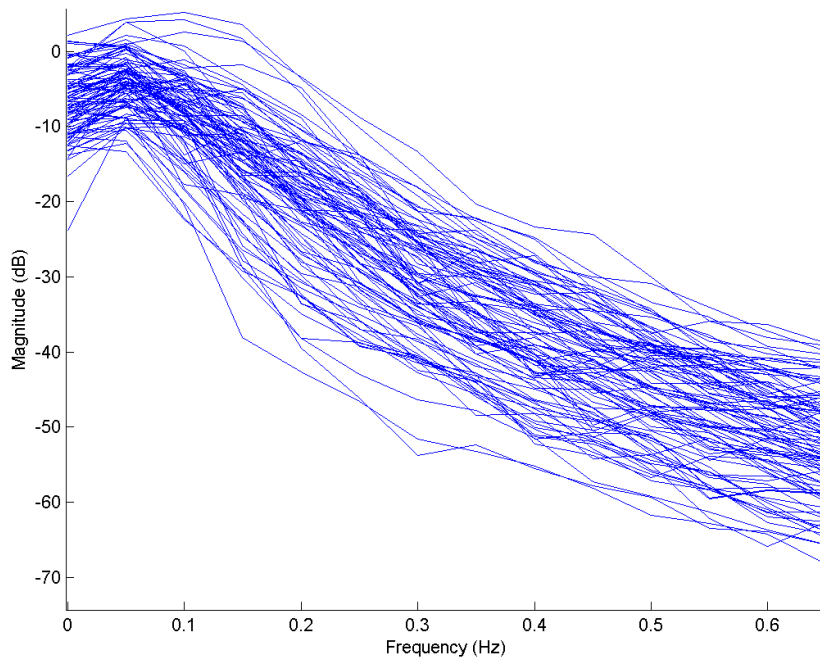


Figure 2.4.: Spectral densities of the 96 lane position trajectories from Figure 2.3 after subtracting individual mean lane positions

The data for Figure 2.3 was sampled with 60 Hz by the driving simulation. In Matlab a power spectral density estimation<sup>7</sup> was calculated for all 96 trials, after the mean value (DC offset) for each signal was subtracted. The result is displayed in Figure 2.4 for the lane position and in Figure 2.5 for the following headway to the leading vehicle. This calculation has a frequency resolution of 0.05 Hz. As can be seen in Figure 2.4, the power spectra are constant or slightly increase from 0 Hz to 0.05 Hz, and then decrease. In Figure 2.5, the spectra are constant between 0 Hz and 0.05 Hz and then demonstrate a uniform, steep decrease. Therefore, a possible recommendation for this AAM following setup and driving dynamic could be that, for calculation of SDLP, the duration should be at least  $1/0.1 \text{ Hz} = 10 \text{ s}$  and for SDFH  $1/0.05 \text{ Hz} = 20 \text{ s}$ , to capture relevant parts of the lateral and longitudinal control. The result for SDLP is similar to the findings in Östlund et al. (2005, pp. 38–41). To make meaningful comparisons for variability metrics, the durations of (sub)tasks must be equally long, or the MSDLP (Östlund et al., 2005)

<sup>7</sup>Welch’s power spectral density estimation, Hanning window, 1200 samples = 20 s, 600 samples overlap and a 1200 point DFT

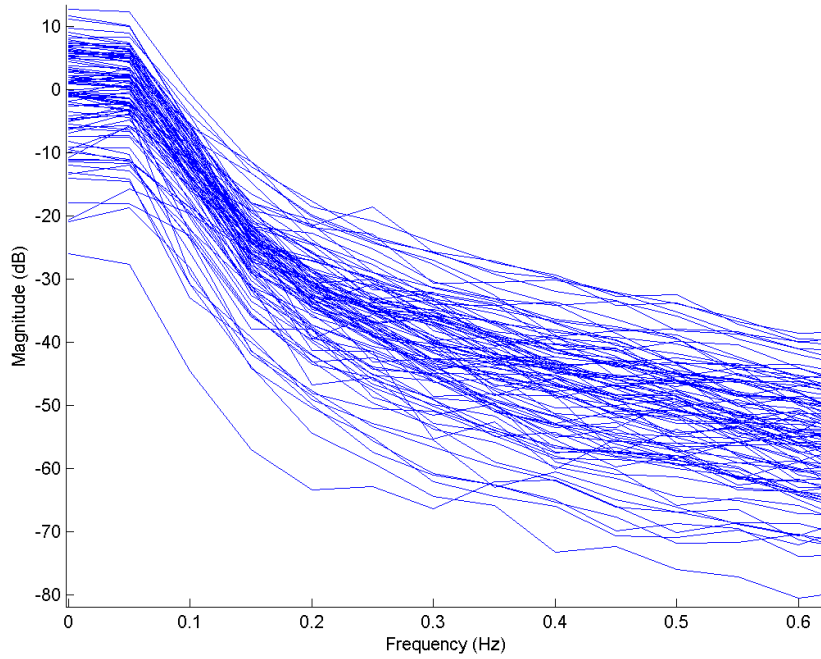


Figure 2.5.: Spectral densities of the 96 following headway time series after subtracting individual mean following headway

(high-pass filtering) should be used. Also with MSDLP one has to adhere to minimum task durations, which are reciprocal of the high-pass filter frequency (cf. Figure 2.2). The Total Task on Time (while driving) of the later-analyzed subtasks are often shorter than 10s–20s. Therefore, the classical metrics and MSDLP do not fit.

To obtain a metric for this question, the following thoughts were included: The task of the subjects is to drive straight forward and follow a leading vehicle with a constant following headway. Therefore, every deviation from driving straight forward (i.e., lateral velocity) or changing the constant headway is of interest (cf. Figure 2.6). This signal of interest can be continuously generated with a derivative, approximated by a differences quotient and further simplified to the difference between (time equidistant) sample points. In other word, the  $\Delta t$  between sample points is neglected in this step. The derivative itself can be interpreted as a kind of filter (high frequency emphasis). The Modified Lateral Position Variation (MSDLP) in Östlund et al. (2005) uses high-pass filtering as an enhancement of the SDLP, too.

$$\frac{dLP_y}{dt} \Rightarrow \frac{\Delta LP_y}{\Delta t} \Rightarrow \Delta LP_y$$

$$\frac{dFH}{dt} \Rightarrow \frac{\Delta FH}{\Delta t} \Rightarrow \Delta FH$$

These differences are rectified and summed up (integrated) over the time period of a subtask. With these metrics, a longer subtask likely gets a worse performance rating; i.e., a higher value. Therefore, these non-normalized metrics are normalized by the duration

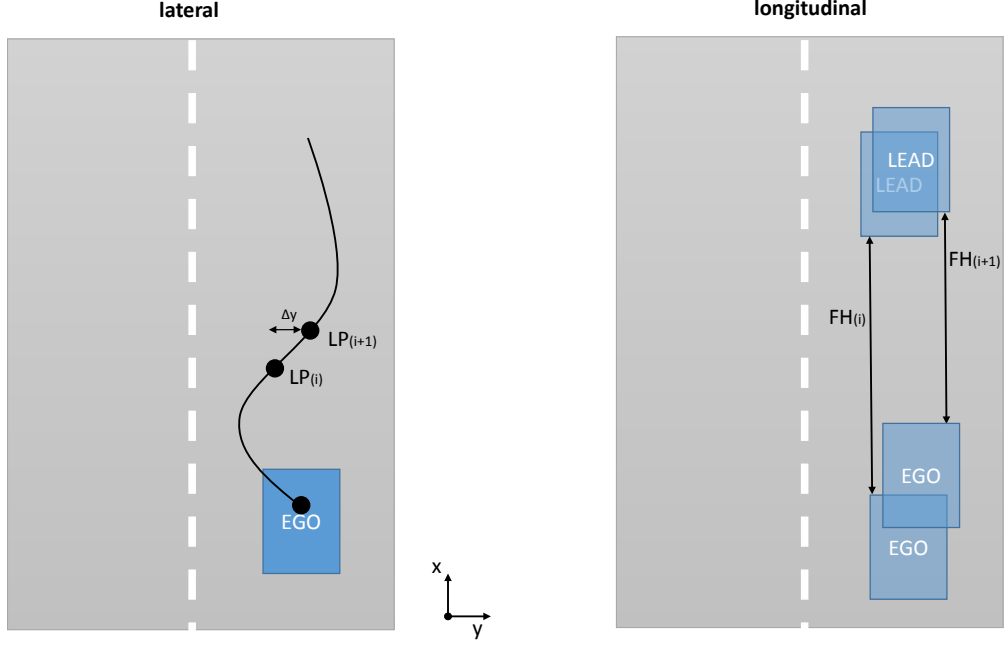


Figure 2.6.: Lateral (Lane Position; LP) and longitudinal (Following Headway; FH) metrics between ego-car and leading vehicle

of the subtask (i.e. the sum of all  $\Delta t$ 's). Because the prevailing reason of this signal is the drifting of the primary task performance away from (or steering toward) an individual mean value (strategy) it is termed in this thesis: Drift in Lane Position (DLP) or Drift of Following Headway (DFH)

$$DLP = \frac{\sum_{i=subtaskStart}^{subtaskEnd} |LP_{y(i+1)} - LP_{y(i)}|}{subtaskDuration}$$

$$DFH = \frac{\sum_{i=subtaskStart}^{subtaskEnd} |FH_{(i+1)} - FH_{(i)}|}{subtaskDuration}$$

In Matlab the metric can be simply coded, e.g.:

$$DLP = \text{sum}(\text{abs}(\text{diff}(\text{LanePosition}))) / \text{subtaskDuration}$$

This is an Average Rectified Value (ARV) calculation. The result for DLP is scaled to millimeters per second (mm/s) and for the DFH to milliseconds per second (ms/s). The metrics are easily interpretable values of the rate of change. DLP: drift of the lateral position (mm) per second. DFH: drift of the following headway (ms) per second. These metrics are also beneficial regarding the assembly of subtasks to tasks: Standard deviations of subtasks cannot be simply summed up. For DLP and DFH the 'non-normalized drift' and the durations of different subtasks are stored separately and can be summed up. Finally,

the task overall result is normalized by the overall duration: 
$$\frac{\sum_{subtasks} subtaskDrift_{nonNormalized}}{\sum_{subtasks} subtaskDurations}$$

The DLP itself is definitely not new; in Salvucci (2005) the metric is called *average absolute lateral velocity*<sup>8</sup> and described as “[...] *common in empirical studies of driver behavior* [...]”. However, in the last decade of driver distraction assessment the SDLP was the workhorse in judging lateral driving performance.

The SDLP physically depends on deviations in lane position (from a mean), while DLP examines related lateral velocities. Two other approaches to assess lateral driving performance are:

- the Time to Line Crossing (TLC) (cf. SAE J 2944, 2013; Johansson et al., 2004; Östlund et al., 2005)
- and the Mean Deviation (MDEV) in ISO 26022 (2010) (Lane Change Test).

For longitudinal performance, the analog to the (lateral) TLC is the Time To Collision (TTC) (cf. SAE J 2944, 2013; Johansson et al., 2004; Östlund et al., 2005).

When compared to TLC and MDEV, the DLP can be seen as a hybrid: The integration and normalization is more similar to the MDEV. However, similar to TLC, DLP is based on lateral velocity.

TLC assessments typically consider minimums and lack easy additive capabilities when combining subtasks. The TLC calculation also requires more geometric and dynamic data from the vehicle and the vehicle environment (or some simplifications and approximations).

The MDEV is the ARV between the lane position and a predefined reference trajectory. The authors of the LCT standard were aware of individual behaviors and proposed an ‘Adaptive MDEV’ (Annex of ISO 26022, 2010). For the Adaptive MDEV a baseline drive is performed to acquire values for the individual mean lane positions and lane change behavior. This is used to adapt the reference trajectory to the individual behavior.

The previously mentioned metrics all rely on lane position or following headway. Lane position and headway can be seen as the result (output) of the control loop of driver and vehicle (cf. Jürgensohn, 2007; Michon, 1985). The control loop is principally closed by the (foveal) visual perception of the driver. The feedback loop is potentially impaired by a dual-task setting (secondary tasks while driving) with eyes-off-road tasks. When eyes are off the road, there are strong indications that experienced drivers can obtain more additional useful cues for lane keeping from peripheral view than beginners (cf. Summala et al., 1996). The driver makes corrections (feedback loop) to the car primarily through the steering wheel, throttle and break. These correction inputs can also be used for metrics. An established metric is, e.g., the Steering Reversal Rate (cf. SAE J 2944, 2013; Östlund et al., 2005). These values are sampled before the inertia of the vehicle dynamics and are typically more agile. Nevertheless, for this thesis, the resulting ‘Ground Truth’ lane position and following headway are used for metrics. It is recognized that sometimes increased steering activities do not ‘punch through’ to road metrics or are hard to interpret: *“The increased steering activity did however not result in any change in lateral position variation (st\_lp) or any of the time to line crossing measures (e.g. mn\_tlc).”*

---

<sup>8</sup>If one assumes that the arithmetic mean (and not the harmonic mean) is used, *average absolute lateral velocity* and DLP are equal for equidistant sampled data.

(Östlund et al., 2004, p. 165). This phenomenon is also noted in Johansson et al. (2004, p. 20): “[...] increased steering activity can be associated with both increased and reduced lane keeping performance”

An implicit assumption for DLP and DFH is that a characteristic detrimental influence of a subtask onto these metrics can be observed during the time of a subtask (and not after). So, the influence can be used to rank and classify subtasks. Furthermore, possible interactions between subtasks are neglected, e.g., the aftereffect of the last subtask onto the current subtask.

SDLP and DLP metrics would not detect if a subject drives perfectly straight forward, but outside of the lane. Completely disregarding the task instruction and experimental setting is untypical and should be (hopefully) detected by the examiner or data analyst.

In a plausibility check, the data from Krause et al. (2015a) were used to calculate DLP and DFH values. These were correlated (Pearson correlation,  $N = 24$  subjects) to the established SDLP and SDFH for four experimental conditions (baseline, and radio tuning on three devices). The correlations between DLP and SDLP were .623, .595, .906, .857; and DFH to SDFH .869, .823, .898, .901. These are medium to high positive correlations.

Another plausibility check was performed in the subtask database when the subtasks were classified into ascending order based on the DLP or DFH metric. In this categorization, longer delays of 4 s and 8 s (i.e. subtasks that consist of waiting) have a better primary task performance. Touchscreen subtasks typically have an impairment in the primary driving task (higher DLP or DFH); rotary knob interactions typically can be found in-between the delays and touchscreen subtasks. The DLP seems to be more sensitive than the DFH.

The experimental condition of **tuning a hardware radio** from Krause et al. (2015a) was used to calculate preliminary DLP and DFH criteria for this thesis and the experimental setup (i.e. the AAM following task at this specific driving simulator):

- DLP  $M = 85.8$  mm/s
- DFH  $M = 61.8$  ms/s

For **baseline** driving performance (without radio tuning):

- DLP  $M = 39.4$  mm/s
- DFH  $M = 40.6$  ms/s

The driving performance in this thesis is judged relative to the baseline performance. This approach is also used for the DRT reaction times in this thesis. In all of these metrics, a higher magnitude stands for a lower performance:

$$\text{relativeDeterioration} = \frac{\text{metric}_{\text{withSecondaryTask}} - \text{metric}_{\text{withoutSecondaryTask}}}{\text{metric}_{\text{withoutSecondaryTask}}} * 100\%$$

Therefore, the reference deteriorations of the radio tuning compared to baseline driving are:

- DLP 117.7%
- DFH 52.2%

## 2.4. Task Analysis and Modeling

Different methods would be possible to calculate the prediction error percentage of a task analysis. In this thesis, the most common procedure is used (cf. Pettitt, 2006; Harvey and Stanton, 2013): The absolute difference between prediction and actual measurement is referenced to the actual measurement. For example, if the prediction is 10 s and the actual measurement is 5 s, the error is +100%. An acceptable prediction error could be  $\pm 20\%$  (cf. Pettitt, 2006; Harvey and Stanton, 2013). Harvey and Stanton (2013) mentioned that it is harder to predict higher percentiles in right-skewed distributions and propose a relaxed criterion of 40% for the 90<sup>th</sup> percentile.

An extensive model useful in understanding human perception, cognition and reaction is the Model Human Processor (MHP) (Card, 1981; Card et al., 1983, 1986). The model integrates empirical data from literature, other models and ‘laws’ (e.g., Working Memory, Fitts’s Law, Power Law of Practice, Hick’s Law, etc.). The model aids the understanding of the limits of human performance and the time is needed for an action. For different capabilities, a typical, nominal value is presented to model a ‘Middleman’; often also a range is specified, to model best- and worst-case capabilities (Fastman, Slowman).

GOMS models “[...] hypothesize that the user’s cognitive structure consists of four components: a set of Goals, a set of Operators, a set of Methods for achieving the goals, and a set of Selection rules for choosing among a goal’s competing methods.” (Card et al., 1980a). In Card et al. (1980a) an example is given to model a text-editing task. A goal can be composed of ‘unit tasks’ to reach ‘subgoals’. The smallest unit to carry out activities is the ‘operator’ (“Operators are elementary motor or information-processing acts,[...]”). “A method describes a procedure for accomplishing a goal.” Methods can have conditional statements, e.g., to repeat an operator or operator sequence. Some goals (and subgoals) can be accomplished with different methods Selection rules decide which method is used. The modeling was tested in different time domains i.e. with fine and coarse modeling (‘grain of analysis’): “[...], the rather surprising answer is that accuracy was essentially independent of the grain.” Card et al. (1980a). The GOMS modeling and notation has features and the appearance of a programming language and can be used to predict Total Task on Time.

The Keystroke-Level Model (KLM) is a simplified model used to predict task times (Card et al., 1980b). It uses four motor operators (keystroking, pointing, homing and drawing), one mental operator and the response time of the system. These six components are additively summed up. A keystroke is the most obvious operator. ‘Pointing’ is related to input devices (e.g., a mouse); ‘drawing’ is another operation time using a mouse. ‘Homing’ is the time needed to switch between input devices. The mental operator is inserted, e.g., when the user needs a short time to plan or prepare an action. Several heuristics are provided concerning how a mental operator should be inserted in a KLM. The need and intention of the KLM is to “[...] be quick and easy to use, if it is to be useful during the design of interactive systems.” (Card et al., 1980b)

GOMS/KLM modeling assumes a highly trained operator who works on one single task without any errors or problems. Both methods have been extensively adapted and modi-



fied. Nevertheless, their usage is most often academic. An older and established modeling technique from planning production systems is the Methods-Time Measurement (MTM) (Maynard et al., 1948). MTM splits and codes human movements into small parts of activities. Thus, there are similarities to the KLM approach. While the main focus of MTM is movement times, KLM also has an explicit reference to cognitive activities (mental operator). In MTM, the times are recorded in Time Measurement Units (TMU). One TMU is  $1\text{ h}/10^5 = 0.036\text{ s}$ . The handling times are presented in tables and are further specified regarding the movement length and complexity. The method is applicable to planning work places regarding cycle time, even before they are built in real life. Therefore, MTM is part of computer systems for factory planning (e.g., Siemens Product Lifecycle Management Jack Task Analysis Toolkit).

This led to the remarkable situation that mechanical engineers are taught about MTM and use it for producing work places while human factors engineers learn MHP/KLM, which results in increasing numbers of academic modifications and studies.

The standard SAE J2365 (2002) uses an approach based on MTM and KLM (cf. Elwart et al. (2015)) to model the static Total Task Time when operating navigation systems. It includes age factors depending on the age group (1.4, 1.7 and 2.2). Therefore, e.g., the elderly (55–60 years) should need 1.7 more time than the young (18–30 years). The standard SAE J2365 (2002) is connected to SAE J2364 (2004), which specifies the ‘15-seconds rule’. This standard proposed that a task is acceptable for use while driving if it can be finished within 15 s when the car is standing still. Baumann et al. (2004) demonstrated that this rule cannot detect problematic tasks, for example reading dynamic text messages, while the occlusion method is able to spot such problems.

Schneegaß et al. (2011) also adopted KLM modeling for more general automobile interfaces and validated it with a prediction error of approximately 20% in time on task.

Standard GOMS assumes single-task operation. Therefore, Urbas et al. (2008); Leuchter (2009) enhanced it to a multitask GOMS (named mtGOMS or MT-GOMS) to model the operation of a secondary task while driving. The approach uses a resource profile of the minimal cognitive, visual, auditory and manual effort to accomplish the primary driving task. These profiles were derived empirically in a driving simulator. To model the secondary task, GOMS is enhanced with ‘checkpoints’, where the secondary task can be interrupted. For operators and methods, it must be specified which resources are needed (motor, visual, audio, cognitive). The methods of the secondary task can be declared (un)interruptible and a resume-method can be defined. Then, a scheduling algorithm tries to arrange the resource profile (primary task) and the MT-GOMS description (secondary task) and obtains the Total Time on Task while driving.

Pettitt (2006) extends the KLM with three assumptions to predict occlusion metrics: When the occlusion shutter<sup>9</sup> is open, the test subject can operate the task normally. The operator can continue working with closed shutters, except if s/he needs new visual information. An operation during closed shutter can only start if it does not need visual information. In a study, the Total Task Times while standing still were predicted with

---

<sup>9</sup>His occlusion protocol used a 1.5 s shutter open / 2 s shutter closed timing

conventional KLM and measured in an experiment for eleven tasks on two systems (A and B). The average KLM accuracy for system A was 13.5% and for system B, 15.4%. In a retrospective check, the extended KLM was used for two tasks on two systems (four tasks) to model the occlusion metrics TSOT and R-ratio (TSOT/TTT). All errors were below 20%. The average error for TSOT was 7.2% and for R, 8.6%. In an evaluation study, three tasks were tested on two devices (six tasks). For one task, the 20% error was exceeded (TTT and TSOT). The average errors for TTT (10.6%), TSOT (13.5%) and R (7.1%) were below the 20% criterion. The correlation for TTT ( $r = 0.98$ ) and TSOT ( $r = 0.93$ ) was high. In a reliability study, an external expert provided a prediction for four tasks. The average error in TSOT was 23.7% and for R, 12.8%.

While recognizing the work provided and contributions achieved, the predictions offer estimations for the mean value. With new guidelines (e.g., NHTSA, 2014) the importance of percentile values has been increased, which are not estimated by the above method.

In Kang et al. (2013) Pettitt's method was modified for two groups (young and middle-aged subjects) based on adapted and interpolated operator times from other documents (e.g., SAE J2365, 2002). In the evaluation, seven tasks of three trials each were tested. A regression analysis demonstrated a high coefficient of determination for the two age groups between prediction and measurement ( $R^2 = 0.88$ ,  $R^2 = 0.92$ ). The analysis revealed an overestimation of the prediction of 6.82 s (young) and 3.57 s (middle-aged) for the occlusion task completion time.

Elwart et al. (2015) provides an extensive database of interaction prototypes (e.g., flick, scroll, press button) on a level similar to MTM for occlusion task times, when operating a touchscreen and a hardware knob. The report also raises several questions regarding the assumptions of Pettitt (interactions when occlusion glasses are closed). *“[The] analysis revealed the mean element time for middle-aged subjects (45-55) was only about 16% longer than young (25-35) subjects, whereas the mean task time was 44% greater, primarily because there were 32% more occurrences of elements to complete tasks.”* Elwart et al. (2015).

Purucker et al. (2017) used KLM modeling to predict Total Eyes-Off-Road Times (TEORT) with four operator elements (keystroke, search in list known content, search in list unknown content, rotate knob 180° anti-clockwise). A multiple log-linear regression model led to a parameter for every operator that connects the time on task (KLM) to the TEORT. An additional parameter should account for age and allows calculating distributions (e.g. of a subject group) based on the individual age. The sparse age distribution used to estimate the age parameter, can be found in Purucker et al. (2014, Figure 3). In a validation study, the method revealed visually promising results (box plots), while the mathematical correlation between prediction and measurement was medium ( $r = 0.58$ ;  $R^2 = 0.34$ ).

Jorritsma et al. (2015) used KLM, GOMS and CogTool to model three tasks on three interfaces and compared them to empirical data. The results indicate *“[...] that KLM, GOMS and CogTool are not reliable tools on which to base a decision between multiple interface alternatives [...]”* and *“[...] raises questions about the validity of these cognitive*

modeling tools in interface design practice [...]”.

A study that extensively relied on (interpolated) age parameters was the aforementioned Kang et al. (2013); based on the also previously mentioned SAE J2365 (2002). They observed that when tasks grow longer and more complex, age seems important and leads to longer occlusion task times for older subjects (see Figure 2.7). The task time for the young group is typically equal or shorter than the middle-aged group. The figure demonstrates that the differences become larger when the task time increases, from right to left (B=Block Task Type, T=Task Trial).

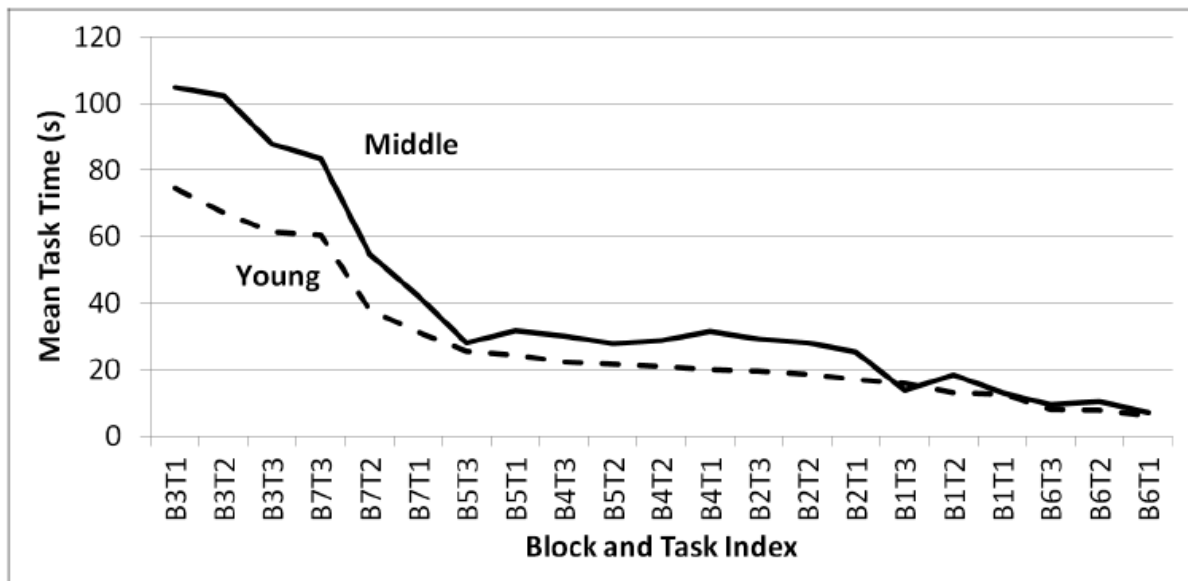


Figure 2.7.: Distribution of mean occlusion task times of two age groups from Kang et al. (2013, p. 20). For longer tasks, the differences between Young and Middle increase.

For a single button actuation (with about 700 ms) while driving, Conti et al. (2015) found no effect derived from age. These would be indications that the task length and the age of test subjects are perhaps interacting. On the one hand, this would make it challenging to model long tasks. On the other hand, when a modeling technique is intended for short tasks anyway, maybe the influence of age is not essential.

Harvey and Stanton (2013) used a technique commonly known from project resource management (the critical path method), to predict static Total Task Times of 14 in-vehicle tasks. Critical path analysis (CPA) is also part of CPM-GOMS (John and Gray, 1995), a GOMS derivative to arrange parallel perceptual, cognitive and motor operators. The core of modeling in Harvey and Stanton (2013) is similar to KLM, while the CPA is used to handle parallel operations (visual, manual, cognitive). This approach is further extended, not only to predict a median performance (middleperson); also a best case (10<sup>th</sup> percentile, fastperson) and a worst case (90<sup>th</sup> percentile, slowperson) are calculated. The comparison to experimental data shows an average error of 8.43% for the middleperson, 12.89% for the fastperson and 20.01% for the slowperson. A contribution of the study is the attempt to extend the KLM beyond the typical predicted mean/median; similar to the MHP

‘Fastman’ and ‘Slowman’ approach from Card (1981) on the capability level. An implicit assumption of the modeling used is that a fastperson is in all operator times a fastperson and a slowperson is comprehensively a slowperson in every operation and decision; which can be doubted. For example, in anthropometry it is known that to predict the 95th percentile human height, the simple sum of the 95<sup>th</sup> upper body and the 95<sup>th</sup> lower body is insufficient. This is also known by the authors (Harvey, 2011, p .171). However, for an extension of the method to predict Total Task Times while driving, this knowledge is not used when combining operators. Otherwise, it is confusingly used to justify why fast-/middle-/slowperson will be modeled with the same glance behavior. The enhancement to predict dual tasks is implemented in a ‘Dual-Task CPA Calculator’. The tool separates visual operations and non-visual operations. The visual operations are grouped according to a supposed glance behavior. The glance behavior assumes short glances to the IVIS (430 ms) followed by road glances of 687ms, based on experimental data. In an evaluation experiment (14 tasks), the predictions of Total Task Times while driving were severely inaccurate (fastperson: 87.55%; middleperson: 56.1%; slowperson: 44.03%). After reanalysis of data, a new glance behavior was introduced: Two glances to the IVIS (430 ms) are connected with a ‘shared glance’ of 360 ms (between IVIS/road). Therefore, effectively, a glance to the IVIS can be 1220 ms long. An easier explanation, instead of the ‘shared glance’ construct, are perhaps measurement artifacts in the former experimental data. With this post-hoc adjustment, the accuracy was improved (fastperson: 22.29%; middleperson: 16.42%; slowperson: 25.33%). Despite the continuous emphasis of the visual system, no glance metrics were used for evaluation, instead the Total Task Times are assessed.

Kurokawa (1990) implemented a Pascal program for an Apple II Macintosh: The Instrument Panel (IP) Analyzer (IPanalyzer). The program estimates seven metrics:

- Total Task Time while driving
- Hand-off-wheel time
- Total Glance Time to IP
- Number of Glances to IP
- Average Single Glance Duration to IP
- Average Single Glance Duration to the road
- Average eye transition time

Furthermore, four ‘merit’ ratings are calculated to rate the manual demand, the visual demand, a combined demand and a combined demand which is adjusted by the frequency of use of the estimated task. The estimated values can be adjusted by selecting and specifying: age, gender, concurrent driving workload, location of the instrument and label characteristics (color, size, luminance, abbreviations).

The tool can be used in three modes. In the empirical mode, a task from a database can be selected. The database holds approximately 50–60 tasks. The tasks are diverse and range from checking a speedometer and adjusting a mirror to more complex tasks such as tuning a radio. The data are empirical, but can be also estimates for some metrics

(“Therefore, assumptions were made, and best estimates were used when the values were not available.” Kurokawa (1990, p. 207)).

In a second mode, the user can enter his/her own estimates when a task is missing. Three values have to be estimated: Average Single Glance Duration (see Figure 2.8), Number of Glances and Hand-off-Wheel Time.

Representative Average IP Glance Times	
<input type="radio"/>	0.00 Driver does not need to glance at task.
<input type="radio"/>	0.60 Driver reads 2-digit speedometer.
<input type="radio"/>	0.70
<input type="radio"/>	0.80 Driver adjusts power mirror, glancing between the mirror and the controls.
<input type="radio"/>	0.90
<input checked="" type="radio"/>	1.00
<input checked="" type="radio"/>	1.10 Driver selects 1 of 5 pushbuttons, then retrieves digitally displayed information.
<input type="radio"/>	1.20
<input type="radio"/>	1.30 Driver makes several discrete activations on a bar LED display.
<input type="radio"/>	1.40 Driver enters a 7-digit telephone number.
<input type="radio"/>	1.50
<input type="radio"/>	1.65 Driver determines, from a navigator display, the name of street to turn onto to get to destination.

Figure 2.8.: Task estimate dialog box from Kurokawa (1990, p. 284; Figure 89)

In the task analysis mode, the user makes a task analysis with 15 behavioral element categories (see Figure 2.9 and Figure 2.10) and enters the amount of the ‘task elements’. The parameter, ‘Number of instruments in the dashboard’, can be used to adjust for ‘macro clutter’. The 15 categories are calculated from the empirical database and literature. The thesis does not include a comprehensive validation experiment, but the task analysis mode was tested to model four tasks from literature. While three tasks, which used different behavioral element categories showed promising results, modeling the entering of a 7-digit telephone number (repeated the same element) was severely inaccurate, by a factor of 4.

The extensive data resource and computer science work implemented (approximately three decades before this thesis) is impressive. The 15 behavioral task elements in the task analysis mode can be perceived as a KLM.

The Federal Highway Administration (FHWA) started a project in 1996 about IVIS design (Hankey et al., 2001a,b). Within the project, a simplified driver behavioral model with five resource components was proposed (visual input, auditory input, supplemental information processing, manual output, speech output). This model is incorporated into a behavioral prototype software called IVIS DEMAnD (In-Vehicle Information System Design Evaluation and Model of Attention Demand). DEMAnD primarily moves the idea of Kurokawa (1990) to Windows software and the database includes 198 tasks. The data

<b>Visual detection &amp; monitoring requiring no manual demand</b>	<input type="checkbox"/>
<b>Check-reading (Number or label) requiring no manual demand</b>	<input type="checkbox"/>
<b>Simple locate &amp; reach no selection involved</b>	<input type="checkbox"/>
<b>Manual delay</b>	<input type="text" value="0"/> seconds
<b>Selective activation (1 of N choices)</b>	
<b>Random / Unfamiliar Labels: N =</b>	<input type="text" value="1"/> <input type="checkbox"/>
<b>Sequential / Familiar Labels: N =</b>	<input type="text" value="1"/> <input type="checkbox"/>
<b>Gross Adjustment with visual feedback... and continuous input</b>	<input type="checkbox"/>
<b>and discrete input</b>	<input type="checkbox"/>
<b>Gross Adjustment with auditory feedback... and continuous input</b>	<input type="checkbox"/>
<input type="button" value="Next"/> <input type="button" value="Exit"/>	

Figure 2.9.: Task analytic procedure dialog box from Kurokawa (1990, p. 298; Figure 93)

<b>Fine adjustment with visual feedback... and continuous input</b>	<input checked="" type="checkbox"/>
<b>and discrete input</b>	<input type="checkbox"/>
<b>Fine adjustment with tactile feedback... and continuous input</b>	<input type="checkbox"/>
<b>and discrete input</b>	<input type="checkbox"/>
<b>Simple interpretation (visual)</b>	<input type="checkbox"/>
<b>Complex interpretation (visual) and decision making</b>	<input type="checkbox"/>
<b>Number of instruments in the dashboard</b>	<input type="text" value="1"/>
<input type="button" value="Previous"/> <input type="button" value="Exit"/>	

Figure 2.10.: Task analytic procedure dialog box from Kurokawa (1990, p. 299; Figure 94)

comes from literature, practitioners and four experiments. DEMAnD uses three levels for internal organization: system, task and subtask. On the system/vehicle level, for example,

the vehicle dimension can be specified. Within a system, different tasks can be created, which can consist of subtasks. The task-level offers parameters which can be modified (*'modifiers'*), e.g., age, traffic density and road complexity. On the subtask level, e.g., the character height, contrast, display density and anthropometric can be modified. Additional subtasks can be added by the user with *'interpolation screens'* to enter estimates or empirical data for up to 15 metrics. Subtasks can be also programmed with a text editor into config files. The common metrics are Average Single Glance Duration, Number of Glances and Task Time; the rest of the metrics seems slightly uncommon (e.g., *'Subjective Supplemental Information Processing Time-Sharing Demand Rating'*). The tool calculates a proposed overall Figure Of Demand (FOD) rating. The final statement emphasizes the prototypical proof-of-concept state (Hankey et al., 2001a, p. 57) and recommends a validation for further research (Hankey et al., 2001a, p. 56). Therefore, no validation data are provided.

GOMS, KLM and their derivatives are manual (paper and pen) methods, but clearly can be supported with computer programs, e.g., to generate, edit and calculate KLM lists. Regarding the MHP there is a strong background of experimental psychology. A slightly different approach are cognitive architectures. They are heavily based on computer programming. The scientific background is again drawn from knowledge of the human brain and behavior. The aim is to refine computer models so a situation can be entered into the computer model and the model behaves and decides like as a human would. Previously measured experimental results are attempted to be analyzed and explained with cognitive architectures or heuristic models are developed, e.g., splitting situation handling into declarative rules and procedural rules and placing them into different subsystems such as perception and locomotor systems. Afterward, this analysis must be mapped to architecturally specific notations (programming).

The methods used in this thesis are in the (academic) tradition of MHP, GOMS and KLM. Nevertheless, the calculations behind the selected methods are so lengthy that paper and pen are not suitable and a supporting online tool is implemented. On the other hand, the approach is not so highly sophisticated that the term *cognitive architecture* would fit. Cognitive architectures are therefore out of the scope of the thesis and only briefly mentioned. An overview of some cognitive architectures and applications to real-world cases can be found in Leuchter (2009). Also Mavor et al. (1998) contains an overview of cognitive architectures and aims at military purposes. The perhaps most frequently mentioned cognitive architecture in the field of human factors is *Adaptive Control of Thought-Rational* (ACT-R) (Anderson and Lebiere, 1998).

The programming for cognitive architectures can be cumbersome. Therefore, John et al. (2004a) implemented a tool that captures the interactions of a person with a (mockup) user interface and automatically generates ACT-R code for the cognitive architecture via an intermediate (language) step with ACT-Simple/KLM. This enables the curious situation in which a developer can interact with an interface (under evaluation) and the computer predicts and models how long his/her interaction should have taken. Perhaps a user test with  $N = 1$  and a stopwatch can provide similar results. If additional knowledge of this single test subject is available (e.g., from a reference task) the result may be judged further. For example, this single user needed the average time in the reference task, has an average error-rate tradeoff and often uses long glances. The cognitive architecture approach seems valueless if the evaluated interface is for everyday interactions (e.g., an

office program or web page) that every developer can access with low costs. For special purposes that would have high access costs for testing (e.g., flight cockpits), it might be valuable however. In John et al. (2004b), the name *CogTool* is mentioned for the tool and the first steps of integration with a driver model are presented. The CogTool is open source and further modified to the Human Efficiency Evaluator applied to an aeronautical example in Feuerstack et al. (2015).

In ACT-R, Salvucci (2006) implemented a driver model and evaluated it regarding lateral/longitudinal vehicle control and gaze location. This approach was further evolved into a tool: Distract-R (Salvucci, 2005, 2009). Distract-R encapsulates a subset of the cognitive architecture, ACT-R, and offers a graphical user interface: “[...] *intended for any designer or engineer who is part of the in-vehicle design process, particularly those (in the majority) with no prior experience in cognitive modeling.*” Salvucci (2005). In a WYSIWYG-editor, the modeler draws the intended interface then s/he carries out the tasks that should be assessed (*‘Modeling by Demonstration’*). Afterwards, the modeled driver can be parametrized by options in the user interface. Salvucci (2009) states: “*The theory behind how individual differences map to cognitive models and architectures is currently very incomplete*”. Nevertheless, Distract-R offers three parameters: driver age, steering aggressiveness and a stability factor (desired stability; driver’s safety tolerance). The age can be young (20–30 years) or old (60–70 years). When old is selected, the cognitive processing time is scaled by 13%, which leads to non-trivial effects within the cognitive architecture (Salvucci, 2009). In a configuration panel, the situation has to be specified (speed, straight road, curved road, leading vehicle, leading vehicle speed, random breaking). A result panel displays the predicted Total Task on Time while driving and lateral vehicle control performance. An internal player (driving simulator) allows viewing how the simulated driver steers the vehicle and operates the interface under evaluation. Concerning the predicted task time while driving (seconds), a study in Salvucci (2005) with four short tasks (< 10 s) reports an accuracy of  $R^2 > .99$ ,  $RMSE = .53$  and in Salvucci (2009) with eight tasks of various durations (about 5–160 s):  $R = .988$ ,  $RMSE = 22.4$ . There had been also plans to give Distract-R the ability to retrieve visual features (salience) from a vehicle interface (Lee et al., 2012).

An earlier approach to assess (alphanumeric) workplace displays automatically, can be found in Tullis (1984). The thesis derives and defines six metrics (Overall Density, Local Density, Number of Groups, Size of Groups, Number of Items, Item Uncertainty). These metrics can be calculated with a given C program to assess an interface (display page). In an experiment, two regression equations are identified to predict search times based on four of the metrics and to predict subjective ratings based on all six metrics. In an evaluation experiment, the approach is validated with a correlation of  $r = .800$  for the search time and  $r = .799$  for the subjective ratings.

A scheme originating in the aeronautical domain is the SEEV model for glance allocation. SEEV stands for Salience, Effort, Expectancy and Value (Wickens et al., 2001). Salience is the conspicuity of an area/signal, effort describes the physical effort needed to switch to that area (attention movement, eye/head movement), expectancy is the frequency of events (bandwidth) in an AOI, value is the product of the relevance of an AOI for a task with the task priority (relevance x priority). These parameters are estimated



by experts and result in the percentage dwell time for each AOI. In Horrey et al. (2006), the model has been also transferred and tested in the automobile domain (driving with an in-vehicle task). The SEEV model is for the prediction of the final steady-state percentages of dwell times. It has been further extended to the NSEEV in Steelman et al. (2011), where N stands for noticing. The NSEEV model, with its dynamic capabilities should enable a prediction, if events in AOIs are missed. The SEEV model is also dynamically incorporated into the often-mentioned Man-Machine Integration Design and Analysis System (MIDAS) v5 from NASA to obtain dynamic visual scanning behavior (Gore et al., 2009; Gore, 2011).

## 2.5. Own Previous Work and Motivation

From personal communication, it became clear that human factors practitioners have coarse rules of thumb in mind when first inspecting an interface for use while driving. For example, actuating one button needs approximately one second of task time; one glance needs approximately one second of glance time. These heuristics can be seen as imprecise and simplified KLMs. The review in Section 2.4 implied that the modeling techniques in driver distraction assessment are typically academic. The work of Purucker et al. (2014, 2017) with a researcher from Hyundai and Krause et al. (2015b) with a consortium of OEMs demonstrate that there is an interest for refined and practical methods regarding the common industrial driver distraction tests.

This thesis is based on previous ideas and findings reported in Krause et al. (2015b). While the database of Krause et al. (2015b) is confidential (industry project), the Car Connectivity Consortium graciously allowed that this thesis could run some comparisons. This opportunity is used in Section 3.6.1 for some subtasks and in Section 4.6.5 for one complete task.

Krause et al. (2015b) focused on the glance and occlusion criteria in Driver Focus-Telematics Working Group (2006). Therefore, the subject selection was also according to the AAM definition (45–65 years old). The project included three experiments. In the first experiment, the interactions of people with a touchscreen were measured; in a second experiment with a rotary knob. Based on these data, the outcome of a third evaluation experiment was predicted before the experiment was conducted. The metrics included the Total Shutter Open Time, the Total Glance Time to the secondary task, Single Glance Durations (task-related) and the Number of Glances (task-related). The result was promising and demonstrated the feasibility of the methods selected.

The background of the project was the idea to analyze and predict tasks on a subtask level. A subtask could be, for example, the input of a number with ten digits or the selection of a name from a list. In the project, the idea was used also that the System Response Time could be a subtask. The cumbersome modeling with, e.g., KLM operators is lifted one level up to coarser subtask operators. These operators are recorded and stored from real test subjects in a driving simulator. This solves the multitasking requirement (driving and secondary task) of the modeling approach. Most models simplify the times for an operator and use one average value (e.g., mental operator =  $x$  seconds). In this project, the subtask value of each single test subject was stored in a database. From this subtask database, the specific values for each person can later be composed to a task. This was called a ‘virtual experiment’ or the ‘Berlin-Munich-Method’. In other words, the model does not calculate a single outcome, it calculates 24 models in parallel (for each subject). This results in 24 values (i.e. a distribution). This distribution can be used to derive, e.g., the 85<sup>th</sup> percentile.

Problems arose during the project when the Number of Glances and Single Glance Durations were calculated based on the subtasks. The problem is posed by glances which are split by subtask boundaries. This artificial splitting increases the Number of Glances while incorrectly decreasing the Single Glance Duration. Each data recording must have a start and end. Therefore, this problem is inevitable and relevant for all eye-tracking

data. When the amount of (subtask) boundaries is increased, the problem also increases. It is interesting that this issue is not further addressed in guidelines and standards; while it is known: “If a test participant eye glance was in progress at the start of data collection, only use the segment after the start of data collection.” (NHTSA, 2014, p. 15). “If a test participant eye glance was in progress at the end of data collection, only the portion that occurred before the end of data collection is used.” (NHTSA, 2014, p. 14).

If a task needs three glances each of 1 s and a short part of a glance (e.g., 0.2 s) accidentally slips in at the start or end of the (sub)task, it is likely that an eye-tracking system or analysis script would calculate a Single Glance Duration of  $3.2s/4 = 0.8s$  (see Figure 2.11). The countermeasure in Krause et al. (2015b) was to allow and use fractional Number of Glances. E.g, when one and a half glance is within the boundaries of a subtask, the Number of Glances is 1.5. This approach is also used for this thesis.

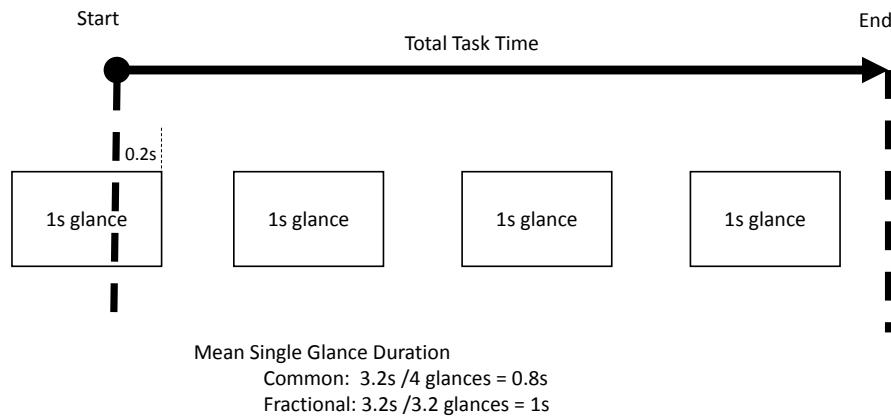


Figure 2.11.: Split glance problem (cf. Krause et al., 2015b)

The main differences and enhancements of this thesis to Krause et al. (2015b) are explained below. Krause et al. (2015b) recorded the touchscreen and rotary knob interactions from different people in two experiments. Therefore, these data (subtasks touchscreen and subtasks rotary knob) have no internal connection. For instance, it is not possible to model a hybrid task (touchscreen/rotary knob). The touchscreen and rotary knob data collected for this thesis are **intra-individually connected**.

To enable the collection of touchscreen and rotary knob data in one experiment, a different approach is used. Krause et al. (2015b) used recordings of complete task interactions and manually extracted subtasks afterward with the help of several student assistants (video coding). For this thesis, subtasks are implemented in an application that **automatically sends appropriate triggers/markers** to connected systems and therefore supersedes manual coding. This also slightly changes the origin of the subtasks. While Krause et al. (2015b) used ‘real’ tasks. This thesis uses ‘clean’, laboratory subtasks (GUI widgets).

In Krause et al. (2015b), four metrics were incorporated (from eye-tracking and occlusion). This is extended in this thesis to **13 metrics** from four methods (driving, eye-tracking, occlusion, DRT):

- Total Time on Task (TTT static; non-driving)
- Total Time on Task while driving
- Glance – Total Glance Time (task related)
- Glance – Single Glance Duration (task related)
- Glance – Number of Glances (task related)
- Glance – Total Eyes-Off-Road Time
- Glance – Single Glance Duration (eyes-off-road)
- Glance – Number of Glances (eyes-off-road)
- Occlusion – Total Shutter Open Time (TSOT)
- Occlusion – R-Metric (TSOT/TTT)
- Tactile Detection Response Task – Deterioration in Reaction Time (%)
- Driving – Deterioration in Lateral Drift (%)
- Driving – Deterioration in Longitudinal Drift of Headway (%)

The data from different measurement methods also have the advantage of being intra-individually paired.

While Krause et al. (2015b) used the AAM subject sampling (45–65 years old), the persons in this thesis are approximately 20–30 years old. The comparison of some subtasks to Krause et al. (2015b) should offer some insight regarding the implications. The addition of younger test subjects is also one of the major differences of NHTSA (2014) to Driver Focus-Telematics Working Group (2006).

As proposed in Krause et al. (2015b), the result can be used in two ways: The prediction model to compose a task from subtasks is one output, but the subtasks database itself is also a valuable outcome; e.g., to show developers the effects of different subtasks or enable researchers to check their own results for plausibility.

Similar to Krause et al. (2015b), the prediction is not intended to replace empirical testing methods. The motivation is to support the development process and reduce the likelihood that a clearly unsuitable application makes its way into a time and cost consuming driving simulator test. Most tests need a prototype and are therefore late in the development process. For the European method (comparing interface alternatives to choose the best one), the model and data can hopefully aid in ascertaining (theoretically) improved candidates for empirical testing.

In addition to these primary goals (prediction model and database) the thesis has a secondary focus on the influence of delays (SRTs) on metrics; especially on Single Glance Durations. Therefore, System Response Times form a considerable part of the subtask database of the prediction model.

---

## 3. Building the Model

This chapter describes the steps to build the prediction model. The chapter has the following structure:

In Section 3.1, the *Hardware Setup* of the subtask experiment is described; also the (network) connections between devices are illustrated.

*Application and Subtasks* (Section 3.2) documents the Android application used to present subtasks in the experiment and automatically mark the subtasks on connected systems (eye-tracking and driving simulation).

Section 3.3, *Test Subjects and Procedure*, characterizes the group of test subjects and explains the experimental procedure used.

The postprocessing and treatment of problems (e.g., drop-outs) of the experimental data is addressed in Section 3.4 *Postprocessing and Problems*.

Section 3.5 *Prediction Model – Calculation Methods* explains some basics for an easier start, e.g., for a developer who wants to transfer ideas. The implementation of the prediction model is open source. Therefore, the section does not go into detail; for a closer examination the source code is available.

The experiment is intended to construct the prediction model, thus no hypotheses are stated before. However, the intra-individually connected data sets of different measurement methods (driving data, eye-tracking, DRT, occlusion, baseline) invite descriptive analysis. Therefore, in Section 3.6, *Descriptive Results*, some results and comparisons are presented and discussed.

## 3.1. Hardware Setup

An experiment was carried out in April 2015 to assess subtasks in the driving laboratory (mockup '1') of the institute. The Bachelor Thesis of Andreas Janiak included parts of the experiment, scripting for DRT and occlusion calculations and the in-depth assessment of driving metrics. For this purpose, a metric similar to the MDEV (ISO 26022, 2010) was used. This lateral metric was also adapted to calculate a longitudinal MDEV. For this thesis, other metrics are used based on 'drifting' (see Section 2.3).

As can be seen by the following description, the setup is quite complex. In a former experiment (December 2014), this led to unpleasant large data drop-outs, due to different errors (e.g., unnoticed network disconnections). The setup, application, subtasks, procedures and checks were revisited and refined afterward. Therefore, the former results are incompatible and not used within this thesis and model.

The overall laboratory situation can be seen in Figure 3.1 and Figure 3.2. After explanations and training, the examiner was located behind the test subject. The driving scene is a car-following scenario similar to AAM and NHTSA guidelines, adapted to German Autobahn specifications and used in several experiments at the institute (e.g., Krause et al., 2015a). For more technical details, see also the description of Figure 2.3 (p. 22). The driving simulation was SILAB 4 (WIVW GmbH, Veitshöchheim). The mockup has one screen (55") for the driving scene and a separated LC-panel for the speedometer. The mockup has a hi-fidelity steering wheel, an accelerator pedal and a brake pedal.



Figure 3.1.: Laboratory setup subtask experiment

Eye-tracking was achieved with the head-mounted Dikablis system (titan frame, 25 fps), with two USB-frame-grabbers and the Dikablis Recorder 2.5. The tablet to simulate an IVIS was a Sony Xperia Z Ultra (6.4"). The rotary knob was a BMW spare part (order



Figure 3.2.: Laboratory setup subtask experiment

number: 6944884) with 24 indents per rotation; internally modified with an Arduino Nano and a Bluetooth module to transmit signals to the Android tablet. This small side project (modified rotary knob) was released open source (Krause, 2015c). The rotary knob was mounted into the armrest and coupled to the tablet (IVIS) via Bluetooth.

PLATO spectacles (Translucent Technologies, CA) were used for the occlusion method. To connect the occlusion goggles to the Ethernet, an Arduino with an Ethernet-shield was programmed and connected to the PLATO driving circuit via the western-plug extension port. So, the occlusion goggles transmitted the current state (open/close) to the tablet, which enabled the tablet to record the state in protocol files. Later shutter open times for each subtask were calculated based on these files. The Arduino paced a 1.5s open, 1.5s closed occlusion protocol. This small side project was also released open source (Krause, 2015b).

To assess the cognitive workload, the Detection Response Task (DRT) method was used in the variation: Tactile Detection Response Task (TDRT) (ISO/DIS 17488, 2014). The DRT continuously presents a stimulus every 3–5 seconds and the test subject has to respond quickly with a button press. The reaction time or missed reaction holds information about the cognitive workload a test subject is currently exposed to. Higher workload prolongs the reaction times. In TDRT, the stimulus is given with a vibration motor. For the experiments, the open source Ethernet Arduino DRT was used (Krause and Conti, 2015). The driving circuit was built with an TIP120 transistor and one forward diode to reduce the driving voltage of the motor. The motor was a coin vibration motor (type number: 308-100) from Precision Microdrives (UK).

To connect the different systems locally, a Linksys WRT54GL (DD-WRT) Wifi-router was used and not linked to other networks. The data traffic for the nodes in this separated local network was low. Most connections used the connection-oriented, potential slower (non-realtime) TCP instead of the stateless, fast UDP. It can be assumed that the non-realtime behavior of, e.g., the Android application itself is more severe than potential network latencies in the intentionally small and separated Local Area Network. The different connections for the different measurement methods are summarized (see Figure 3.3):

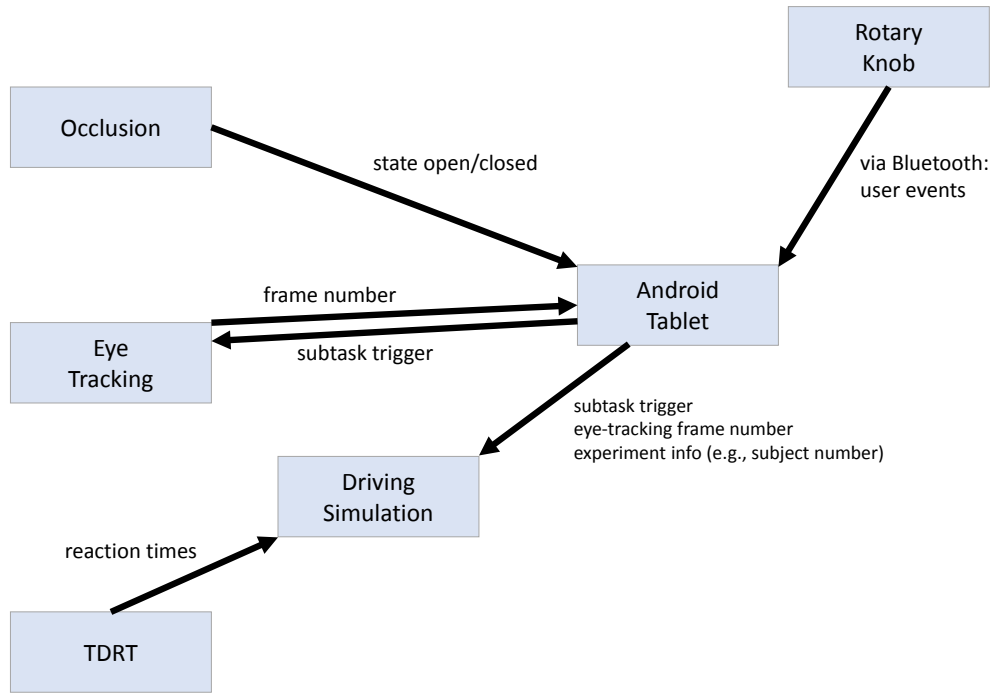


Figure 3.3.: Network Connections

- During occlusion measurements, the occlusion spectacles (Ethernet) transmitted their state to the tablet (WiFi), where the state was logged together with subtask performance.
- When driving the car-following task, the tablet (IVIS) sent subtask triggers to the eye-tracking system and the driving simulation. The required Dikablis format of triggers is mentioned in Section 3.2. In exchange, the eye-tracking systems sent back the current frame number of the recorded video file to the tablet. The tablet forwarded this information to the driving simulation. The recording of frame numbers in the driving simulation can be seen as a fallback solution for synchronization. During the experiments, a continuously increasing frame number (shown in the driving simulation administration panel) provided feedback to the examiner that the eye-tracking system is recording and connections are established and working. In the Android app on the tablet, the subject number and the type of measurement are available, and are transmitted to the driving simulation for logging.



- In the TDRT trials, the same equipment as in car following is used. Moreover, the Ethernet-capable Arduino DRT sends the measured reaction times (or misses) to the driving simulation. These are logged together with the subtask triggers from the tablet in the driving simulation system to enable later assessment of reaction times during subtasks.

## 3.2. Application and Subtasks

The subtasks are implemented in an Android application. This application is available open source (Krause and Prasch, 2016). The open source repository also contains a compiled APK which can be installed on Android devices<sup>1</sup>. The following holds a description of the application; the appearance and function of the subtasks are particularly illustrated. An alternative or valuable support for this section could be to install and interact with the application. Details of the requested inputs can be perused in the source code or in Appendix C (App Parameters). The subtasks were selected based on some years of practical experience with IVIS assessment (cf. Popova-Dlugosch et al., 2011), to represent typical interactions with the common devices (touchscreen and rotary knob). Most subtasks had been tested with different parameter settings (e.g., enter 2, 4, and 8 characters). Therefore, the database holds a reasonable range of interactions. Future versions of the prediction model may be further enhanced with voice interactions and touch pad (handwriting recognition) subtasks.

The application first asks the examiner for general information (Figure 3.4): subject number, touchscreen or rotary knob, experimental condition (accommodation, occlusion, etc).



Figure 3.4.: App config/start screen

After the configuration, the subtasks are presented in randomized order. Presenting a subtask is divided into three parts (Figure 3.5):

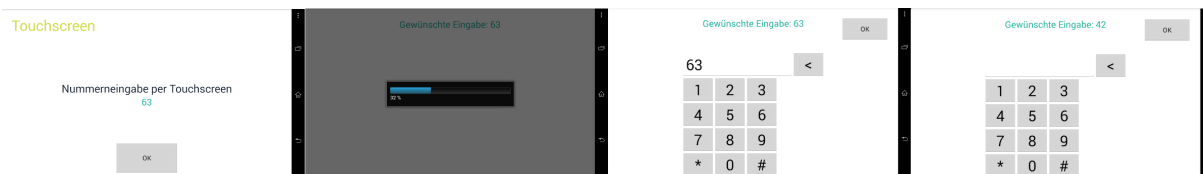


Figure 3.5.: Example of a workflow for one subtask block

<sup>1</sup><https://github.com/MichaelKrause/widgets/blob/master/app/build/outputs/apk/app-debug.apk> (accessed 10/18/2016)

- the instruction screen (e.g., please enter 63) prepares the subject and offers an OK button (or push on the rotary knob) to start the subtask
- an optional delay with different visualizations after the OK-press
- the subtask itself. Some subtasks are carried out one time, others use repeated measurements (trials) without any further instruction screen. Each trial is ended with a click on ‘OK’

Afterward, the next subtask is assessed, starting with an instruction screen, until the app has presented all subtasks. An experimental entity consisting of: instruction, delay, subtask trials; in this thesis is called a *subtask block*. The direct sequencing (repeating) of trials within a subtask block is intentional to prolong the duration, e.g., to enable DRT and occlusion measurements and improve measurement quality.

The Dikablis 2 eye-tracking systems can receive special formatted strings via the network and store them as triggers. The strings begin with two characters, e.g., ‘ES’ for event start, ‘EE’ for event end or ‘EP’ for events that have no duration (point). Then four numbers follow, each with two digits. The numbers are originally intended to characterize: condition, task, subtask and subsubtask. A complete string can be, e.g.,: ES01122000 to mark the start in the experimental condition ‘01’, the task ‘12’, subtask ‘20’. For this experiment, the four numbers are used in the following manner:

- the first number can have two states
  - 01 Touchscreen
  - 02 Rotary knob
- the second number is the ID of a subtask. The ID can be seen in Appendix C: first byte-cast parameter for each GUI widget (e.g., the determined visualized 8 s delay has ID 03)
- the third number can have five states
  - 01 Instruction screen
  - 02 Delay (optional)
  - 03 Subtask trial 1
  - 04 Subtask trial 2 (optional)
  - 05 Subtask trial 3 (optional)

Therefore, when the eye-tracking system receives EE02100100 ES02100300, it knows that in the condition rotary knob (‘02’), subtask ID (‘10’), the instruction screen (‘01’) was ended/acknowledged (‘EE’) and the first trial (‘03’) of this subtask now starts (‘ES’). The number is also sent to the SILAB driving simulation which does not handle strings and interprets and stores 02100300 as 2100300 (without a leading zero). This numbering scheme also emerges in the coding and storage of the results into the HTML and javascript online tool. As can be seen by the previous descriptions, this is not easily readable for humans. Therefore, the eye-tracking analysis software (D-LAB Basis Version 2.0 Feature 2.1; Ergoneers GmbH) can read in a ‘test procedure’-file. This XML connects the numbering to human readable descriptions.

The delays are mixed (randomized) into the procedure in subtask blocks. The delays precede a number input task (typically two numbers, seldom five numbers). However, the short number input is not of interest and discarded; the delay period is the interesting part. In each of the two conditions, nine delays are tested, full factorial 2x3x3: Condition (touchscreen, rotary knob) x Length (2s, 4s, 8s) x Visualization (determined, indetermined, freeze).

After releasing the OK button on the instruction screen (respectively pushing the rotary knob), the delay shows up in delay subtask blocks. The determined visualization can be seen in Figure 3.6(a): A progress bar with a percentage indicates, with a grayed out screen, that the system needs time. For the indeterminate situation, an Android circle animation is used (Figure 3.6(b)). In the condition ‘freeze’ (or ‘stalled’), no direct indication is given; the instruction screen changes to the input screen but without any input option and appears stalled (Figure 3.6(c)). After the delay, a number input widget appears.

All delays are non-cancelable and show up after confirming the instruction screen. Regarding Section 2.2, the user always has first-level feedback, that the system recognized the confirmation of the instruction screen. Therefore, all three delay types are second-level delays (dialog level).

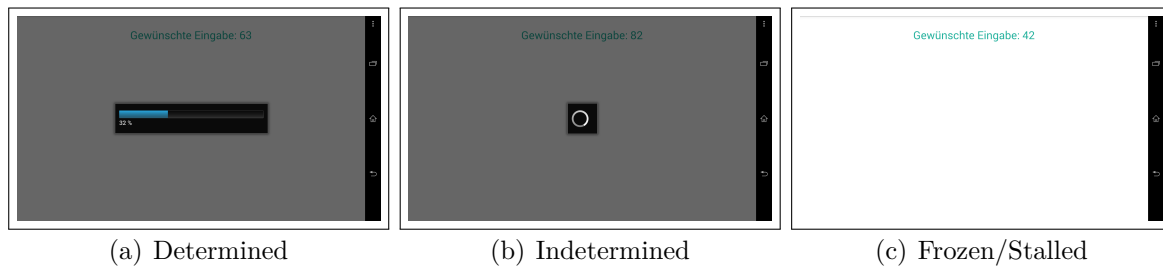


Figure 3.6.: Delay visualizations

The number input (Figure 3.7) is tested for rotary knob and touchscreen in different subtask blocks:

- 3 digits; three trials within block
- 5 digits; two trials within block
- 10 digits; one trial within block

For the ten digits, the number is collected in 3-3-4 groups (Figure 3.7); three and five digits are presented without chunking. The numbers are not containing repdigits (e.g., ‘777’). Input in the touchscreen condition is possible through a virtual numberpad (Figure 3.7(a)). For the rotary knob, a (flat/linear) number ray with a green cursor is used. The OK and backspace functionality is embedded at the ends; the number ray has no wraparound feature (stops at the tails). When a number is confirmed (OK) and the subtask contains repeated measure trials, the test input field is reset.

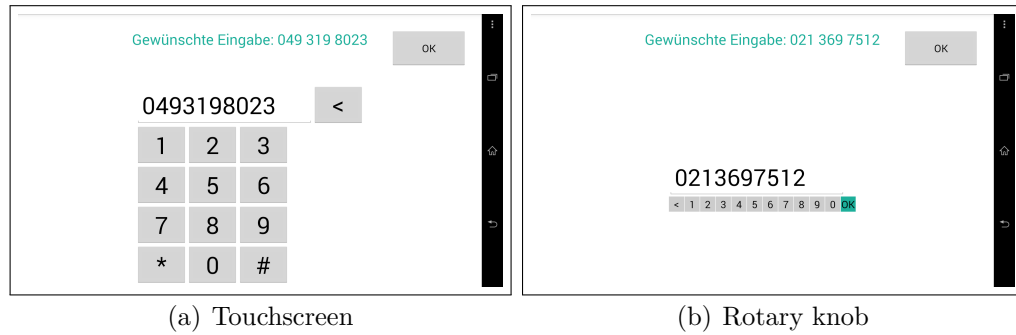


Figure 3.7.: Number input

The list selection (Figure 3.8) is carried out in both conditions. The list contains 100 large German cities in alphabetic order. In the scroll-window six entries are visible. The cursor can be controlled by the rotary knob and the selection is made by pushing the knob. In touchscreen operation, the OK button has to be pressed. When confirming the first trial, the list is reset to the first entry for the second trial. The list selection is performed in three subtask blocks:

- target item is on first visible page ('First'); two trials within block
- target item is in the middle of the list ('Middle'); two trials within block
- target item is on last or second last page ('End'); two trials within block

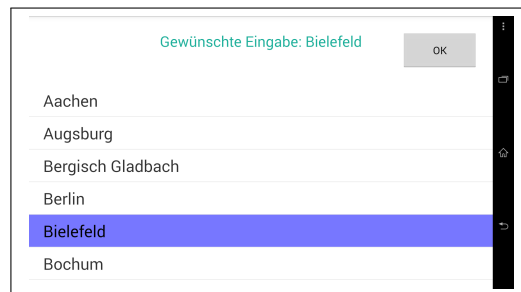


Figure 3.8.: List selection

On the touchscreen, the subtask blocks 'Middle' and 'End' are presented twice with different behavior

- kinetic/inertial scrolling. This is the default Android behavior. A list can be operated by 'fling' gestures (finger movement to kick the list in a direction, the movement decays or can be stopped with the finger).
- non-kinetic/non-inertial scrolling. The friction of the list is set to maximum. Therefore, the list immediately stops when the finger stops scrolling gesture.

While the number input described before (number pad) is, e.g., typical to enter a zip-code or phone number, another opportunity to enter and edit a short and limited number

range is described below. In Android, this interaction is called ‘number picker’; Windows developers know it as ‘spin box’. On Android, the interaction can be shown in two different ways: A number field with +/- buttons above and below or the roll appearance of Figure 3.9(b). The original +/- number picker of Android has the + button directly above the number field. This has the drawback that, when operating the + button, the finger likely hides the numeric field. Therefore, this widget has been implemented for this thesis slightly differently with +/- buttons below the numeric field (Figure 3.9(a)). The +/- buttons have no automatic key repeat feature when held down and need repeated single tap gestures. The roll Figure 3.9(b) can be operated with scroll gestures. It would be also possible to use single taps on the gray numbers above/below, but test subjects are instructed to use scrolling. At the start of each single trial the number is set to 50 and should be adjusted to the given number. This target number is:

- +/-2 different from 50; two trials within block
- +/-4 different from 50; two trials within block
- +/-8 different from 50; two trials within block

In the rotary knob condition, only the visualization from Figure 3.9(b) is used and operated with the rotary knob. A push on the knob is the OK-confirmation.

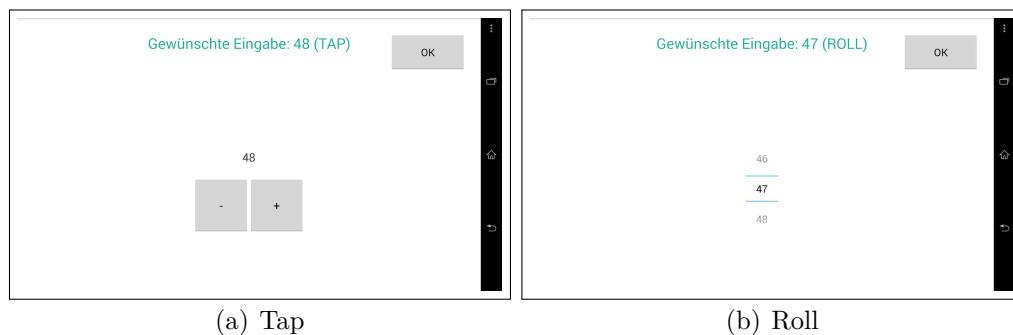


Figure 3.9.: +/- Number input

To adjust a slider is part of two subtasks blocks. At the start of each single trial, the slider is set to zero (left end). The slider implementation allows a maximum value of 100 and snaps automatically to multiples of five (0, 5, 10, 15, ...). In the rotary knob mode, the (clockwise) rotary function increments in steps of five. Pushing the rotary knob confirms the selection. The slider is adjusted in both conditions (touchscreen and rotary knob) in two different settings:

- ‘numerical’; three trials within block. The target value is given numerically e.g., 50. Figure 3.10(a)
- ‘visual’; three trials within block. The target value is given visual on a second slider. Figure 3.10(b)

On the touchscreen, the slider can be operated by sliding or simply by pointing and adjusting (e.g., rolling the finger). Therefore, the task can be also seen as pointing or dragging to a specific 2D point. While on the rotary knob, the task is reduced to one

dimension. The difference between the two variants (numerical/visual) is the way the target is communicated. An example for the visual target could be a climate control without an exact temperature display, but with a slider between cold (blue) and hot (red). An example for the numerical communication could be a control for a temperature display the user wants to adjust to  $20.5^{\circ}$ . The three trials incorporate different targets: near the start, in the middle and closer to the end.

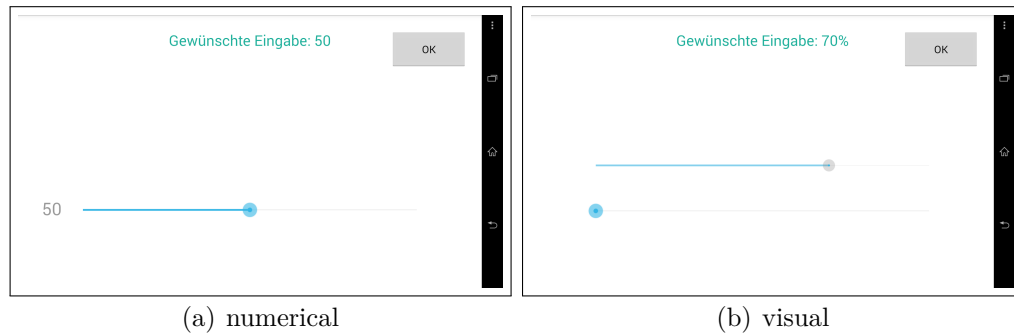


Figure 3.10.: Slider

The text input is tested in three subtasks blocks:

- 2 characters; three trials within block
- 4 characters; two trials within block
- 8 characters; one trial within block

In the touchscreen condition, a default onscreen keyboard is used without any input helpers (e.g., auto completion, word suggestions); Figure 3.11(a). The case is neglected—however, default is lower case. The (touchscreen) trials must be ended by the OK button (not the enter button).

With the rotary knob a linear character selector bar is used (Figure 3.11(b)), which can be operated with a green cursor in the same way as the number ray in the number input described previously (Figure 3.7(b)). The char selector has no internal dictionary or rules to disable impossible characters. The text field is always emptied at the start of each subtask trial. The onscreen keyboard directly opens, the text field does not have to be explicitly selected.

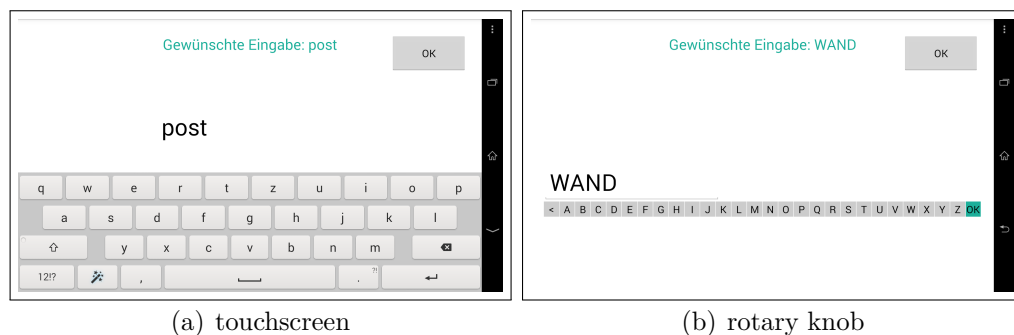


Figure 3.11.: Text input

It is foreseeable that the user can unintentionally hit the confirmation function (OK button or press on the rotary knob) multiple times. Without any precautions, this would skip subtask trials. Therefore, confirmation is only possible if the test subject made a modification to a subtask (e.g., entered a number).

When the app is switched to touchscreen mode, 28 subtask blocks are presented. In rotary knob mode, 23 subtask blocks are presented. The Android tablet was configured to provide acoustic feedback (bright click) on touch taps, with maximum volume.



### 3.3. Test Subjects and Procedure

The age of the test subject was between 20–32 years (median 23 years). All drivers had a driver license, typically issued between the ages of 16–19. Therefore, the years of driver experience was between 3–14 years (median 6 years).

The mileage per year can be seen in Figure 3.12

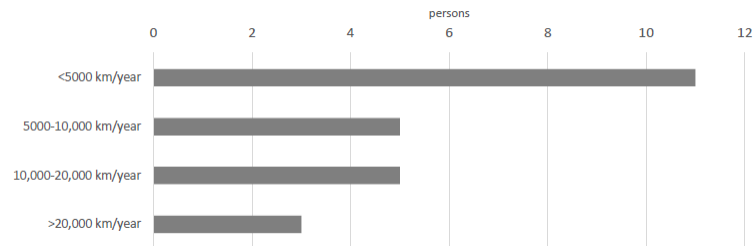


Figure 3.12.: Mileage

Out of the 24 persons, 11 (46%) were females. Two persons needed glasses, four used contact lenses; one person had a known red-green blindness. Four persons were left-handed. The simulated vehicle has automatic gears; six persons had no previous experience with an automatic car. A driving simulation was not driven before by ten people. Six persons were classified as extensive simulation drivers (> 5 experiments).

The frequency of touchscreen usage for different devices is queried with a five point Likert scale (never—often). Most experience stems from mobile phones; 20 persons chose the rightmost option, four persons one scale point below. Four persons had no previous experience with a rotary knob.

Test subjects who participated in the previous, discarded (pre-test) experiment in December 2014 were not allowed to participate. The participation was voluntary, without compensation.

#### Procedure

Subjects signed a consent form concerning voluntary participation. They were informed that they could quit at any time without any justification and that the eye-tracking system records video and audio. A statement also clarified that the subject is not judged, and only the system and the situations would be assessed. With three presentation slides about general instructions and driving-task-related notes, the experiment started (cf. Appendix B). The subjects drove the driving simulation for at least two minutes or until they felt comfortable with the feedback of the examiner about the vehicle-to-vehicle distance. For training, the application (Section 3.2) was operated completely one time with a touchscreen and one time with a rotary knob while standing still. During this training, it was verbally clarified that the instruction ‘Don’t correct errors’ is intended for hard-to-correct mistakes (e.g., mistyping at the second digit in 089 319 8023 and noticing the error when entering the last digit). It was acceptable to edit the last entered digit/character. On touchscreens many users do this automatically. This procedure is for the automated (unstoppable) test program. More importantly, the data sets should intentionally include, to some extent, non-perfect interactions to improve modeling.

It is noteworthy that NHTSA (2014, p.14) states: "Error means that a test participant has made a significant incorrect input [...]".

The secondary task was also trained alongside the primary task (touchscreen and rotary knob) until the subjects acknowledged understanding of the dual-task situation (typically after some widgets). The last step of the training was the recording of a baseline driving performance. The subjects drove 2.5 minutes without a secondary task. The last 90s of this measurement were later used to calculate some baseline driving metrics.

The core of the experiment consists of four sections:

- *Baseline*. Operating the application without driving; Total Task on Time measurement.
- *Occlusion*. Operating the application with occlusion glasses
- *AAM*. Operating the application while driving (with eye-tracking)
- *DRT*. Operating the application while driving and with a Detection Response Task

The four sections were counterbalanced and within the sections the order of rotary knob and touchscreen operation was also changed.

#### **Baseline**

This is the most obvious condition; without driving, DRT or occlusion

#### **Occlusion**

The participants were instructed for the occlusion (Appendix B). Before each input device (touchscreen/rotary knob) persons were briefly educated in the occlusion setup by operating some subtask widgets.

#### **AAM**

In this condition, the application was operated while driving. The glance behavior was recorded with the head-mounted eye-tracking.

#### **DRT**

In addition to the setup of the condition AAM (including the eye-tracking), this condition used a Tactile-DRT setup. The subjects were instructed for the DRT (Appendix B). A baseline reaction time without driving was recorded for 1 minute (about 15 stimuli); subsequently called *static DRT baseline* (only operating the TDRT; single task). For another minute, a second baseline was recorded while driving; *dynamic DRT baseline*. This can be seen as a dual-task setup (driving and TDRT). The application was then involved, this can be seen as a triple-task setting (driving, secondary task and TDRT).

The duration of the experiment per person was about one and a half hours.

## 3.4. Postprocessing and Problems

General workflow:

The experimental data for the various metrics are recorded on different systems (log files on the tablet, eye-tracking system and driving simulator) and therefore different file locations and formats. These sources are examined with different Matlab scripts to save the subtask metrics to Excel files. Within the Excel files, some manual adjustments are made to cope with recording errors and input flaws. The content of the Excel files are transferred (copy&paste) to a local offline instance of the open source conversion tool Mr. Data Converter (Carter, 2010). In the tool the ‘JSON Array of Columns’ format (in file DataGridRenderer.js) has been slightly adjusted to the needs of this thesis. The JSON structures are saved to javascript files which are included and accessed by the online prediction tool. These JSON structures (i.e., javascript variables) are the database.

### Eye-tracking Data

The eye-tracking data were manually inspected to maximize the pupil detection and adjust, e.g., shifted headunits. For this purpose the Dikablis Analysis tool of the D-Lab software suite was used. Within D-Lab, three Areas of Interest (AOI) were defined for each subject: Driving Scene, Speedometer, IVIS (Figure 3.13).

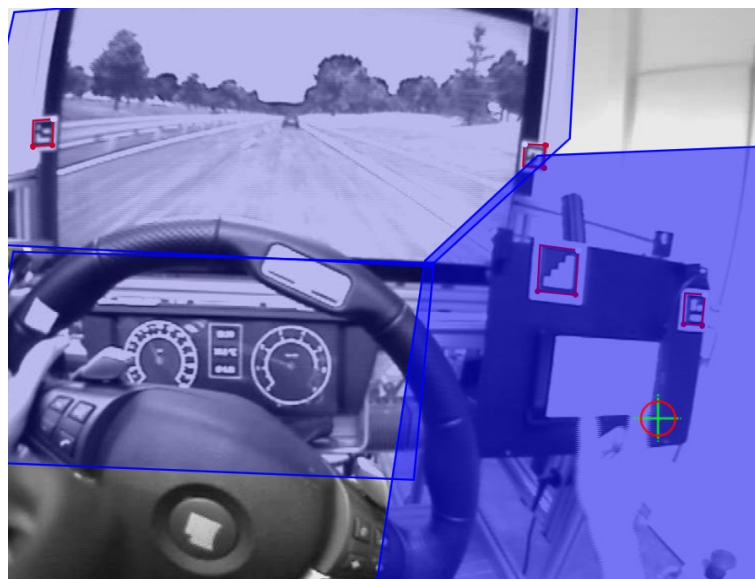


Figure 3.13.: Areas of Interest (Driving Scene, Speedometer, IVIS) in D-Lab

The glance data were post-processed with D-Lab (Basis Version 2.0 Feature 2.1; Ergoneers GmbH) default options: 120 ms blink removal and 120 ms cross through glance handling. The resulting glances were exported to XML files. Together with trigger files (XML), which hold the start and stop frames of subtasks, the exported gaze data are used by Matlab scripts. The Matlab scripts implements the fractional Number of Glance (NOG) approach mentioned in Section 2.5 to counteract artificial increase of the NOG and decrease of the Single Glance Duration. The scripts also calculate eyes-off-road glances (non-driving AOI glances). The Dikablis system stores the location of the pupil without

saccade and fixation differentiation. The term ‘dwell time’ could be more appropriate than ‘glance time’ (cf. DIN EN ISO 15007-1, 2003). Therefore, the AOI are intentionally drawn in a manner that, on the one hand, minimizes the chance of a false AOI detection and, on the other hand, splits the way of saccades approximately evenly between the driving scene and IVIS; to get close to the glance time definition (leading saccade + dwell time). The Dikablis system records with 25 fps (40 ms). If one assumes a saccade of 100 ms, this would result in about two frames. On the saccade toward the IVIS, there is a chance that one frame of the saccade is within the IVIS AOI and on the saccade away from the IVIS too.

The data are finally stored in two javascript files. *glance.js* holds the values used for the AAM glance predictions and is based on glances toward the IVIS task AOI. The file holds, e.g., that a test subject needed for a specific subtask a total glance time of 9.8 s and a Number of Glances of 5.5. From these values, the Single Glance Duration ( $9.8\text{ s}/5.5 = 1.78\text{ s}$ ) is derived.

The file name *eor.js* stands for eyes-off-road and is used for the NHTSA glance predictions. The file holds an array for the count (i.e. Number of Glances) and the Single Glance Duration for every subtask of every person. The type of the glance is coded with characters and described in the file. For example, *count[pt0.5, t1, s1]* and *sgds[pt1, t2, s0.5]* would signal that the subject had a half glance toward the ‘t’ask that started ‘p’reviously (pt0.5) before the subtasks, the single glance time within the subtask of this glance is 1 s (pt1). Additionally, one complete glance (t1) with two seconds (t2) toward the IVIS, and finally one 0.5 s glance toward the speedometer. This, obviously more complicated, structure is also used for the glance data visualization (Appendix, Figure A.7, p. 127).

## Driving Data

The driving simulator records the distance to the leading vehicle. However, only if the center of gravity (COG) of the simulated car (ego-car) is within the intended lane. In rare cases, even the COG crosses the lane boundary for a short time and the following headway calculation returns zero. Drift of Following Headway (DFH) is based on differences (differentiation). Even rare drop-outs could have a significant impact on a small amount of subtask values. Therefore, these small gaps are filled by linear interpolation<sup>2</sup>.

The driving data are stored in *driving.js*. The file holds the non-normalized sum of the rectified differences of the lane position (*sumAbsDiffLanePosition*), the non-normalized sum of the rectified differences of the following headway (*sumAbsDiffTimeHeadway*) and the duration of the subtasks. The normalized DLP and DFH can be calculated by summing up the *sumAbsDiffLanePosition*-values of subtasks and normalizing them by the duration (see Section 2.3). The file also holds the baseline driving performance of each subject, to calculate the percentages of driving performance deterioration.

---

<sup>2</sup>Matlab function *inpaint\_nans()* by John D’Errico 2009. release 2 release date 4/16/06

## Occlusion Data

The *occ.js* stores the occlusion and baseline values. The array *tttbase* holds the baseline Total Time on Task (TTT while standing). *tttocc* holds the Total Task on Time during the occlusion condition. *tsot* is the Total Shutter Open Time within the *tttocc*. The occlusion glasses transmitted their current state to the task tablet (see Section 3.1). Therefore, the TSOT within *tttocc* for subtasks did not have to be estimated by approximations, it was calculated based on log files.

## Detection Response Task Data

The values of the TDRT are saved in *drt.js* (or alternative *drtmedian.js*). The *totalCount* variable holds how many stimuli were presented. The *hitCount* and *missCount* shows how often the subjects reacted (hit) or missed the reaction. Any difference in the count variables would be a rare ‘cheat’ reaction ( $< 100$  ms):  
 $cheats + hitCount + missCount = totalCount$ .

The *rt* variable stores the reaction times in microseconds. In the currently used *drt.js*, *rt* holds the average reaction time during a subtask of each person. The alternative *drt-Median.js* holds all reaction times during a subtask and would calculate internally on time, e.g., the median reaction time, when needed.

The files also contain the baseline performance of the subjects: *baselineRt* (single task baseline reaction time, only TDRT) and *baselineDruRt* (dual-task baseline reaction time, TDRT and driving).

To check for cheating strategies (e.g., repeatedly pressing the button) during analysis a button down ratio has been calculated (button presses divided by count of stimuli). For the 24 subjects, the minimum ratio is 0.84 and the maximum 1.41 (average 1.03, SD 0.09). The inter-quartile range (Q1–Q3) is 0.99–1.05. Therefore, there are no indications for continuous cheat strategies of a test subject.

The hit rate of the 24 subjects is 76%–100% and the inter-quartile range (Q1–Q3) is 96%–98%. Therefore, all subjects were able to work on the TDRT.

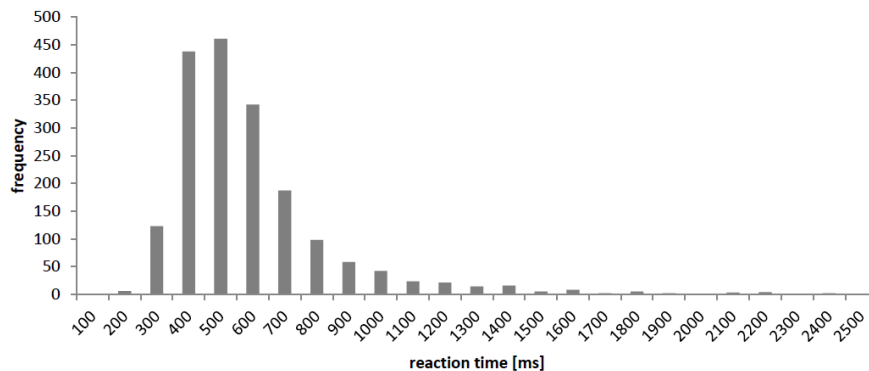


Figure 3.14.: Histogram of TDRT reaction times (hits only) of all subjects

In a former experiment (Krause et al., 2014a), one person regularly responded at 1 s after stimulus onset which was interpreted as the subject unconsciously waiting 1 s for the second stimulus (vibration switch-off after 1 s). The data in Figure 3.14 demonstrates no untypical artifacts. A potentially small discontinuity in the exponential decay at around 1400 ms may be interpreted as an indication of a small ‘second stimulus’ switch-off artifact (reminder). A possible minor improvement for DRTs might be to fade out stimuli. Overall, the reaction times appear typical (Figure 3.14).

#### Input Errors and Connection Problems

The inputs entered were checked against the requested input. In four experimental conditions the persons had to enter overall 88 inputs ( $24 \times 4 \times 88 = 8448$ ). Of these, 228 (2.7%) were not according to the requesting instruction. This indicates a high engagement of the test subjects. The input after a delay subtask (enter a number) was discarded; the interesting part was the delay itself. The user input is seen as necessary padding (sacrificial task). Therefore, only 195 out of the 228 identified conflicting inputs were further inspected. Out of the 195 events, three were judged to be potentially severe (crucial mismatch in requested and given input length). As a solution the affected metrics of a single person for a specific subtask were replaced by the average value of all other persons:

- baseline condition, subtask 1250500, test subject 4
- AAM condition, subtask 2120300, test subject 9
- occlusion condition, subtask 2120300, test subject 11

In the conditions with driving simulation (AAM and DRT condition) the tablet started a server, the driving simulation (client) connected itself to this server, with a retry cycle of several seconds. Therefore, in some situations the subject already worked within the application (IVIS task) when the driving simulation connected. This primarily affected the start trigger of the first instruction screen which is not severe, because the performance metrics during the instruction screens are neglected. Nevertheless, in rare cases some information was lost:

- AAM condition, subtask 2070200 (delay, 2s freeze, rotary knob), test subject 6. Solution: replaced by subtask 1070200 (delay subtask, 2s freeze, touchscreen)
- AAM condition, subtasks 2230300/2230400 (visual slider, rotary knob), test subject 18. Solution: replaced by average value of all other test subjects
- AAM condition, subtask 1060200 (delay, 8s, indetermined, touchscreen), test subject 18. Solution: replaced by subtask 2060200 (delay, 8s, indetermined, rotary knob)
- DRT condition, subtask 2180300 (list selection, end, rotary knob), test subject 15. Solution: replaced by 2180400 (second trial)

## 3.5. Prediction Model – Calculation Methods

This chapter addresses the calculations carried out in the prediction model to combine sub-tasks to a task. For details, the open source code and online tool itself can be consulted; available e.g. via: <http://www.distract.one>.

The 13 predicted metrics can be divided into metrics that are cumulative summed up and more complex metrics.

The cumulative metrics are:

- Total Time on Task (TTT static, non-driving)
- Total Time on Task while driving
- Glance – AAM – Total Glance Time (task related)
- Glance – AAM – Number of Glances (task related)
- Glance – NHTSA – Total Eyes-Off-Road Time
- Glance – NHTSA – Number of Glances (eyes-off-road)
- Occlusion – Total Shutter Open Time (TSOT)

More complex combined are:

- Glance – AAM – Single Glance Duration (task related)
- Glance – NHTSA – Single Glance Duration (eyes-off-road)
- Occlusion – R-Metric (TSOT/TTT)
- Tactile Detection Response Task (TDRT)– Deterioration in Reaction Time (%)
- Driving – Deterioration in Lateral Drift (%)
- Driving – Deterioration in Longitudinal Drift of Headway (%)

Some subtasks are tested repeatedly (trials). These trials are typically stored separately and merged to an average value on time.

The prediction of **single glances** for the AAM procedure (task-related glances) and for the NHTSA procedure (eyes-off-road) is different. Both use the fractional NOG approach, discussed before (see Section 2.5), to obtain Single Glance Duration based on the Total Glance Times and the Number of Glances (TGT/NOG-task-related, respectively TEORT/NOG-eyes-off-road). For subtasks with repeated trials, the predictions calculate an average for each individual test subject based on all trials. To combine the single glances of subtasks to a complete task, SGD calculations use a weighted mean. The factory for weighting is the fractional NOG. For instance, if one subtask has one glance with 3s and another subtask has two glances with 1.5s, the combined single glance task metric for the complete task would result in  $(1 \times 3s + 2 \times 1.5s) / (1+2) = 2s$ . Both procedures

calculate 24 individual values, i.e. one result for each subject. This distribution and characteristic values (e.g., the median) are visualized in the online interface. Furthermore, from this distribution, 24 values are randomly sampled (with replacement) 1000 times. These 1000 bootstrapping distributions are compared to AAM and NHTSA criteria. The result is reported as a percentage concerning how often the criteria have been met.

The occlusion **R-metric** (TSOT/TTT) is based on the additive TSOT and TTT. In the configuration menu, a checkbox can be enabled or disabled to change whether System Response Times should be incorporated or neglected when calculating occlusion metrics.

The **TDRT metric** is calculated as deterioration (percentage) of the reaction time compared to the dual setting baseline reaction time (TDRT and driving). Therefore, -10% would indicate that someone responded during this subtask faster than his/her baseline. One hundred percent means the reaction time was prolonged, e.g., from 200 ms to 400 ms. The composition of subtasks to tasks uses a weighted mean of the reaction times, similar to the calculation of Single Glance Durations. The factor for weighting is the number of reactions (hits). If a subtask was tested in repeated trials, it is attempted to incorporate an average of all trials. Due to a miss or the short duration of a subtask (e.g., a 2 s System Response Time subtask) for some test subjects and some subtasks, it is possible that no reaction time is available. The availability is indicated by the N value in the subtask selection window below the data table (Figure 3.15). This unavailability is simply accepted within the modeling approach; it could also happen in any other experiment. Nevertheless, the repeated trial setting for short subtasks was intended to increase availability and quality (also for occlusion measurement).

Symbol-Image	ID	Mode	Type	Subtype	Subsubtype	TTT [s]	DRT [%]
[Icon]	01	Touchscreen	Delay	Determined	2s	2	63
[Icon]	02	Touchscreen	Delay	Determined	4s		
[Icon]	03	Touchscreen	Delay	Determined	8s		
[Icon]	04	Touchscreen	Delay	Indetermined	2s		
[Icon]	05	Touchscreen	Delay	Indetermined	4s		
[Icon]	06	Touchscreen	Delay	Indetermined	8s		

Min	-36%
Q1	3.8%
Median	63%
Q3	81%
Mean	57%
SD	60%
P85	110%
N	13

Figure 3.15.: Availability (“N=13”) of test subjects for a subtask TDRT metric below the subtask data table

The **driving metrics** (DLP, DFH) use the approach described in Section 2.3. The drifting within subtasks is stored non-normalized; i.e., not divided by the subtask duration. The non-normalized subtask metrics are additively summed-up to a task and are normalized by the additively summed up task duration. This result is referenced to the baseline driving performance (deterioration %) of every test subject.



## 3.6. Descriptive Results

For the experiment, no hypotheses had been stated before. Therefore, the results are compared descriptively. The subtask data are also programmed into the open source online tool: <http://www.distract.one>. This enables a closer examination of individual topics of interest.

For this thesis, the alpha level in statistics is set to 5%, if not indicated otherwise (e.g., Bonferroni correction). Alpha level 5% is also used when calculating observed powers.

### 3.6.1. Comparison to Former Experiment (Age)

Results from this experiment (subjects aged 20–32 years), *Experiment Y*, have been compared to subtasks of former experiments in Krause et al. (2015b), *Experiments M*. Within Krause et al. (2015b) groups according to the AAM subject sampling (45–65 years, middle-aged) operated touchscreen and rotary knob subtasks in different experiments to finish normal task interactions. These interactions were manually coded by student assistants into subtasks. From these subtasks, eight were chosen for comparison (four on a touchscreen, four with a rotary knob) to compare experiment Y and experiments M.

The touchscreen subtasks included entering a phone number (approximately 10 digits), which was judged to be comparable to subtask ID12 of this experiment (Touchscreen Input Number 10 digits). Krause et al. (2015b) used a subtask where the subjects pressed a special button on the alphabetic virtual keyboard to switch to the numeric input and entered one number. This is compared to ID26 (Touchscreen Input / Typing Alphabetic 2 chars). Also a slider was adjusted in Krause et al. (2015b) and related to ID25 (Touchscreen Slider Visual). The list scrolling from Krause et al. (2015b) is contrasted with ID14 (Touchscreen List Selection Kinetic Scrolling mid). On the rotary knob, Krause et al. (2015b) included subtasks to enter 5 numbers, 10 numbers, 2 characters and 4 characters. This is comparable to the equivalent ID39, ID40, ID41, ID42 of this experiment.

Table 3.1, Table 3.2 and Table 3.3 present the comparison for the TGT, SGD and TSOt (mean and 85<sup>th</sup> percentile). A positive percentage indicates that the middle-aged group needed more time.

When looking at the TGT for the touchscreen interactions (Table 3.1). The character and number input seems comparable, while the slider adjustment and scrolling demonstrates a larger difference. The tablet in Krause et al. (2015b) was a low performance device, compared to the tablet for this experiment. This could contribute to the longer TGTs for the two subtasks (slider and scrolling), which may later on profit from the high performance tablet.

Regarding the rotary knob, the middle-aged group needed slightly more glance time when entering 2 characters. For five digits the total glance times are comparable and for the more complex interactions (4 characters and 10 digits), the middle-aged group needed (counterintuitively) about 20% less glance time. A reasonable explanation could be again the hardware. While the experiment presented in this thesis used a BMW rotary

Device	Subtask	Exp. Y mean [s]	Exp. M mean [s]	$\Delta\%$	Exp. Y P85 [s]	Exp. M P85 [s]	$\Delta\%$
touch	input 10 digits	6.8	7.7	13%	7.7	9.8	27%
touch	input 2 chars	2.4	2.5	3%	2.8	2.9	7%
touch	adjust slider	2.7	3.2	21%	3.5	3.8	10%
touch	scroll list	6.3	8.3	33%	7.1	10.2	43%
rotary	input 5 digits	7.6	7.6	0%	9.0	8.8	-2%
rotary	input 10 digits	15.5	12.7	-18%	18.2	14.8	-19%
rotary	input 2 chars	3.9	4.5	16%	4.6	5.7	25%
rotary	input 4 chars	8.0	6.1	-24%	9.2	7.5	-19%

Table 3.1.: Comparison of subtasks from this experiment (Y young) to former experiments (M middle-aged). **Total Glance Time**

Device	Subtask	Exp. Y mean [s]	Exp. M mean [s]	$\Delta\%$	Exp. Y P85 [s]	Exp. M P85 [s]	$\Delta\%$
touch	input 10 digits	2.3	1.7	-27%	3.0	2.1	-32%
touch	input 2 chars	2.2	2.1	-8%	2.9	2.6	-10%
touch	adjust slider	2.0	1.5	-27%	2.7	2.2	-18%
touch	scroll list	1.8	1.5	-14%	2.4	2.0	-14%
rotary	input 5 digits	1.7	1.5	-14%	2.3	1.5	-33%
rotary	input 10 digits	1.8	1.2	-33%	2.3	1.6	-31%
rotary	input 2 chars	1.6	1.4	-12%	2.0	1.8	-11%
rotary	input 4 chars	1.6	1.4	-12%	2.1	1.7	-19%

Table 3.2.: Comparison of subtasks from this experiment (Y young) to former experiments (M middle-aged). **Single Glance Duration**

Device	Subtask	Exp. Y mean [s]	Exp. M mean [s]	$\Delta\%$	Exp. Y P85 [s]	Exp. M P85 [s]	$\Delta\%$
touch	input 10 digits	5.7	7.8	37%	6.4	9.1	42%
touch	input 2 chars	1.9	2.4	26%	2.3	3.5	54%
touch	adjust slider	2.4	3.8	56%	3.2	4.7	50%
touch	scroll list	5.4	8.4	57%	6.4	11.7	83%
rotary	input 5 digits	6.9	9.0	29%	8.3	10.5	26%
rotary	input 10 digits	11.9	15.5	30%	14.6	18.3	25%
rotary	input 2 chars	4.3	5.3	24%	5.1	6.1	20%
rotary	input 4 chars	6.9	8.8	27%	8.1	9.8	21%

Table 3.3.: Comparison of subtasks from this experiment (Y young) to former experiments (M middle-aged). **Total Shutter Open Time**

knob with 24 indents, Krause et al. (2015b) used a Mercedes rotary knob with 30 indents. The ratio (24/30) may possibly contribute to the glance-time saving. Together with the generally longer Single Glance Duration strategy of the younger group (Table 3.2) the potential coarser device (24 indents) may contribute to the counterintuitive result, when input interactions grow longer.

This comparison only can offer indications. Not only was the age different between the groups, the hardware was also different. On the other hand, the results could indicate that the hardware is perhaps more important for the Total Glance Time than the age. This would render precise predictions impossible anyway, without knowing the hardware of a prototype or the software performance of a specific implementation. A tool for approximate estimations in an early development stage could nevertheless be valuable in making reasonable decisions.

When comparing Table 3.1 to Table 3.3 it can be seen that occlusion (TSOT) is a reliable low-cost method to obtain the visual demand (TGT).

Table 3.2 for the Single Glance Durations demonstrates that the two middle-aged groups (touchscreen, rotary knob) from Krause et al. (2015b) chose a strategy with shorter glances (negative percentages). This indicates that testing and predicting single glance times for the younger group can be seen as a kind of worst case, related to guideline criteria.

The TSOT results in Table 3.3 illustrate that both middle-aged groups (touchscreen and rotary knob) from Krause et al. (2015b) seem to be challenged by the occlusion and need longer Total Shutter Open Time (positive percentages) than the younger group. This would indicate that testing and predicting TSOT with the younger group could be seen as a kind of best case, related to guideline criteria.

### 3.6.2. Glance Metrics With and Without TDRT Measurement

During the TDRT measurement, the glance data were recorded. This is not necessary for TDRT, however the experimental protocol allowed the gathering of this extra data. While visual DRTs (RDRT, HDRT) could obviously interfere with uninfluenced glance behavior, due to the usage of the same channel (visual stimulus), a potential interference of the TDRT (tactile stimulus) with the visual task behavior would be more surprising. If the measurement of glance data simultaneous with a dynamic TDRT setup (driving) would be valid, glance data (visual), DRT data (cognitive) and driving data (visual-manual interference) could be gathered in one trial and used to assess the driver distraction potential.

The DRT stimulus appears every 3–5 seconds. Therefore, for this assessment the data of six longer touch subtasks (ID17, ID16, ID12, ID14, ID15, ID28) and six longer rotary knob subtasks (ID40, ID43, ID42, ID39, ID45, ID46) are used. The mean Total Task on Time while driving for these subtasks is 8–25s. The data for TTT, TGT, SGD and NOG for each subtask is normalized for each subtask for each of the 24 test subjects. For example, if a test subject needed for a subtask with TDRT a TGT of 11s and without TDRT 10s, the normalization ( $11\text{ s} / 10\text{ s}$ ) would give 1.1, indicating that in the measurement with TDRT the TGT was 10% longer. If a subtask was measured with repeated measurement, the additional trials are handled separately. This led to 18 data subtask trials and  $(18 \times 24) N = 432$  overall data points for this assessment.

The ratios for TTT ( $M = 1.17$ ;  $SD = 0.52$ ; Median = 1.07), TGT ( $M = 1.18$ ;  $SD = 0.46$ ; Median = 1.10), SGD ( $M = 1.18$ ;  $SD = 0.48$ ; Median = 1.08) and NOG ( $M = 1.12$ ;  $SD = 0.55$ ; Median = 1.00) demonstrate that there is a tendency (with large standard deviations) that the TDRT can lead to slightly longer TTT, TGT and SGD. The potential influence from the tactile stimulus channel to the visual glance behavior is not obvious; perhaps via manual task interference of the response button press or cognitive processes.

The DRT methods are promising and helpful. However, they are performed simultaneous to the task under evaluation and can have the potential to slightly alter the performance and behavior of subjects.

### 3.6.3. Glance Metrics During Delays

There has no been hypothesis stated before the experiment concerning the delays and visualizations. Therefore, the results are indications, e.g., for further experiments. Nevertheless, an analysis procedure from inference statistics is used to further examine and interpret the glance metrics during System Response Times.

The dependent variables (DV) are the glance metrics:

- Total Glance Time (TGT, respectively TEORT)
- fractional Number of Glances (NOG). For the fractional approach cf. Section 2.5
- Single Glance Duration (SGD) based on TGT/NOG (division by zero was replaced by 0)

The independent variables (IV) are:

- Measurement respective calculation method of glance metrics (eyes-off-road / task-related)
- Experimental setup (without TDRT method / with TDRT method)
- Input device (touchscreen / rotary knob)
- Delay visualization (determined / indetermined / freeze)
- Delay duration (2 s / 4 s / 8 s)

The full factorial (2x2x3x3) 36 delays were experienced by every subject. The independent variable *measurement and calculation method* is introduced to check if the different calculations of metrics in guidelines would have an influence. This analysis uses a five-way repeated-measures MANOVA.

The eyes-off-road method is in related to the NHTSA guideline, while the task-related approach is related to the AAM guideline (glances toward the IVIS AOI). When browsing the glance visualization and screening the columns of the metrics in the online database, the metrics (eyes-off-road versus task-related) seem comparable for typical subtasks; while for SRT subtasks they differ. Opening the visualization (see Figure A.7 p. 127) clarifies that some subjects use the delays to check the speedometer. The assumption that eyes-off-road related metrics and task-related metrics are similar could be wrong, especially when long System Response Times are part of a task.

### Eyes-Off-Road versus Task-Related (IVIS AOI) Glance Metrics

Calculation of eyes-off-road metrics versus using task-related glances to the IVIS AOI would show a **significant** Wilks'  $\lambda = .239$ ,  $F(3, 21) = 22.272$ ,  $p < .001$ ,  $\eta_p^2 = .761$  the power to detect the effect was  $>.999$

A closer examination of the related univariate tests:

Total Glance Time:

$F(1, 23) = 45.871$ ,  $p < .001$ ,  $\eta_p^2 = .666$  the power to detect the effect was  $>.999$

Number of Glances:

$F(1, 23) = 29.761$ ,  $p < .001$ ,  $\eta_p^2 = .564$  the power to detect the effect was  $.999$

Single Glance Duration:

$F(1, 23) = 5.163$ ,  $p = .033$ ,  $\eta_p^2 = .183$  the power to detect the effect was  $.586$

During a delay **all three metrics would be significantly higher for the eyes-off-road measurement approach compared to using task-related glances to the IVIS AOI.**

### Concurrent TDRT Measurement

The measurement without or with a parallel TDRT results in a **not significant** Wilks'  $\lambda = .75$ ,  $F(3, 21) = 2.337$ ,  $p = .103$ ,  $\eta_p^2 = .250$ ; power to detect the effect was  $.506$

It should be noted that this can not be inversely interpreted as a test for equality. Together with the descriptive results reported previously (Section 3.6.2), it can be also seen as an indication that the parallel TDRT changes the glance behavior slightly. When looking one level deeper into the MANOVA analysis (despite the not significant result), the SGD appears almost uninfluenced, while TGT and NOG are slightly increased during the TDRT.

### Input Device

The input device **touch versus rotary knob** would demonstrate a **significant** Wilks'  $\lambda = .611$ ,  $F(3, 21) = 4.448$ ,  $p = .014$ ,  $\eta_p^2 = .389$  the power to detect the effect was  $.806$

A closer inspection of the related univariate tests:

Total Glance Time:

$F(1, 23) = .749$ ,  $p = .396$ ,  $\eta_p^2 = .032$  the power to detect the effect was  $.132$

Number of Glances:

$F(1, 23) = 9.352$ ,  $p = .006$ ,  $\eta_p^2 = .289$  the power to detect the effect was  $.834$

Single Glance Duration:

$F(1, 23) = 6.764$ ,  $p = .016$ ,  $\eta_p^2 = .227$  the power to detect the effect was  $.702$

A more detailed view of the data reveals that the NOG for the rotary knob ( $M = 1.791$ ,  $SE = 0.097$ ) is slightly higher than for the touchscreen ( $M = 1.666$ ,  $SE = 0.088$ ). However, the SGD for the rotary knob ( $M = .836$  s,  $SE = .039$  s) is slightly lower (touchscreen  $M = .932$  s,  $SE = .055$  s). This is, to some extent likely influenced by the fractional glance calculation. These calculations transfer the glance duration before and after the delay partly into the delay duration.

### Delay Visualization

The **visualization of the delay (determined, indetermined, freeze)** would show a **significant** Wilks'  $\lambda = .200$ ,  $F(6, 18) = 11.963$ ,  $p < .001$ ,  $\eta_p^2 = .800$  the power to detect the effect was  $> .999$ . For the univariate view all Mauchly tests for sphericity are not significant (therefore no correction):

Total Glance Time:

$F(2, 46) = 24.667$ ,  $p < .001$ ,  $\eta_p^2 = .517$  the power to detect the effect was  $> .999$

Number of Glances:

$F(2, 46) = 32.583$ ,  $p < .001$ ,  $\eta_p^2 = .586$  the power to detect the effect was  $> .999$

Single Glance Duration:

$F(2, 46) = 6.497$ ,  $p = .003$ ,  $\eta_p^2 = .220$  the power to detect the effect was  $.888$

The pairwise tests show that the **TGT for determined and indetermined visualization is not significant** different  $p > .999$ . The **TGT when freezing is significant higher** ( $p < .001$ ) than during determined or indetermined visualizations.

Regarding **NOG all conditions are significantly different** (determined/indetermined  $p = .009$ ; indetermined/freeze  $p = .001$ ; determined/freeze  $p < .001$ ). During freezing, the NOG is the highest. The lowest NOG is evoked by the determined visualization.

**SGD is significantly lower for indetermined** versus the other two conditions (determined  $p = .007$ ; freeze  $p = .017$ ). Determined and freeze condition are not significantly different ( $p > .999$ ).

### Delay Duration

The **duration of the delay (2 s, 4 s, 8 s)** would show a **significant** Wilks'  $\lambda = .077$ ,  $F(6, 18) = 36.150$ ,  $p < .001$ ,  $\eta_p^2 = .923$  the power to detect the effect was  $> .999$ . While this is not surprising, it is a mandatory part of the MANOVA to handle the data correctly. For the univariate view, all Mauchly tests for sphericity are significant and were corrected to Greenhouse-Geisser:

Total Glance Time:

$F(1.227, 28.221) = 155.066$ ,  $p < .001$ ,  $\eta_p^2 = .871$  power to detect the effect was  $> .999$

Number of Glances:

$F(1.248, 28.705) = 178.046, p < .001, \eta_p^2 = .886$  power to detect the effect was  $> .999$

Single Glance Duration:

$F(1.505, 34.618) = 25.403, p < .001, \eta_p^2 = .525$  power to detect the effect was  $> .999$

All pairwise comparisons (2 s, 4 s, 8 s) for all metrics are significantly different.

Figure 3.16 holds the averaged data of the touchscreen and rotary knob trials. The data from the trials with TDRT are not included. Therefore, two data sets (touchscreen and rotary knob) for each test person ( $N = 2 \times 24 = 48$  values per data point)

When considering the regression coefficients of the lines in the middle (indetermined visualization) in Figure 3.16(a) and Figure 3.16(b), the delay duration is related to the task-related glance duration (IVIS) by a factor of approximately 0.14. For the Eyes-Off-Road Time this relation is doubled to 0.28. Part of the reasoning behind the occlusion method is to measure the total visual demand without using a driving task. Therefore, the TSOT of the occlusion method should reflect the TGT from eye-tracking. The (informational) hints of ISO 16673 (2007) to mathematically subtract delay durations, could slightly disconnect this relation, particularly when long System Response Times are present.

The regression coefficient (indetermined visualization) for the Number of Glances (Figure 3.16(c) and Figure 3.16(d)) indicates that during delays (0.24) the subjects look approximately every fourth second to the IVIS and about every 2.3 s (0.43) off-the-road.

In Figure 3.16(e) and 3.16(f) it seems that the SGD converge to a lower limit. Therefore, a logarithmic trend line is used. The higher values for the shorter delays are also influenced by the calculation of the ‘fractional’ glances, which counteracts artificially short glance durations. When browsing the glance visualizations of the delays in the online prediction tool, it is apparent that the subjects stop their glances to the IVIS after a short time, perhaps when they realize the delay. This is also visible in Figure A.7 p. 127 for a freeze delay. The ‘fractional’ calculation connects and biases the SGD of the delays with the subtasks before and after the delay. Thus, when the Number of Glances within the delay subtasks increase (e.g., for 4 s and 8 s delays) the NOG and SGD become increasingly ‘pure’ characteristics of the delay itself. Therefore, the SGD of 0.6–0.8 s for the 8 s-delays could be appropriate times for a practitioner to keep in mind for *check glances*, instead of the 300 ms proposed by the AAM guideline.

The data and heuristics may be useful in optimizing and engineering tasks for driver distraction tests. From the user experience, it is clear that intentionally freezing the system is not an option for a programmer. Also, Figure 3.16(a) and Figure 3.16(b) show that these (not communicated) delays have the highest TGTs. The TGT is likely the limiting factor for this ‘glance engineering’. For example, if a task will likely require 8 s TGT and the guideline has a limit of 12 s, there would be 4 s TGT that can be filled by (artificial) delays to decrease the SGD values and therefore increase the likelihood of



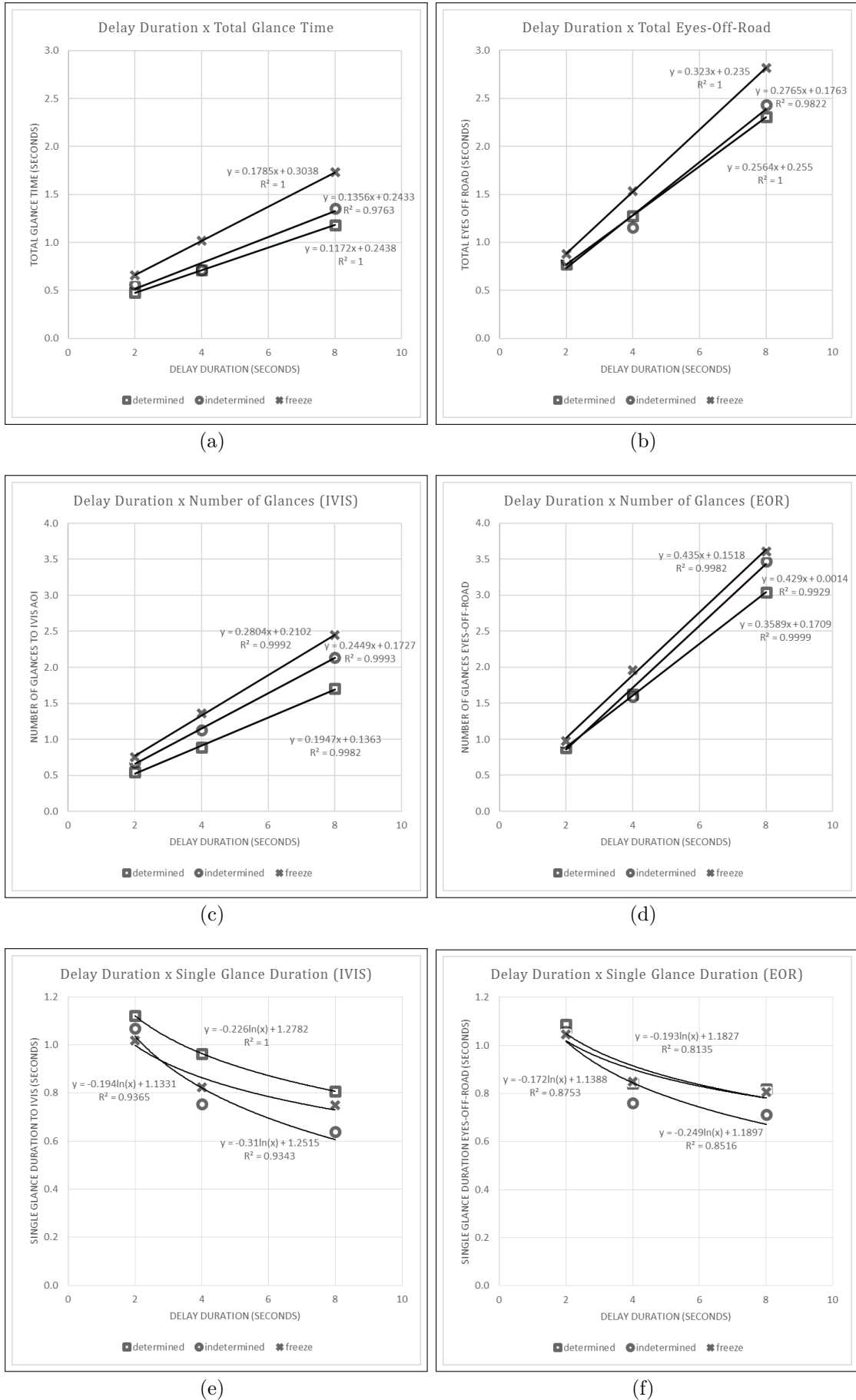


Figure 3.16.: Glance metrics during delays for IVIS (a,c,e) and Eyes-Off-Road (b,d,f)

passing SGD criteria. Figure 3.16(a) and Figure 3.16(b) demonstrate that the difference between determined and indetermined visualizations is not important regarding TGT.

Figures 3.16(c) – 3.16(f) and the previous statistical analysis led to the conclusion that an indeterminate delay results in more and shorter glances than a determined visualization. Therefore, indeterminate delays could be the choice to influence glance metrics.

In the experiment, the delays were inserted between an instruction screen and a widget to enter a number (see also p. 48). The input entered was not of interest and discarded. The delays are non-cancelable, second-level delays (i.e. *System Response Times*). During the touchscreen condition with 28 instruction screens, nine instruction screens with delays were inserted. For the rotary knob, nine of 23 instruction screens included a delay. Therefore, the potential point in time when a delay can appear should be clear for the test participants (i.e., after an instruction screen). The test persons were not specifically instructed to expect the delays. However, the training and accommodations used the same application, including the delays.

The look of the instruction screens before a delay was the same as for the 3, 5, and 10 digit input subtasks. One third of the instruction screens before a delay asked for a five digit number. Two thirds asked for a two digit number. Therefore, attentive persons could have a clue that a delay will follow when the five digit screen appears a second time during a setup and the first did not include a delay. Also, the two digit screen would be an indication that a delay will follow. This was not intentional. It was originally planned that the test subjects would work on 2, 5 and 10 digit number input tasks. This was later changed to enter 3, 5 and 10 digits without adjusting the delay instruction screens. For the tap and roll subtasks on the touchscreen, the instruction screen also displays an instruction with a two digit number and for the rotary knob the instructions are even identical (cf. Appendix C).

In sum: While there are possible cues that a delay will follow an instruction screen, it is assumed that the procedure was so complex that it was not obvious for the participants. Nevertheless, with 32% of the touchscreen instruction screen (9/28) and 39% of the rotary knob instruction screens (9/23) there is some likelihood for the participant that an instruction screen is followed by a delay. The delay duration and visualization are not foreseeable.

---

## 4. Evaluation Experiment

This chapter describes the evaluation experiment. The chapter has the following structure:

In Section 4.1, the *Hardware Setup* of the evaluation experiment is reported.

Section 4.2, *Tasks*, documents the ten tasks used in the evaluation experiment and how they were modeled before the experiment. Of these, six used the touchscreen for interaction and four the rotary knob.

Section 4.3, *Test Subjects and Procedure*, characterizes the group of test subjects and explains the experimental procedure used.

The hypotheses and issues are stated in Section 4.4, *Hypotheses and Questions*. Of interest are:

- The general performance of the prediction model (evaluation).
- If Single Glance Durations of the phone task can be lowered by inserting a System Response Time.
- If Single Glance Durations of the phone task can be lowered by using display blanking in the task (forced occlusion).
- How some metrics are changed, when the TDRT measurement is used.
- How the subject age affects a (configuration) task, that has been used before by an older subject group.
- How training and accommodation affects glance metrics of a radio-tuning task.

The postprocessing and treatment of problems with the experimental data is addressed in Section 4.5, *Postprocessing and Problems*.

Section 4.6, *Results and Discussion*, starts with a comparison of the experimental measurements of the evaluation experiment to guideline criteria (pass/fail). Afterward, the results for the hypotheses and issues stated in Section 4.4 are presented and discussed.

## 4.1. Hardware Setup

To evaluate predictions made with the tool constructed in Chapter 3, an experiment was conducted in the static driving simulator of the Institute of Ergonomics in January 2016. The Bachelor Thesis of Christina Krutzenbichler included parts of the experiment. Her thesis particularly focused on the DRT metric.

The overall laboratory situation can be seen in Figure 4.1 and Figure 4.2

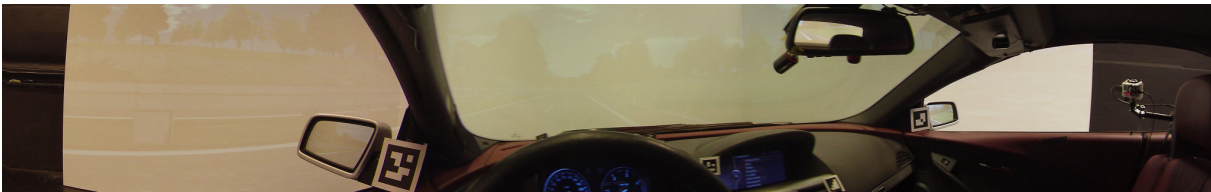


Figure 4.1.: Laboratory setup for the evaluation experiment (panorama)



Figure 4.2.: Laboratory setup for the evaluation experiment. Touchscreen tablet (in use). The on-board screen for rotary knob interactions is above. The rotary knob is visible in the lower right corner

The static driving simulator ran a SILAB 5 (WIVW GmbH, Veitshöchheim) driving simulation on six screens around the vehicle mockup. The car-following track is the same as used before (cf. Chapter 3 and descriptions of Figure 2.3 p. 22). Slight adaptations in the data recording were needed to receive triggers via a remote control to mark experimental conditions.

For touchscreen interactions, the same tablet type as in Krause et al. (2015b) was mounted in the vehicle. The Intenso Tab 824 was adjusted to 800x480 pixels. Unused display area was covered with a plastic shield to be equal to the setup of Krause et al. (2015b). The display resolution was 160 ppi.

Rotary knob tasks were performed with the hardware inside the BMW 6 convertible (E64). When the driving simulator was originally built (2009), an Adobe Flash mockup was installed by Usaneers GmbH, Munich, which mimics the original on-board IVIS (see Figure 4.2 upper screen).

Eye-tracking was achieved with the same Dikablis systems as before (cf. Chapter 3). Only the head-unit was changed to a more sensitive black/white-head-unit which normally achieved better results in the driving simulator.

The PLATO spectacles (Translucent Technologies, CA) were used again with the same Arduino control circuit Krause (2015b). Unlike in Chapter 3, the experimental results were not logged via Ethernet. The control circuit was connected to an Android tablet as USB OTG device. Therefore, the tablet powered the control circuit via USB. The tablet also received the experimental results (Total Task on Time and TSOT) via USB, which were written down by the examiner (paper & pen). To control and receive information from the Arduino control circuit the open source application Krause (2015a) was used. This application is intended to control DRT experiments. However, it has been repurposed without any change to control the occlusion experiment.

An open source Android application was implemented to send task triggers to the driving simulation and the eye-tracking system (Krause, 2016b). The application on a tablet, connected via WiFi to the simulator network and allowed the examiner to mark the current task and trial. With three buttons (start, fail and stop), appropriate signals were sent to the driving simulation and the eye-tracking system. The tablet also forwarded the current eye-tracking frame number to the driving simulation. Due to the functionality, the app was named *remote control* (rc).

For the TDRT, the same setup as in Chapter 3 was used and connected to the driving simulator network.

A camera (GoPro, Hero2) was mounted on the co-driver's seat. The main purpose was the connection to a screen, because the examiner was locally separated in the large laboratory and had no view into the car (in experimental conditions without an eye-tracking system). Nevertheless, the previously mounted camera was also used to record the experiment, which helped when doing checks in analysis

## 4.2. Tasks

For the experiment, 10 tasks were specified and modeled for use with a touchscreen (six tasks) and rotary knob (four tasks). The test subjects were also trained in two additional tasks for the overall test procedure (acclimatization to measurement methods). For the occlusion methods, the delays are not ignored (i.e., not subtracted from the TSOT); the modeling and evaluation measurement includes the System Response Times. This is different from the (informational) recommendations presented in the annex of ISO 16673 (2007).

The tasks were selected to cover both devices (touchscreen, rotary knob), with different amounts of subtasks and span a range of about 5–20s Total Glance Time. When including only a few tasks or those with very different lengths (e.g., 5s and 160s), it can be expected that correlation coefficients between prediction and measured values would become high, just due to an inappropriate evaluation setting.

Three tasks on the touchscreen were similar (entering a phone number). The difference between these were experimental factors: In one task the phone numbers were entered normally ('Phone Normal'). In another task, the phone interface had an initial System Response Time of 8s ('Phone Delay'). In the third condition, the phone interface calculated probable eyes-off-road times based on button presses at the touchscreen and intervened with a forced occlusion (display blanking) when the eyes-off-road glance grew probably too long ('Phone Blanking'). This does not represent a full factorial design as only the touchscreen phone interface is tested with these conditions (Delay, Blanking). It must be mentioned, that the display blanking (forced occlusion) is made by the touchscreen (tablet) itself. This should be not confused with the independent occlusion spectacles. The display blanking phone interface is also assessed with the occlusion glasses. In this condition (Phone Blanking), and with the occlusion glasses measurement, two independent occlusion mechanisms are operating at the same time.

### 4.2.1. Task 1, Touchscreen – ‘Config’

**Task flow:** The subjects should start the Android application Philips CarStudio from the app screen 4.3(a). 5x3 app icons are present. The app needs 1.5–2s to start and displays a splash screen 4.3(b). Within the app, the test subjects had to swipe (vertical) one time, from screen 4.3(c) to screen 4.3(d). On this screen, the button ‘Setting’ is pressed to jump to screen 4.3(e). In the settings, the subjects scroll downward (4.3(f)) and check (or uncheck) the vibration checkbox (4.3(g)). Next, they go out of the setting with the Android back button in the left lower corner (4.3(h)) and, with a second click on the same button, out of the application. The application asks: ‘Are you sure you want to close?’ (4.3(i)). The subjects click ‘Yes’ and the application needs 2–3s to close with an indeterminate indicator (4.3(j)). When screen 4.3(k) appears, the test subjects tells the examiner they are ‘done’.

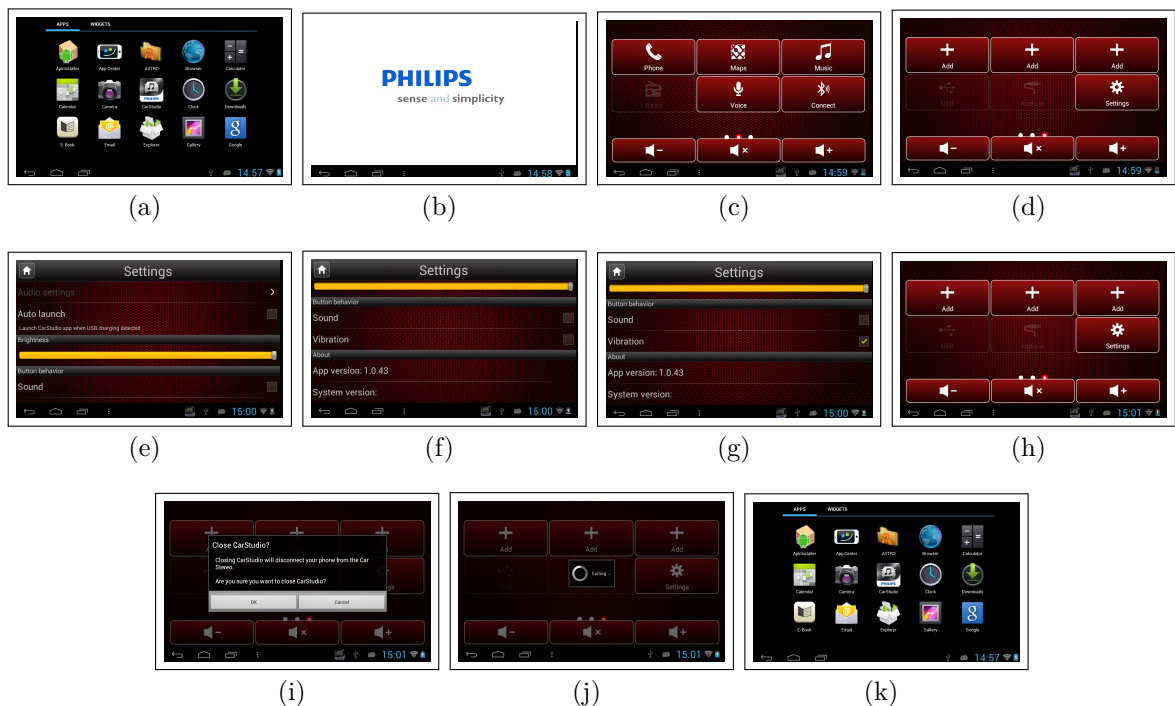


Figure 4.3.: Task flow – Task 1, Touchscreen – ‘Config’

This task was also used in two experiments in Krause et al. (2015b). This enables comparisons (see Section 4.6.5; p. 106)

**Modeling:** The start of the application was modeled with the subtask ID18 (Adjust Number Picker Tap +/-2). This subtask is the shortest active touchscreen interaction in the database and includes a short reading, an easy decision (greater/lower), two adjustment clicks and one click on an OK button. This is typically done within one occlusion cycle, respectively 1.5 TGT. This shortest subtask is afterward often used to model short interactions; therefore, it represents a kind of general-purpose helper. The delay 4.3(b), is modeled with ID7 (Touchscreen Delay No Indication 2s). Swiping and selecting the settings button 4.3(c) and 4.3(d) is mapped to ID13 (Touchscreen List Selection, first page). The subjects are trained to swipe (they know it is on the second screen) and then also know the approximate position of the settings button. This is judged to be

comparable to a search task in a list with the target on the first screen (ID13 Touchscreen List Selection first). Scrolling 4.3(f) and selecting the check box 4.3(g) is mapped to ID21 (Touchscreen Adjust Number Picker Roll +/-2). Subtask ID21 includes some scrolling on a number picker and pressing the OK button. The consecutive (fast) clicks on the back button 4.3(h) and 4.3(i) are modeled with ID18. Acknowledging the dialog 4.3(i) is again modeled with ID18. The final delay is mapped to ID4 (Touchscreen Delay Indetermined 2s).

**Subtask Model (IDs):**  $18 + 7 + 13 + 21 + 18 + 18 + 4$   
[in a digital version this is a link to the online prediction model]



## 4.2.2. Task 2, Touchscreen – ‘Radio Tuning’

### Task flow:

This task uses an open source app that closely resembles the radio-tuning task from the Driver Focus-Telematics Working Group (2006) and was tested in Krause et al. (2015a). In contrast to Krause et al. (2015a), the test subjects start not within the app, but on the app selection screen 4.4(a). Thus, the beginning of all touch tasks was the same for the test subjects: select an app icon to start the application. Within the app (4.4(b)), the test subjects had to switch to the radio mode, select the right radio band, and tune to the given frequency by repeated presses on the ‘</>’-buttons. The application shows OK when the task is finished. At this point in time the test subjects tells the examiner: ‘done’.

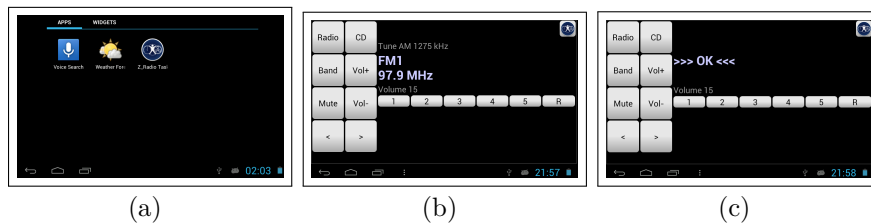


Figure 4.4.: Task flow – Task 2, Touchscreen – ‘Radio Tuning’

**Modeling:** Clicking to start the application, switching to radio mode and switching to the right radio band is ignored in the first step. It is assumed that the dominant aspect is the tuning to the correct frequency. The app randomly places the right frequency 40–44 button presses apart, so on average 42 adjustment button presses should be needed. This is modeled by using five times ID20 (Touchscreen Adjust Number Picker Tap +/-8). Subtask ID20 also includes some mental aspects (reading and a simple decision: greater/lower) and pressing an OK button. Therefore, by concatenating this subtask five times (5x8 adjustment taps), it is assumed that some of the overhead of ID20 (reading, decision, OK button) compensates for the previously neglected portions of the task modeled (start app, switch mode, select right band).

**Subtask Model (IDs):** 20 + 20 + 20 + 20 + 20  
[in a digital version this is a link to the online prediction model]

### 4.2.3. Task 3, Touchscreen – ‘Phone Normal’

**Task flow:** The following three tasks use one application. The code is open source (Krause, 2016a). The application registers three icons (white phone, black phone, blue phone) in the Android system. Every icon opens the app in another mode. The white phone is the normal phone mode for Task 3. The subjects start on the app selection screen 4.5(a) and are instructed to open the white phone app and input the given number 4.5(b). When pressing OK, the app directly closes 4.5(c) and the test person tells the examiner: ‘done’.

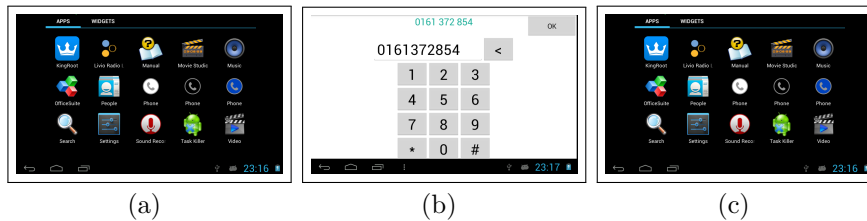


Figure 4.5.: Task flow – Task 3, Touchscreen – ‘Phone Normal’

**Modeling:** To model the opening/closing the short subtask, ID18 is used. Entering the telephone number is mapped to ID12 (Touchscreen Input Number 10 digits).

**Subtask Model (IDs):** 18 + 12

[in a digital version this is a link to the online prediction model]

### 4.2.4. Task 4, Touchscreen – ‘Phone Delay’

**Task flow:** The procedure is the same as explained for Task 3. The only difference, when opening the phone with the black phone icon for Task 4, is an initial delay of 8 s (4.5(b)) before the subject can enter the phone number.

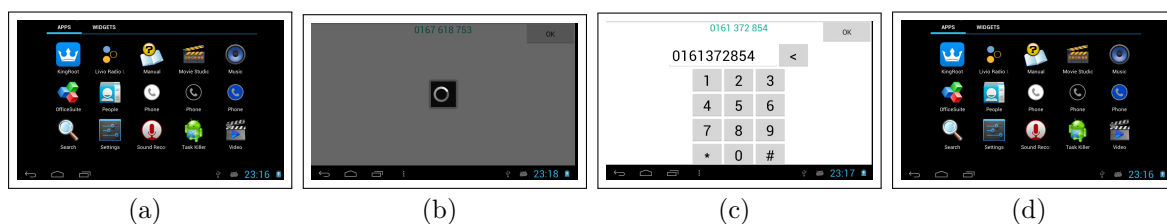


Figure 4.6.: Task flow – Task 4, Touchscreen – ‘Phone Delay’

**Modeling:** The modeling is the same as for Task 3. Additional ID6 (Touchscreen Delay Indetermined 8s) is included to model the inserted startup delay.

**Subtask Model (IDs):** 18 + 6 + 12

[in a digital version this is a link to the online prediction model]

### 4.2.5. Task 5, Touchscreen – ‘Phone Blanking’

**Task flow:** The procedure is the same as explained for Task 3. The difference, when opening the phone with the blue phone icon for Task 5, is that a display blanking occlusion mechanism is used. When an occlusion screen (display blanking) is triggered, based on the duration of the user input (see below), it hides the user interface for 1.5 s (Figure 4.7(c)). The interface (i.e. number pad) below the gray overlay would accept ongoing (blind) user interactions. It is assumed that people are forced (or motivated) to look back to the road.

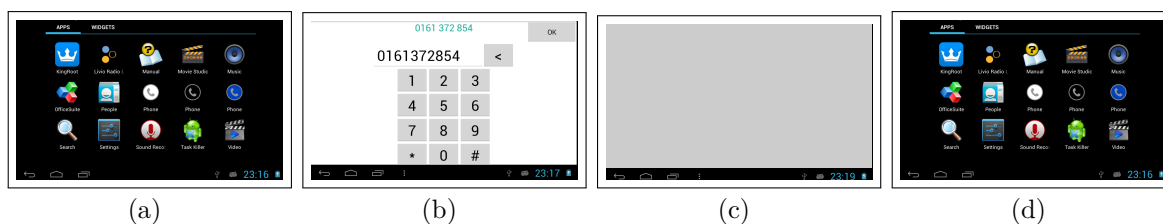


Figure 4.7.: Task flow – Task 5, Touchscreen – ‘Phone Blanking’

The algorithm for this display blanking mechanism assumes that operating a touchscreen needs visual control (eyes-off-the-road). Therefore, five assumptions (rules) are combined to calculate the current ongoing Single Glance Duration in an eyes-of-road counter (*eorSum*) based on touchscreen interactions. The rules are checked when an Android click listener is invoked on release of a button. Beside the *eorSum*-counter also the time difference (*diff*) to the last click event is calculated and used (Figure 4.8).

1. If  $\text{diff} < 0.3\text{ s}$  set  $\text{diff}$  to  $0.3\text{ s}$ . Rational: This should cap unreasonably low values.
2. If  $\text{diff} > 1\text{ s}$  set  $\text{eorSum}$  to  $0\text{ s}$ . Rational: The test subject probably looked back to the road. Therefore, reset the counter.
3. If  $\text{diff} \leq 1\text{ s}$  add  $\text{diff}$  to  $\text{eorSum}$ . Rational: The subject is likely continuously glancing away from the road.
4. If the current  $\text{diff}$  would be added a second time to  $\text{eorSum}$  and the result is above  $2\text{ s}$ , trigger the occlusion screen (display blanking). Rationale: The pace of users is different. The pace (time difference) for the current button presses is perhaps a reasonable forecast for the next button press in a continuous input task. If the next button press will raise the glance time above  $2\text{ s}$ , force an occlusion (screen blanking) directly and reset the  $\text{eorSum}$ -counter.
5. If the  $\text{eorSum}$  is  $0\text{ s}$ , set  $\text{eorSum}$  to  $0.7\text{ s}$ . Rationale: This is the first button press. A button press needs around  $0.7\text{ s}$  visual control. For instance, the average TGT is  $6.8\text{ s}$  to enter 10 digits (subtask ID 12).

A fast person (e.g., someone who enters a repdigits like ‘7777777’) can enter a maximum of five digits before a screen blanking is triggered on the fifth entry. A slower person, who enters the second digit  $651\text{--}1000\text{ ms}$  after the first digit, would trigger the display blanking on the second input. Therefore, 2–4 inputs are typically possible between display blankings.

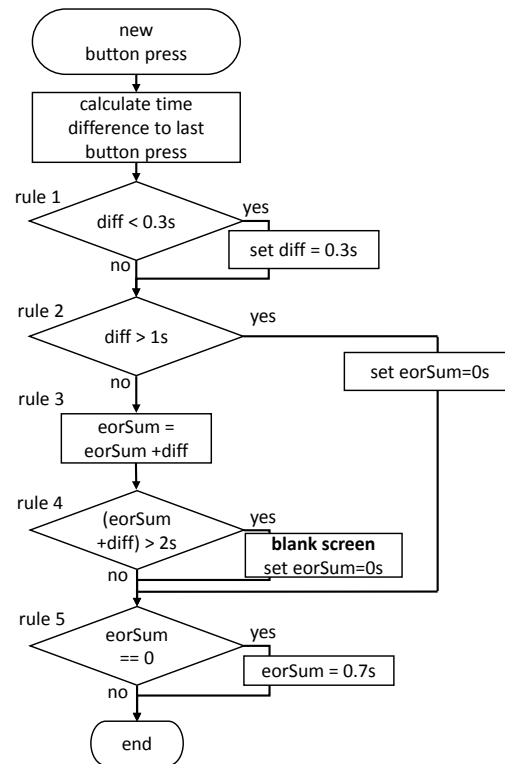


Figure 4.8.: Display Blanking Algorithm

**Modeling:** For modeling, the start of the application ID18 is used. It is assumed that the number is entered in approximately three blocks. Therefore, ID10 (Touchscreen Input Number 3 digits) should model each block. Three display blankings between/after the blocks are modeled with ID4 (Touchscreen Delay Indetermined 2s).

**Subtask Model (IDs):**  $18 + 10 + 4 + 10 + 4 + 10 + 4$   
[in a digital version this is a link to the online prediction model]

### 4.2.6. Task 6, Touchscreen – ‘Spell’

**Task flow:** The test subjects open the search application from the app screen 4.9(a). Within the application, one word with four characters is entered (not case sensitive). The search is started with ‘go’ 4.9(b). The tablet is (intentionally) not connected to the internet, so screen 4.9(c) directly appears. The test person signals to the examiner that they are ‘done’. The words cycled through and entered were: Post, Haus, Tank, Bahn, Stau, Zaun, Turm, Berg, Warm, Kalt

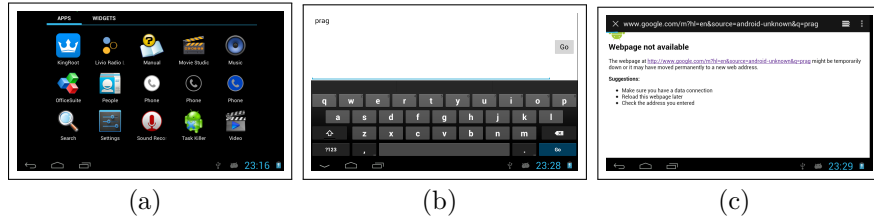


Figure 4.9.: Task flow – Task 6 Touchscreen – ‘Spell’

**Modeling:** Starting the application is again modeled by ID18. Entering the four characters is subtask ID27 (Touchscreen Input / Typing Alphabetic 4 chars).

**Subtask Model (IDs):** 18 + 27

[in a digital version this is a link to the online prediction model]

### 4.2.7. Task 7, Rotary Knob – ‘Contacts’

**Task flow:** The task starts in the main menu with the focus already on contacts 4.10(a). With a push on the rotary knob, the test subjects jump into the contacts menu 4.10(b). The menu is populated with a data set of 100 mockup contacts; generated by randomly combining common German first and last names. The test subject is instructed to search a name. The random targets are in the middle of the list (position 43–63). However, names that could potentially cause trouble (e.g., Maier, Mayer, Meyer) are omitted. When the target is selected 4.10(c), the subjects should jump into the ‘edit contact’ menu with a second press on the rotary knob 4.10(d). From this menu, the test person should jump back to the main menu with the home hardkey beside the rotary knob 4.10(e). In this state the test subjects say: ‘done’.

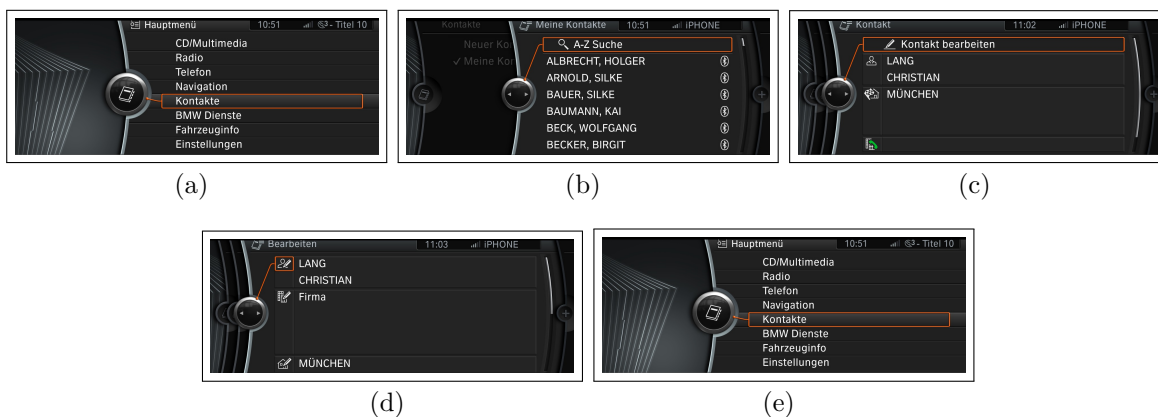


Figure 4.10.: Task flow – Task 7 Rotary Knob – ‘Contacts’

**Modeling:** The selection from the middle of the list is modeled with subtask ID45 (Rotary List Selection mid). The further steps (jump into edit menu, jump back to home screen) are modeled with ID47 (Rotary Adjust Number Picker +/-2). ID47 is the shortest interaction with the rotary knob and used like ID18 for the touchscreen interactions previously: short general purpose interaction. ID47 includes a short reading (number), a decision (lower/higher), two adjustments-indentents and a confirmation push. The typical TSOT is 1.2 s and TGT 1.5 s.

**Subtask Model (IDs):** 45 + 47

[in a digital version this is a link to the online prediction model]

### 4.2.8. Task 8, Rotary Knob – ‘Spell’

**Task flow:** The task starts on the main screen with the focus on navigation 4.11(a). With a press on the rotary knob, the test subjects jump into the navigation menu 4.11(b). With one rotary intend and a press on the knob the speller shows up 4.11(c). The test subject enters the four characters of the instructed word (Kiel, Wien, Bern, Genf, Graz, Pisa, Prag) and closes the speller with ‘OK’ 4.11(d). From this screen, the test person jumps to the main menu 4.11(d) with a hardkey and signals ‘done’.

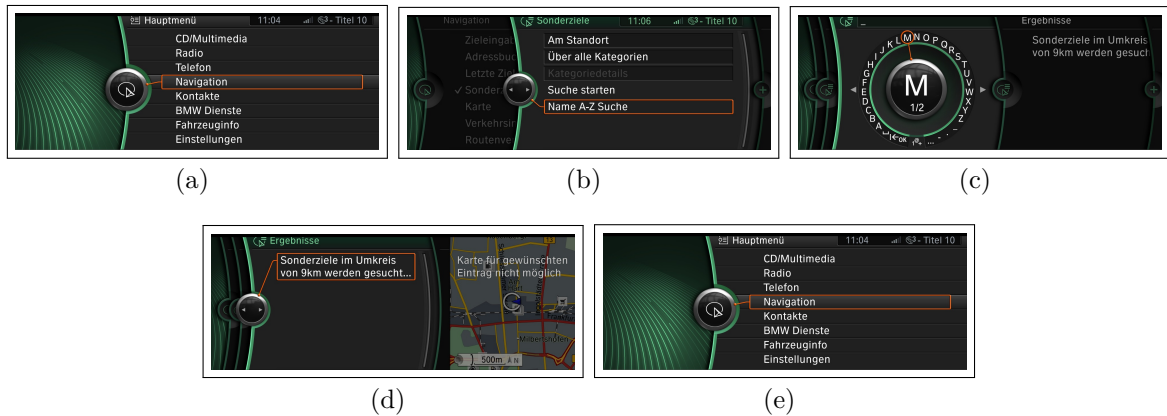


Figure 4.11.: Task flow – Task 8 Rotary Knob – ‘Spell’

**Modeling:** The input of the four characters is modeled with subtask ID42 (Rotary Input Alphabetic 4 chars). The navigation to the speller and from the speller to the main menu are both modeled with subtask ID47.

**Subtask Model (IDs):** 47 + 42 + 47

[in a digital version this is a link to the online prediction model]

### 4.2.9. Task 9, Rotary Knob – ‘Phone’

**Task flow:** Starting in the main menu with the focus on the phone option 4.12(a), the test subjects go into the phone menu and can directly enter a phone number with a dial wheel 4.12(b). A ten-digit mockup phone number (e.g., ‘0151-614-279’) in large digits with no repdigits is presented on the tablet that was used before for the touchscreen interactions (cf. hardware setup Figure 4.2). When the number is completed, the test subject presses the home hardkey and signals ‘done’ (the dialing function is not implemented in the mockup and would crash the system).

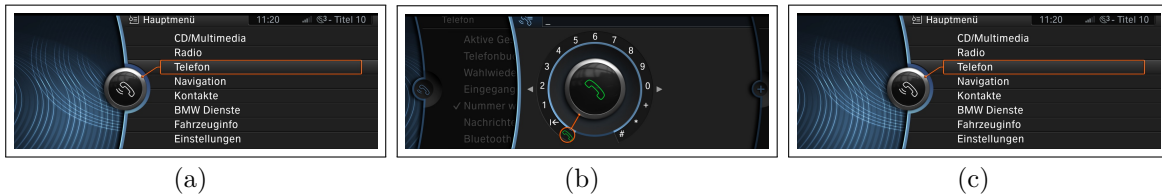


Figure 4.12.: Task flow – Task 9 Rotary Knob – ‘Phone’

**Modeling:** It is assumed that entering the 10 digits is the dominant aspect modeled with subtask ID40 (Rotary Input Number 10 digits). The initial single press on the knob is ignored.

**Subtask Model (IDs):** 40

[in a digital version this is a link to the online prediction model]



### 4.2.10. Task 10, Rotary Knob – ‘Config’

**Task flow:** The task starts in the main menu with the focus on the config option 4.13(a). The test subjects pushes the knob and goes into the config menu 4.13(b). Within the config menu, a direct second push on the knob selects the clock/date option 4.13(c). Within the clock/date option, a direct third click enables editing the date. First, the day is adjusted to an instructed day with rotary edit movements. After confirmation with a push on the knob, the month is also adjusted, and finally the year. After this editing, the test person jumps back to the main menu with a hardkey and signals ‘done’. The dates are designed so the three fields are randomly adjusted by +/-2, +/-4 and +/-8 rotary indents. (e.g., +8, +2,-4 from 04.09.2010 to 12.11.2006).

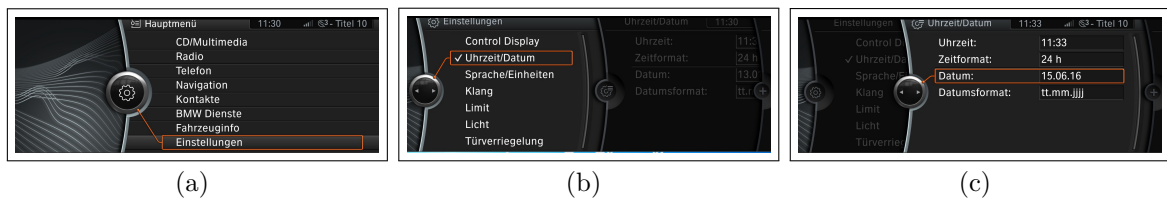


Figure 4.13.: Task flow – Task 10 Rotary Knob – ‘Config’

**Modeling:** The navigation to the date field is modeled with subtask ID47. The editing of the date is modeled with ID47 (Rotary Adjust Number Picker +/-2), ID48 (Rotary Adjust Number Picker +/-4) and ID49 (Rotary Adjust Number Picker +/-8)

**Subtask Model (IDs):** 47 + 47 + 48 + 49

[in a digital version this is a link to the online prediction model]

### 4.2.11. Acclimatization Tasks

The subjects are also trained in two additional tasks. These are only used for acclimatization to the measurement methods (e.g., occlusion setting, dual-task driving setup, DRT setup). The data of these tasks are not gathered and the tasks are not modeled. On the touchscreen, the test persons are trained to open the calculator and enter a calculation. For the rotary knob, the subjects are trained to go into the radio menu and change from one preset to another.

## 4.3. Test Subjects and Procedure

The age of the subjects was 20–26 years (median 23 years). All drivers had a driving license; for Germany very typically issued at the age of 17–18. Therefore, the years of driver experience had a median of 5.5 years. The annual mileage can be seen in Figure 4.14

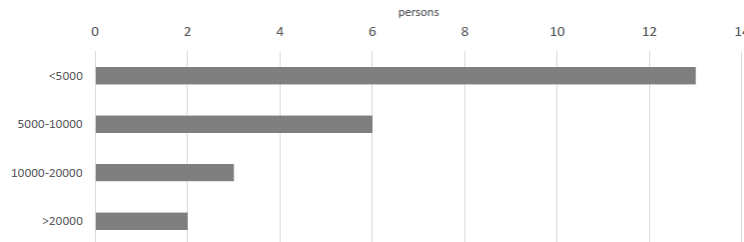


Figure 4.14.: Mileage

Out of the 24 subjects, 13 were male (54%). Two persons drove with contact lenses and five drove with glasses; no one had a known red-green blindness. Three subjects were left-handed. The simulated vehicle had automatic gears; 12 persons had no previous experience with an automatic car. A driving simulation had not been driven before by 17 people. One person was classified as an extensive simulation driver (eight experiments).

The frequency of touchscreen usage for different devices is queried with a five-point Likert scale (never–often). Most experience stems from mobile phones; 21 persons chose the rightmost option, one person one scale point below. One person had no experience with touchscreen devices. For one person, the usage questionnaire was incidentally omitted. Ten persons had no previous experience with a rotary knob.

Test subjects who participated in the previous experiment (subtask experiment to construct the prediction tool) were not intended to participate again. This was checked by asking the persons before inviting them to the experiment. Unfortunately, when anonymizing the folders of both experiments, it became clear that two subjects from the first experiment also participated in the second experiment (consent signature). This was detected after the anonymizing (scrambling), therefore it was not possible to exclude the two datasets. Therefore, the evaluation inadvertently includes an 8% retest (two persons).

The participation was voluntary, with compensation of 15 Euro.

### Procedure

Subjects signed a consent form regarding voluntary participation. They were informed that they could quit at any time without any justification and that the eye-tracking system records video and audio. Also a statement clarified that the subject is not judged and only that the system and situations would be assessed. Printed instructions were presented at the beginning (cf. Appendix B). The subjects drove the driving simulation for at least two minutes or until feeling comfortable. The examiner gave feedback about the vehicle-to-vehicle distance. The subjects then drove another 2.5 minutes without any

secondary task. The last 90 s of this measurement were later used to calculate some baseline driving metrics.

The examiner demonstrated the tasks (Section 4.2) for each subject. After this instruction, the subjects executed the tasks alone when the examiner announced a task via microphone. This also trained the later experimental situation. The task and the timing began when the examiner ended the instructions. Always the wording ‘... *und bitte*’ (start please) was used. When the final state of the task was reached, the subjects ended the task with the announcement ‘*fertig*’ (done). On these two key words (please and done), the examiner operates time taking or trigger remote controls. The task training is ended by operating the two acclimatization tasks (touchscreen calculator and changing a radio preset with a rotary knob) while driving to give the test subjects the opportunity to gain an initial understanding of dual-task settings.

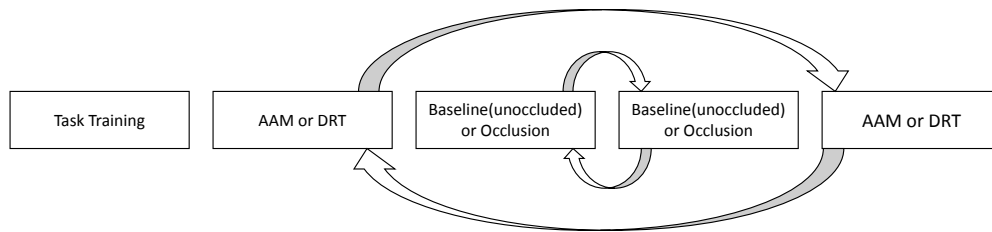


Figure 4.15.: Experimental Procedure

The experiment itself has four sections (Figure 4.15), similar to the subtask experiment before (see Section 3.4)

- *AAM*. Operating the application while driving (with eye-tracking)
- *Baseline*. Operating the application without driving. The measured Total Task on Time is, e.g., required in occlusion calculations
- *Occlusion*. Operating the application with occlusion glasses
- *DRT*. Operating the application while driving and with a Detection Response Task

The six touch tasks and the four rotary knob tasks were grouped within these sections. The order of these two task groups (touchscreen or rotary knob) was randomized. Also, within the groups, the tasks were operated in random order. The AAM and DRT sections were randomly the first or the last section within an experiment. This was intentional in order to obtain eye-tracking results for the radio-tuning task at the beginning, and to assess training effects near the end, of each session. In the middle, Baseline and Occlusion sections changed order randomly.

For all sections, the tasks were performed twice directly after one another.

#### **AAM**

In this condition, the application was operated while driving. The glance behavior was recorded with the head-mounted eye-tracking.

#### **Baseline**

This is the most obvious condition, without driving, DRT or occlusion

#### **Occlusion**

The participants were instructed for the occlusion (Appendix B). Before the section, persons were accommodated to the occlusion setup with accommodation tasks.

#### **DRT**

In addition to the setup in the AAM condition (including the eye-tracking), the condition used a Tactile-DRT setup. The subjects were instructed for the DRT (Appendix B).

To assess possible training effects (Issue 5 – Section 4.4), the radio-tuning task was performed additionally before the actual DRT section without DRT measurement.

A baseline reaction time without driving was recorded for one minute (about 15 stimuli), subsequently termed the *static DRT baseline*. This is a single-task setting (TDRT only). For another minute, a second baseline was recorded while driving: *dynamic DRT baseline*. This can be seen as a dual-task setup (driving and TDRT). Acclimatization tasks were used to accommodate the test persons to the triple-task setting (driving, secondary task and TDRT).

The ten tasks were then recorded with DRT (including the radio-tuning task).

Whenever one of the telephone tasks (Chapter 4.2, Task 3,4 and 5) had been carried out by the test subjects, the examiner verbally asked (after the second trail) verbally for a subjective rating of this interaction. For easy rating, the German school rating scheme with six numbers was used (1–6; very good – insufficient).

The duration of the experiment per person was approximately 90 minutes.

## 4.4. Hypotheses and Questions

The hypotheses and research questions for the evaluation experiment:

### Issue 1 – Predictive Quality of the Model

The modeling of the tasks has been specified in Section 4.2. The predictions of the model (developed in Chapter 3) for these tasks are compared to the results of the evaluation experiment. The metrics and tolerances are mentioned in the fundamentals Section 2.4: Typically  $\pm 20\%$  and for higher percentiles a relaxed criterion of  $\pm 40\%$  perhaps is acceptable. The overall modeling quality (correlations) is also of interest. The 13 metrics (see Section 2.5 p. 39) of the prediction model are assessed. Section 4.6.2 contains the results and discussion.

### Hypothesis 2a – Effect of System Response Time on Single Glance Duration

The task with the artificial 8 s startup delay (*Task 4, Touchscreen ‘Phone Delay’*) will need significantly lower Single Glance Durations compared to *Task 3, Touchscreen ‘Phone Normal’*. The metric of interest is the average task-related SGD (AOI IVIS). The analysis will use a paired t-test (one-tailed).

### Hypothesis 2b – Effect of Display Blanking on Single Glance Duration

The phone with forced occlusion (*Task 5, Touchscreen ‘Phone Blanking’*) will need significantly lower Single Glance Durations compared to *Task 3, Touchscreen ‘Phone Normal’*. The metric of interest is the average task-related SGD (AOI IVIS). The analysis will use a paired t-test (one-tailed). Furthermore, the subjective rating of the three phone tasks is reported.

Section 4.6.3 contains the results and discussion.

### Issue 3 – Metrics With and Without TDRT

The glance metrics described in Section 3.6 were slightly higher during normal subtasks and not statistically different during delays. A MANOVA for the dependent variables TGT, SGD and DLP is calculated. The independent variables (2x10) are the experimental condition (without TDRT / with TDRT) and the ten tasks. Section 4.6.4 contains the results and discussion.

### Hypotheses 4 – Age Effects

*Task 1, Touchscreen ‘Config’* is a retest and was already involved twice in former experiments (Krause et al., 2015b). Therefore, the metrics TGT (AOI IVIS), SGD (AOI IVIS) and TSOT are assessed (dependent variables) with a MANOVA between these three experiments (independent variable). From the descriptive results (Section 3.6), it is expected that middle-aged people (45–65 years) from the two former experiments required higher TSOT, the TGT should be approximately comparable, and the SGD is expected to be higher for the young subjects (20–35 years). Section 4.6.5 contains the results and discussion.

### **Issue 5 – Training/Accommodation Effects**

The Android radio-tuning task (*Task 2, Touchscreen ‘Radio Tuning’*) was already used in Krause et al. (2015a). In Krause et al. (2015a), the task needed surprisingly high task-related SGDs compared to another experiment with the same task. Therefore, the study assumed a possible reason could be that another task, which was involved in the experiment for acclimatization (entering a number on a touch number pad), caused this by carry-over effects. These long SGD strategies are typical for virtual keypads.

An alternative explanation or influence: The test subjects in Krause et al. (2015a) used the radio-tuning task continuously, in different experimental condition, for approximately one hour. Perhaps familiarization motivates longer glances for this task (feeling safe). To check how training and accommodation effects can affect glance metrics, the Android radio-tuning task is conducted two times in the current assessment of each test subject: early and late during the experiment. The glance metrics TGT and SGD (independent variables) are checked in a repeated-measurement MANOVA for the two points in time (dependent variable). The counterintuitive hypothesis to explain the surprising outcome of Krause et al. (2015a) would be that familiarization can lead to longer glances. Section 4.6.6 contains the results and discussion.

## 4.5. Postprocessing and Problems

### Eye-tracking Data

The eye-tracking data were manually inspected to maximize the pupil detection and adjust, e.g., shifted headunits. The Dikablis Analysis tool of the D-Lab software suite was used for this purpose. Within D-Lab, three AOI were defined for each subject: Driving Scene, Speedometer, IVIS (Figure 4.16).

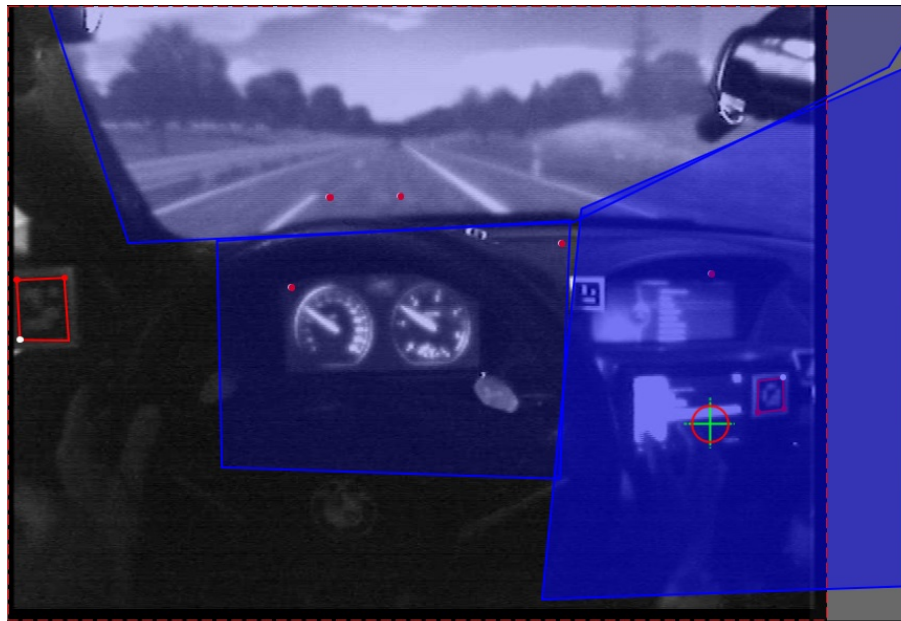


Figure 4.16.: Areas of Interest (Driving Scene, Speedometer, IVIS) in D-Lab

The glance data were post-processed with D-Lab (Basis Version 2.0 Feature 2.1; Ergoneers GmbH) default options: 120 ms blink removal and 120 ms cross through glance handling. The further workflow for the glance data is similar to Section 3.4 p. 55.

Despite thorough checks before the experiments, the markers (i.e. landmarks, 2D-codes) of the eye-tracking system were unreliably recognized. This would have rendered the eye-tracking data useless. To circumvent the problem, an open source helper tool has been implemented (Krause, 2016c). The tool uses template matching (from the openCV library) to track landmarks. The landmarks can be selected via drag-&-drop in the video pictures. It is advisable to choose characteristic and contrasting spots that are similar for every participant (e.g., the speedometer or distinct edges between several displays). Six of these spots have been tracked by the tool for all participants to support the AOI positioning (red dots in Figure 4.16).

### Occlusion and Baseline (Unoccluded) Data

The examiner gave the instructions for a task (shutter closed) and then started the occlusion cycling and timing (shutter open). After the test person finished (verbal indicator ‘done’), the Total Task Time shown by the hardware was recorded by the examiner for each trial. During the baseline trials, the occlusion hardware was also used for time taking; however, the occlusion glasses were not worn and out of sight of the test subject to

obtain  $TTT_{unoccluded}$ . From the stopped  $TTT_{occluded}$  the TSOT was calculated later on with an Excel formula (cf. Krause et al., 2015c; ISO 16673, 2007):

$$TSOT = (1.5s * DIV(TTT_{occluded}, 3s)) + MIN(MOD(TTT_{occluded}, 3s); 1.5s)$$

The first term covers full cycles of 3 s (open/closed); the last term handles the partial (unfinished) cycle (see Figure 4.17).

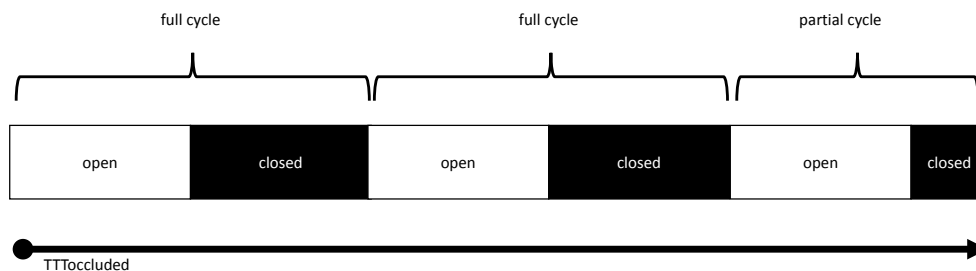


Figure 4.17.: Calculation of TSOT from  $TTT_{occluded}$

The two trials of each subjects were averaged (separately  $TTT_{unoccluded}$  and TSOT). At that point, the R-metric was calculated for each subject and each task:

$$R = TSOT / TTT_{unoccluded}$$

### Driving Data

The driving track records the distance to the leading vehicle. However, only if the COG of the simulated car (ego-car) is within the intended lane. In rare cases, the COG crosses the lane boundary for a short time and the following headway calculation returns zero. The Drift of Following Headway (DFH) is based on differences (differentiation). These rare drop-outs could have a huge impact. Therefore, small gaps in following headway are filled by linear interpolation<sup>1</sup>.

### Detection Response Task Data

Upon initial consideration, Figure 4.18, containing the reaction times of all subjects during driving and operating tasks, appears normal. There is a similar discontinuity as described previously (p. 57) at around 1500 ms. An in-depth inspection revealed that one person continuously responded after 1 s; also in the single-task setup (TDRT only) and in the dual-task setting (TDRT and driving). During the experiment, the subject responded to only 37% of the stimuli (hit rate). That is approximately every third stimulus. This person was excluded from the TDRT analysis.

<sup>1</sup>Matlab function *inpaint\_nans()* by John D’Errico 2009. release 2 release date 4/16/06



In a former experiment (Krause et al., 2014a), one person regularly responded at around 1 s after stimulus onset. This was interpreted as the subject unconsciously waiting 1 s for the second stimulus (vibration switch-off after 1 s). Therefore, the subject changed or misunderstood the task. In both experiments, these artifacts were not detected by the examiners during the experiment. In Krause et al. (2014a), the person also sometimes altered her behavior during the experiment. A hint or note for the examiners (or at least for the data analysts) to be aware of these artifacts could be helpful in ISO/DIS 17488 (2014).

The long tail in Figure 4.18 could be an indication that the switch-off of the stimulus is sometimes a kind of unintended reminder, also for other subjects. A possible improvement for DRTs might be to fade out stimuli instead of switch-off after 1 s.

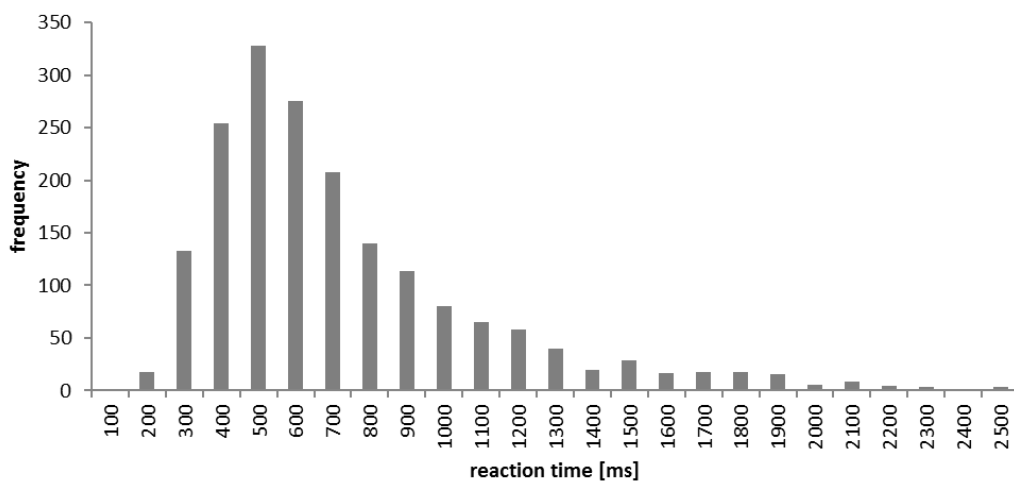


Figure 4.18.: Histogram of TDRT reaction times (hits only) of all subjects

To check for cheating strategies (e.g., repeatedly pressing the button), a button-press ratio has been calculated (button presses divided by count of stimuli). The ratio has been calculated over the whole experimental DRT condition, including times between tasks (i.e., new instructions from the examiner). For the 23 subjects (excluded one subject), the minimum ratio is 0.6 and the maximum 1.32 (average 0.9, SD 0.15). The inter-quartile range (Q1–Q3) is 0.77–1.01. Therefore, no indications for continuous cheating strategies of test subjects were found.

The hit rate of the 23 subjects is 74%–100% when operating tasks. The inter-quartile range (Q1–Q3) is 85%–97%. Therefore, most subjects (except the excluded one) were able to and engaged in work on the TDRT.

## General

For one subject (VP18), the second trial of Task 2 and Task 7 in one experimental condition (AAM) is not available due to technical reasons. When averaging is employed, these two data are based on one trial.

To calculate 85<sup>th</sup> percentiles (e.g., AAM) the interpolating Excel 2013 (V 15.0.4859.1000 64bit) function quantile (p=0.85) is used. For the NHTSA 85<sup>th</sup> percentile checks, the calculations described and clarified in NHTSA (2014) are used.

## 4.6. Results and Discussion

For this thesis, the alpha level in statistics is set to 5% if not indicated otherwise (e.g., Bonferroni correction). The alpha level of 5% is also used when calculating observed powers. Before presenting and discussing the issues and hypotheses stated in Section 4.4, the measurement results are compared to guideline criteria and a pass/fail overview is provided:

### 4.6.1. Pass/Fail Overview

Table 4.1 compares the measurements to several criteria of AAM and NHTSA guidelines (pass/fail). The procedure and subject sampling for the evaluation experiment are not in full accordance with these guidelines. Therefore, the pass/fail results should be seen as informational.

	AAM TSOT P85 15s	AAM TGT P85 20s	AAM SGD P85 2s	NHTSA TSOT 12s	NHTSA TEORT 12s Trial1	NHTSA TEORT 12s Trial2	NHTSA Mean SGD Trial1	NHTSA Mean SGD Trial2	NHTSA P85 SGD Trial1	NHTSA P85 SGD Trial2
Task1	ok (11.3 s)	ok (10.7 s)	ok (1.59 s)	ok 23/24	FAIL 19/24	ok 23/24	ok 23/24	ok 24/24	ok 22/24	ok 22/24
Task2	ok (14.9 s)	ok (16.6 s)	FAIL (2.25 s)	FAIL 8/23	FAIL 2/24	FAIL 2/24	FAIL 19/24	FAIL 20/24	FAIL 12/24	FAIL 15/24
Task3	ok (9.9 s)	ok (9.1 s)	FAIL (3.48 s)	ok 24/24	ok 23/24	ok 23/24	FAIL 17/24	FAIL 18/24	FAIL 15/24	FAIL 15/24
Task4	ok (13.5 s)	ok (11.0 s)	ok (1.77 s)	FAIL 13/24	FAIL 15/24	FAIL 16/24	ok 23/24	FAIL 20/24	FAIL 20/24	FAIL 19/24
Task5	ok (11.8 s)	ok (11.0 s)	FAIL (2.36 s)	ok 24/24	FAIL 17/24	ok 21/24	FAIL 19/24	FAIL 19/24	FAIL 11/24	FAIL 13/24
Task6	ok (7.2 s)	ok (6.7 s)	FAIL (3.98 s)	ok 24/24	ok 24/24	ok 24/24	FAIL 16/24	FAIL 15/24	FAIL 19/24	FAIL 17/24
Task7	ok (13.9 s)	ok (13.8 s)	ok (1.64 s)	FAIL 15/24	FAIL 13/24	FAIL 17/24	ok 23/24	ok 22/24	FAIL 19/24	ok 22/24
Task8	ok (14.7 s)	ok (14.7 s)	ok (1.96 s)	FAIL 6/24	FAIL 5/24	FAIL 8/24	ok 21/24	FAIL 20/24	FAIL 12/24	FAIL 13/24
Task9	FAIL (18.5 s)	FAIL (20.8 s)	FAIL (2.46 s)	FAIL 0/24	FAIL 1/24	FAIL 1/24	FAIL 18/24	FAIL 16/24	FAIL 10/24	FAIL 11/24
Task10	ok (10.5 s)	ok (11.5 s)	ok (1.76 s)	ok 23/24	FAIL 17/24	ok 21/24	FAIL 20/24	ok 21/24	FAIL 16/24	FAIL 17/24

Table 4.1.: Criteria – Measurement Pass/Fail Overview

The *AAM TSOT P85 15s* column compares the average of two occlusion trails to the 15s criteria from the AAM guideline. The values in parentheses show the 85<sup>th</sup> percentiles, calculated with the interpolating Excel function (quantile 0.85). Similarly the *AAM TGT P85 20s* column is based on eye-tracking results.

In the *AAM SGD P85 2s* column the average SGD of two trials (based on the fractional approach) is compared to the 2s criteria.

*NHTSA TSOT 12s* is similar to the *AAM TSOT* column previously mentioned. However, it uses the ‘at least 21 of the 24 test participants’ calculations of the NHTSA guideline. These x of 24 subjects is reported in the cells (e.g., ok 21/24). Interestingly, the original NHTSA occlusion procedure would average five trials, while the NHTSA eye-tracking is based on a single trial. Therefore, the NHTSA eye-tracking criteria are separately applied to the first and second trial of the eye-tracking data to gain more insight. These reported eye-tracking metrics are all based on the eyes-off-road approach and only use full glances (no fractional glances) when calculating SGDs (TEORT/NOG). The NHTSA

guideline has two 2 s SGD criteria. The P85 criterion (NHTSA, 2014, VI.E.14.a) is more complex and requires an intermediate step: For each participant the allowable number of long glances, based on the NOG, needs to be calculated. Afterward, pass or fail rates can be checked.

Two remarkable issues are shortly mentioned:

The TSOT and TGT values (AAM) are very similar, so the criteria difference (TSOT 15 s and TGT 20 s) seems inadequate. With the assumption that middle-aged people have even longer TSOTs (see Hypothesis 4, Section 4.6.4), this difference should be even more anomalous—in other words, the occlusion method seems unreasonably disadvantaged in the AAM guideline.

In particular, the touchscreen tasks (Task 1–7) have poor SGD results (e.g., *AAM SGD P85 2s* column), except those with System Response Times (Task 1 & 4).

## 4.6.2. Issue 1 – Predictive Quality of the Model

### Results

Metric	Unit	MAE	Pearson r	MAPE	CntError <10%	CntError <20%	CntError <40%
TTT unoccluded	s	3.51	.811	23.6%	3	4	9
TSOT	s	2.16	.814	19.8%	4	5	9
R	-	0.07	.649	9.3%	6	9	10
TTT while driving	s	2.37	.875	13.0%	4	8	10
TGT IVIS	s	1.53	.878	15.1%	3	7	10
NOG IVIS	-	0.83	.904	11.4%	6	8	10
SGD IVIS	s	0.18	.519	11.6%	5	8	10
TEORT	s	1.76	.865	15.4%	3	7	10
NOG eyes-off-road	-	1.30	.890	17.5%	1	6	10
SGD eyes-off-road	s	0.20	.490	14.4%	5	6	10
DRT deterioration	%	22p.p.	.843	25.2%	0	3	10
DLP deterioration	%	23p.p.	.724	19.3%	4	7	9
DFH deterioration	%	20p.p.	-.232	55.7%	0	1	4
TSOT P85	s	2.42	.795	19.2%	3	4	9
TGT IVIS P85	s	1.56	.897	12.5%	4	8	10
SGD IVIS P85	s	0.45	.704	18.3%	3	6	9
TEORT P85	s	1.57	.905	11.7%	5	9	10
SGD eyes-off-road P85	s	0.26	.551	14.5%	4	8	9

Table 4.2.: Evaluation overview

Table 4.2 presents an overview of the evaluation results. More details and plots for each metric can be found in Appendix D. The upper part of the table holds the evaluation of 13 metrics, based on predicted and measured medians. The lower part displays some additional information, when evaluating the 85<sup>th</sup> percentile (P85) for some metrics. The mean absolute error (MAE) column reports the average error compared to the prediction. *Pearson’s r* holds the correlation between prediction and measurement ( $N = 10$  tasks). The mean absolute percentage error (MAPE) is presented in the next column. The last three columns  $CntError < x\%$  present how often the percentage between prediction and measurement was below  $x\%$ . Therefore, a fast increasing number (in the 10% and 20% column) is desirable; the maximum achievable is ten (tasks).

The TTT unoccluded and TSOT results are based on averaging two trials for each person. For R the averaged TTT unoccluded and TSOT is divided. The TTT while driving, TGT (to IVIS) and NOG IVIS (fractional) are averaged results of two trials during AAM testing. For SGD IVIS, the SGD for each trial is calculated based on the fractional approach (TGT/NOG) and then averaged. The TEORT, NOG (eyes-off-road) and SGD (eyes-off-road) are also two averaged trials. The NOG (eyes-off-road) and SGD (eyes-off-road TEORT/NOG) are not based on the fractional approach. To measure the DRT deterioration, the median reaction time of two trials is calculated separately,

then averaged and related to the median baseline reaction time (driving with TDRT). Regarding the DLP and DFH, deterioration the driving performance during AAM testing of two trials is averaged and related to baseline driving.

The 85<sup>th</sup> percentiles (P85) of the measurements are calculated with the interpolating Excel function (quantile 0.85).

The tasks and modeling for the predictions are documented in Section 4.2 (p. 74). While some tasks are modeled in a relatively complex manner, Task 3 and Task 9 are mapped to basic subtasks (enter a phone number on a touchscreen and enter a phone number with a rotary knob). Therefore, these two tasks can be also used as a kind of retest check and reference; the detailed results are reported in Appendix D. When considering these detailed tables, it is also advisable to keep in mind that the first six tasks (Task 1 – Task 6) are touchscreen tasks, and Task 7 – Task 10 are rotary knob tasks. Task 3, Task 4 and Task 5 are essentially the same task (entering a phone number) with specific modifications.

### Discussion

For the TTT unoccluded the MAPE (23.6%) would be slightly above the accepted 20% limit (cf. Section 2.4). A deeper examination of the results (Appendix D.1) reveals that the difference primarily originates from the rotary knob tasks. In general, the (static) TTT unoccluded is not too important for driver distraction assessments. While the main difference for the TSOT (MAPE 19.8%) still stems from differences in the rotary knob predictions (Appendix D.2), this underestimation is diminished for the dynamic TTT while driving (MAPE 13.4%; Appendix D.4), TGT (MAPE 15.1%; Appendix D.5) and TEORT (MAPE 15.4%; Appendix D.8). The reason for the surprisingly slow performance (TTT unoccluded and TSOT) of the subjects in rotary knob tasks is unclear. The congruency for TTT while driving indicates that the test subjects would be able to perform similar to the subjects in the subtask database under the given experimental conditions. A possible explanation could be that the subjects had chosen an individually slower user pace on the rotary knob for TTT unoccluded and TSOT. The additional driving task may accelerate the user pace and render it similar to the subtask database. Therefore, the driving task might have a beneficial experimental impact by interacting with the user pace and diminishing differences between experiments. The (static) TTT unoccluded seems surprisingly to be one of the hardest metrics to predict. The R-metric benefits from the cancellation of the user pace by the division (TSOT/TTT unoccluded). The user pace affects the numerator and denominator.

In addition, the R-metric and the Single Glance Durations have typical ranges (e.g., R: 0.7–1; SGD: 1–2 s). These also limit deviations in MAE and MAPE.

When considering the *CntError*<10% column of the NOG IVIS, it is visible that six tasks had been predicted with a deviation <10% (all touchscreen tasks; Appendix D.6). The NOG eyes-off-road seems harder to predict. A short check had been carried out: In the glance visualization of the online interface of the model, it can be recorded that during the 10-digit input subtask on touchscreen, two speedometer glances were registered (for all 24 subjects). In the comparable evaluation of Task 3, the subjects together glanced 16 times at the speedometer during the first trial of the touchscreen phone input. In the second trial, 16 glances are also observed. More (short) unpredicted glances lower

the SGDs (eyes-off-road). This could be a reasonable explanation concerning why the SGD eyes-off-road are over-predicted (Appendix D.10). This indicates that eyes-off-road metrics can make experiments more susceptible to disturbances and can lead to counter-intuitive results: For example, the SGD IVIS P85 (Appendix D.14.3) for Task 3 is about 3.5 s, while the SGD eyes-off-road P85 (Appendix D.14.5) would report 2.3 s. Therefore, the glances to the IVIS appear longer than the glances away from the road.

The evaluation results for the 85<sup>th</sup> percentiles (P85) in the final part of Table 4.2 appear not worse (Pearson's  $r$  and MAPE) than the predictions of the median in the upper part of the table. It is questionable whether the relaxed acceptance criterion of 40% for higher percentiles (cf. Section 2.4) is actually necessary.

The DRT, DLP and DFH deteriorations display considerable variability (Appendices D.11, D.12 and D.13).

Predictions of the DFH deterioration are unacceptable: MAPE 55.7%, weak correlation ( $r = -0.23$ ) and only four tasks could be predicted with  $< 40\%$  deviation.

The DRT predictions are slightly beyond the acceptance criterion (MAPE 25.2%). Seven tasks deviate 20–40% (difference of the last two columns). The high correlation ( $r = 0.84$ ) would be a benefit. A closer investigation of the detailed table (Appendix D.11) reveals that all tasks were under-predicted. This explains why a high correlation, combined with an unfortunate error (MAPE), can be observed. The reasons for the offset are unclear.

The online interface of the prediction model also includes bootstrap indicators. The model bootstraps a sample of 24 persons from the  $N = 24$  subjects 1,000 times. These bootstrapped data sets are compared to guideline criteria. Based on this result, an indicator is calculated as a percentage related to how often the result passed the criteria. The comparison of the indicators to the measurement outcomes can be found in Appendix D. The indicators for TSOT, TGT and TEORT appear valuable, while it must still be kept in mind that the model should present an approximate idea and estimation. For SGD IVIS (AAM), the indication can be helpful. The mean SGD eyes-off-road (NHTSA) indicator is questionable. This can be also due to the potential unreliable eyes-off-road metric.

The DFH bootstrap indicator is based on a likely less reliable metric and is therefore judged useless. The DLP bootstrap indicator demonstrates a positive performance (Appendix D.14.6), with a correlation of -0.75. Nevertheless, the DLP also includes one of the worst predictions: Task 4 (Phone Delay) with an over-prediction of 74% (Appendix D.12). The delay subtasks were measured embedded in a complex application (see construction of the prediction model, Section 3.2). It is possible that some subjects used longer delays to adjust their lane positions, which can cause higher DLP values. However, these DLP values are still lower than typical visual/manual subtask interactions (e.g., dialing a number). In the evaluation of Task 4, the delay is at the beginning of the task. At the beginning, there should be no reason to make larger adjustments to the lane position. This probably resulted in the very low measured DLP values during the evaluation experiment. This is an indication that the position and combination (order) of subtasks within the prediction might be important sometimes. Measuring and storing all of these (hidden) potential interdependences between subtask combinations appears hardly possible. For

the current model, the order of the subtasks is neglected.

The selection of subtasks to model a task is, to some extent subjective. This topic is not assessed in this evaluation and thesis. It is foreseeable that different persons may chose slightly different subtasks. The description of the modeling for the ten tasks (Section 4.2) may help to find a reasonable mapping. It must be also kept in mind that the model handles visual/manual interfaces. When a delay (e.g., after manually dialing a phone number) is ended with an acoustic event (e.g., a ringing tone) this is a mixture of visual/manual and auditory interfaces that cannot be predicted with the current model.

Overall, the model makes generally reasonable predictions. The aim to offer approximate estimates for prototypes is achieved. The DFH deterioration metric should be ignored, disabled or hidden in a future version of the online interface. When DFH is excluded, the mean coefficient of determination of Table 4.2 would be  $R^2 = .614$ . The overall average MAPE without DFH is 16% (min 9.3%, max 25.2%).

Comparing this result to some other evaluation experiments, already reported in Section 2.4, helps in judging the performance and emphasizes the distinctions to other models. With regard to  $R^2 = 0.88$  and  $R^2 = 0.92$ , Pettitt's method was evaluated in Kang et al. (2013) to model occlusion task times. Salvucci (2005) reports a fit of  $R^2 > .99$  to model the task on time while driving of four short tasks. For a TEORT prediction, Purucker et al. (2017) reports  $r = 0.58$ , which would result in  $R^2 = 0.34$ .

Compared to these results, the final outcome of this thesis ( $R^2 = .614$ ) is between the impressive high fits of approximately  $R^2 = 0.9$  and the improvable  $R^2 = 0.34$ . The other models are typically restricted to predicting one or a few metrics and usually do not provide data for, e.g., 85<sup>th</sup> percentiles. The model built in this thesis provides predictions for different assessment methods and uses distributions to derive, e.g., 85<sup>th</sup> percentiles.



### 4.6.3. Hypotheses 2 – Effects on Single Glance Durations

#### Results

After entering a telephone number on the touchscreen, the test subjects were asked for a subjective rating for this interaction with a German school grade (1–6; very good – insufficient). The interface was tested in three tasks; normal (Task 3, T3), with an initial indetermined visualized System Response Time of 8 s on startup (Task 4, T4) and with a display blanking algorithm (Task 5, T5). The algorithm blanked the display when it was continuously operated for about 1.5–2 s (cf. pp. 79 for a detailed description). These tasks were used in four experimental conditions: DRT, Baseline, Occlusion, AAM.

Figure 4.19 presents the box plots of the ratings. These results are used in the discussion together with the hypothesis test for the SGD

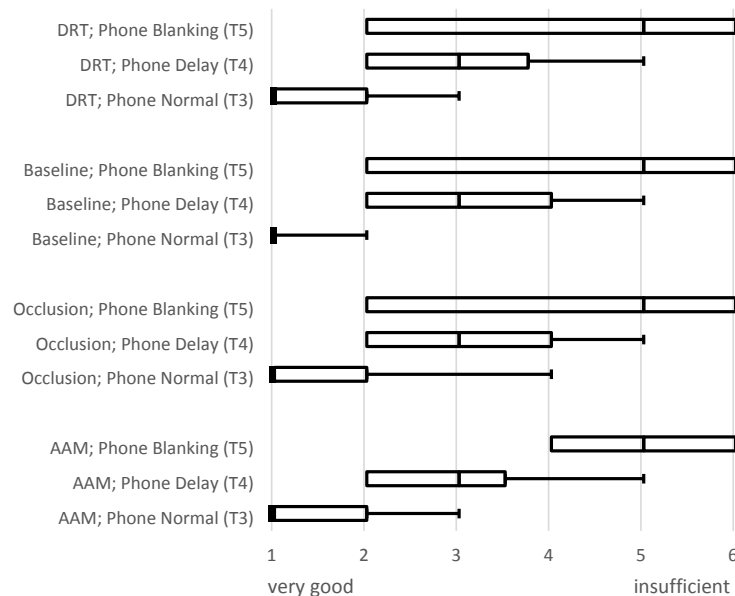


Figure 4.19.: Subjective ratings for the interactions with the phone tasks

The following analysis uses the SGD to the IVIS in the experimental condition AAM (car-following setup with eye-tracking). The mean SGD (TGT/NOG) of two trials is calculated and averaged. The fractional approach is used for the NOG. The two hypotheses (2a, 2b) are closely related and the t-tests ( $N = 24$  subjects) are conducted together. Therefore, the significance level is corrected (Bonferroni) to  $p = .025$ .

**Hypothesis 2a – Effect of System Response Time on Single Glance Duration** The mean SGD of *Task 3, Touchscreen ‘Phone Normal’* ( $M = 2.05$  s,  $SD = 1.01$  s) is **significantly reduced** in *Task 4, Touchscreen ‘Phone Delay’* ( $M = 1.46$  s,  $SD = 0.29$  s)

$$t(23) = 3.59, p < .001, r = .694 \text{ (paired t-test, one-tailed).}$$

**Hypothesis 2b – Effect of Display Blanking on Single Glance Duration** The mean SGD of *Task 3, Touchscreen ‘Phone Normal’* ( $M = 2.05$  s,  $SD = 1.01$  s) is **significantly reduced** in *Task 5, Touchscreen ‘Phone Blanking’* ( $M = 1.72$  s,  $SD = 0.37$  s)

$$t(23) = 2.56, p = .009, r = .813 \text{ (paired t-test, one-tailed).}$$

### Discussion

Figure 4.19 illustrates that the subjective ratings for the three phone tasks (Task 3, Task 4 and Task 5) is different. However, for each task the rating is nearly constant over the four experimental conditions. The normal phone task (Task 3) is rated between 1–2 (very good). The 8 s initial delay lowers this rating to about 3 (satisfactory). The display blanking (forced occlusion) receives the worst rating with 5 (deficient).

Both approaches can reduce the mean SGD significantly. The second approach (forced occlusion) is less effective and is unacceptable (subjective ratings). The medium to high correlations (.694, .813), indicate that the SGDs are to some extent an individual behavior.

When considering the pass/fail overview (Table 4.1), this reduction is also visible in the 85<sup>th</sup> percentile (AAM SGD P85 2s) column: Task 4 would easily pass, Task 5 slightly fails. While Task 3 and Task 5 would pass the TSOT 12 s limit, Task 4 would fail. This can be attributed to the handling of an 8 s delay: For tasks with long System Response Times, ISO 16673 (2007) would recommend using eye-tracking; or to subtract the delays, which was not done.

Task 4 would also fail for the TEORT. The P85 TEORT is approximately 13 s. The 12 s limit is about the 75<sup>th</sup> percentile (Q3) of Task 4. Therefore, e.g., shortening the delay slightly may help. Another approach might be to split the delay into smaller delays. While Task 3 is far from passing the NHTSA SGD criteria, Task 4 is close. It is likely that a test by a group with slightly shorter glances (e.g., incorporating middle-aged people) would pass.

The touchscreen Task 1 is the only task which surprisingly has no problems with all single glance criteria (Table 4.1). This task includes two delays of approximately 2 s when opening and closing the application. While the task includes many different screens and dialogs, these typically need only one single action (e.g., button press), which seems to support interruptibility.

#### 4.6.4. Issue 3 – Metrics With and Without TDRT

##### Results

The dependent variables (DV) are the metrics:

- TGT to the IVIS; average of two trials
- Mean SGD to the IVIS calculated by averaging the SGD of two trials ( $SGD = TGT/NOG$ ); for NOG the fractional approach is used
- Drift in Lane Position (DLP); two trials averaged

The independent variables (IV) are:

- Experimental setup (without TDRT method / with TDRT method)
- The ten tasks (Task 1 – Task 10)

This was fed into a repeated-measures MANOVA. Of interest is the effect of the experimental setup on the three metrics: Wilks'  $\lambda = .406$ ,  $F(3, 21) = 10.25$ ,  $p < .001$ ,  $\eta_p^2 = .594$  the power to detect the effect was .994. **Therefore, the setup (with/without TDRT) had a significant effect on the metrics.**

A closer look into the related uni-variate tests:

Total Glance Time:

$F(1, 23) = 22.749$ ,  $p < .001$ ,  $\eta_p^2 = .497$  the power to detect the effect was .995

Single Glance Duration:

$F(1, 23) = 12.501$ ,  $p = .002$ ,  $\eta_p^2 = .352$  the power to detect the effect was .923

Drift in Lane Position:

$F(1, 23) = .414$ ,  $p = .526$ ,  $\eta_p^2 = .018$  the power to detect the effect was .095

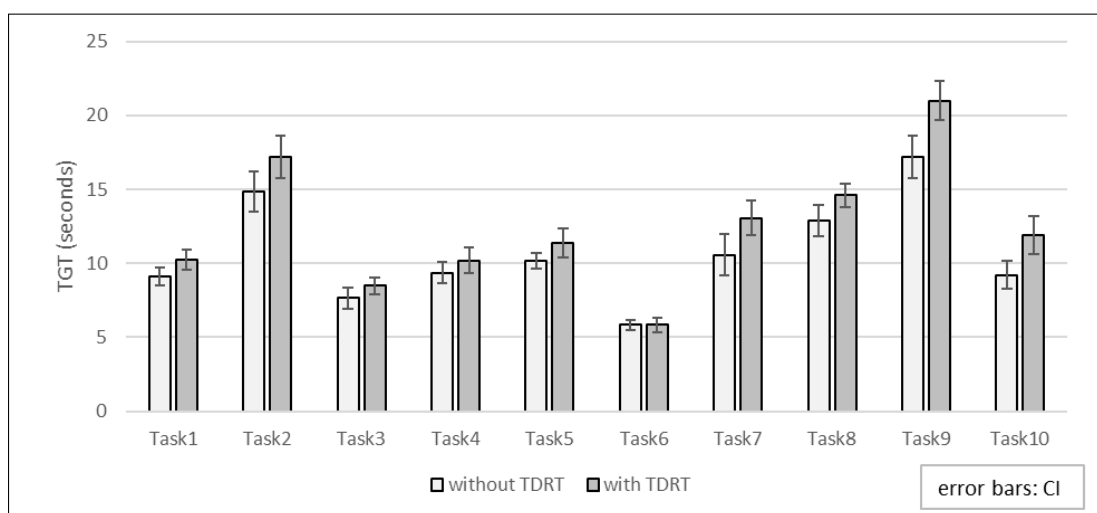


Figure 4.20.: Mean Total Glance Time – With/without TDRT method

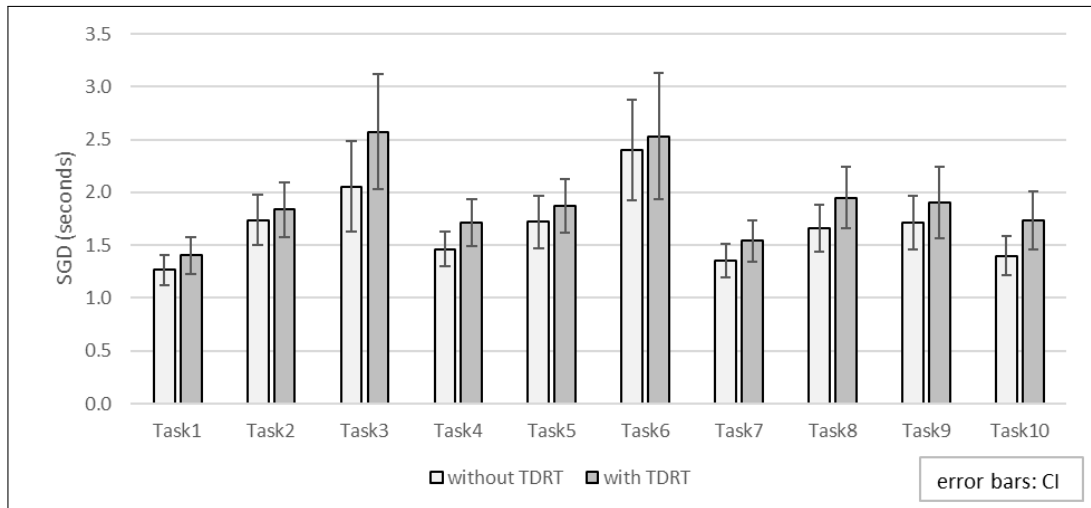


Figure 4.21.: Mean Single Glance Duration – With/without TDRT method

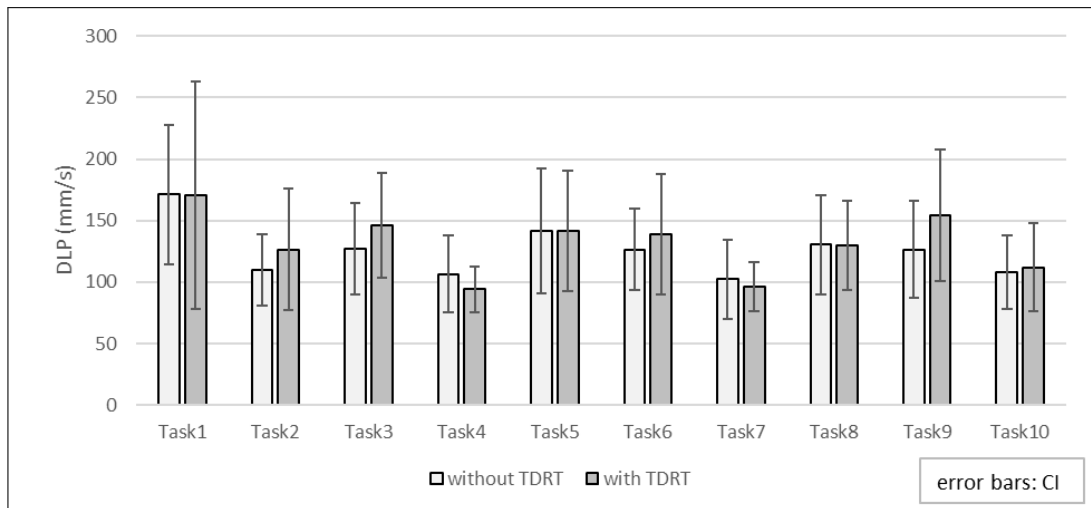


Figure 4.22.: Mean Drift in Lane Position – With/without TDRT method

### Discussion

Based on the statistical results, it could be argued that the glance metrics are significantly different. Figure 4.20 and Figure 4.21 demonstrate that the glance metrics (TGT, SGD) with TDRT are longer. While the reasons are unclear (perhaps manual interference), the outcome would be manageable for testing prototypes: When comparing eye-tracking results during TDRT with criteria and they pass, there is a high likelihood that they would also pass without the TDRT method. The finding of longer glance metrics conforms to the results in Section 3.6.2 (p. 64).

The figures also reveal another issue: Task 1 with the shortest SGDs (Figure 4.21) has the worst driving performance (Figure 4.22). This could be an indication that merely

considering the glance metrics is not enough to judge the likely multidimensional issue of driver distraction. The correlation for the tasks ( $N = 10$ ) between DLP (driving performance) and the SGD (glance metric) in the condition without TDRT is  $r = -.020$ ; (DLP to TGT;  $r = -.130$ ). Therefore, the cognitive aspects and the driving performance are probably missing when assessing tasks with only eye-tracking criteria. The NHTSA guideline completely disregards driving performance. The AAM guideline offers the choice of using eye-tracking **or** driving performance. Based on the results presented, there seems no hindrance to economically perform the three important methods (eye-tracking, cognitive DRT and driving performance) concurrently.

### 4.6.5. Hypotheses 4 – Age Effects

#### Results

The *Task 1, Touchscreen ‘Config’*, was already tested in two former experiments (Krause et al., 2015b). For the analysis, these two experiments with middle-aged people (45–65 years) are called MID1 and MID2, while the new experiment with young subjects (20–26 years) is named YOU. Due to a technical problem in MID1, only 14 out of 21 subjects for TSOT are available. Because of this issue, the data handling of IBM SPSS 22 for a MANOVA also disregards the available TGT and SGD values for seven people (listwise deletion). Therefore, the effective (unequal) group sizes are: MID1 ( $N = 14$ ), MID ( $N = 21$ ) and YOU ( $N = 24$ ).

A MANOVA with the fixed factor experimental groups (independent variable) reports a **significant** influence. Wilks’  $\lambda = .545$ ,  $F(3, 108) = 6.374$ ,  $p < .001$ ,  $\eta_p^2 = .261$  the power to detect the effect was .999.

Two of the three Levene tests for equal group variance are significant: for TSOT and TGT (each  $p = .025$ ). Therefore, the automatically corrected SPSS model is reported:

Total Shutter Open Time:

$F(2, 56) = 19.437$ ,  $p < .001$ ,  $\eta_p^2 = .410$  the power to detect the effect was  $>.999$ .

Total Glance Time:

$F(2, 56) = 5.042$ ,  $p = .010$ ,  $\eta_p^2 = .153$  the power to detect the effect was .796.

Mean Single Glance Duration:

$F(2, 56) = 2.318$ ,  $p = .108$ ,  $\eta_p^2 = .076$  the power to detect the effect was .451

Pairwise tests for TSOT show an insignificant difference ( $p = .480$ ) between MID1 ( $M = 12.28$  s;  $SD = 1.98$  s) and MID2 ( $M = 13.01$  s;  $SD = 1.78$  s). However, MID1 and MID2 have **significantly longer TSOTs** compared to YOU ( $M = 10.15$  s;  $SD = 0.98$  s): MID1  $p < .001$ ; MID2  $p = .001$ .

The TGT of MID1 ( $M = 9.86$  s;  $SD = 2.64$  s) and MID2 ( $M = 11.98$  s;  $SD = 2.84$  s) is **not significantly different** ( $p = .212$ ). While YOU ( $M = 8.92$  s;  $SD = 1.45$  s) has **no significant difference** ( $p > .999$ ) to MID1, YOU is **significantly different** from MID2 ( $p = .008$ ).

The SGD overall test reported no significance, nevertheless the results are reported to complete the picture: The SGD of MID1 ( $M = 1.02$  s;  $SD = 0.23$  s) and MID2 ( $M = 1.07$  s;  $SD = 0.23$  s) is **not significantly different** ( $p > .999$ ). Group YOU ( $M = 1.14$  s;  $SD = 0.29$  s) also shows **no significant difference** to MID1 ( $p = .159$ ) and MID2 ( $p = .350$ ).

## Discussion

It must be kept in mind that this topic uses only one task from different experiments and setups. A dedicated between-subject experiment with diverse tasks would be valuable when looking for age effects. For this discussion, also, the results from the subtask comparisons on pp. 61 are used:

Similar to the observations on the subtask level (Table 3.3, p. 62), the middle-aged people need (significantly) longer TSOTs for *Task 1, Touchscreen ‘Config’*. Therefore, this effect seems robust.

For the TGT in Table 3.1 (p. 62), the two groups (touchscreen and rotary knob) display no clear trend. Also the two groups MID1 and MID2 demonstrate no clear trend when comparing the TGT to the group YOU for *Task 1, Touchscreen ‘Config’*: one comparison is clearly significant, one is far from significant. When considering the standard deviation of YOU (1.45 s), this is approximately doubled for MID1 (2.64 s) and MID2 (2.84 s). This indicates that the variability of TGT spreads for older groups. Depending on the sampling and task, the average performance might sometimes be comparable to younger groups.

In Table 3.2 (p. 62), the younger group exhibits approximately 10%–30% longer SGDs. However, the SGD is not significant longer for the younger group in *Task 1, Touchscreen ‘Config’*, while the trend to longer SGDs is still present (about 7%–12%). A possible explanation could be that the two System Response Times within *Task 1* (each about 2 s) diminishes the difference in SGD between MID and YOU.

### 4.6.6. Issue 5 – Training/Accommodation Effects

#### Results

The radio-tuning task, *Task 2, Touchscreen – ‘Radio Tuning’* was performed in the first part of the experiment and close to the end (see experimental procedure in Section 4.3, p. 86). This is used in a repeated-measures MANOVA.

The dependent variables (DV) are the metrics:

- TGT to the IVIS; average of two trials
- Mean SGD to the IVIS calculated by averaging the SGD of two trials (SGD = TGT/NOG); for NOG, the fractional approach is used

The independent variable (IV) is:

- The point in time (Figure 4.23) when the task is performed in the experimental procedure (early, late). Between these points in time, the experimental blocks *Occlusion* and *Baseline (Unoccluded)* are carried out and give additional training on the task.

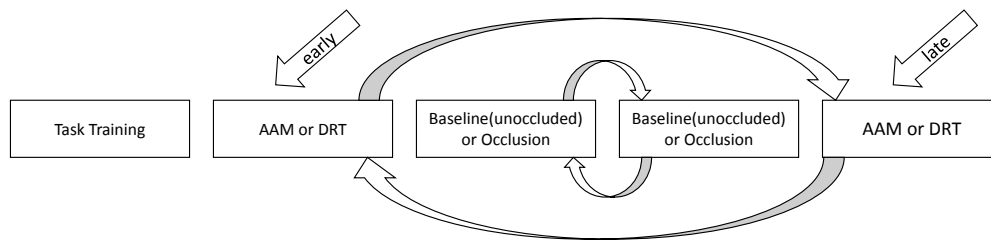


Figure 4.23.: Radio Tuning, Point in Time (early, late)

The analysis reports an overall significant outcome: Wilks'  $\lambda = .375$ ,  $F(2, 22) = 18.323$ ,  $p < .001$ ,  $\eta_p^2 = .625$  the power to detect the effect was  $> .999$ . The univariate and therefore pairwise tests:

Total Glance Time:

$F(1, 23) = 38.209$ ,  $p < .001$ ,  $\eta_p^2 = .624$  the power to detect the effect was  $> .999$ .

Single Glance Duration:

$F(1, 23) = 3.373$ ,  $p = .079$ ,  $\eta_p^2 = .128$  the power to detect the effect was  $.421$ .

The TGT is significantly reduced from the early point in time ( $M = 17.83$  s;  $SD = 3.29$  s) to the later retest ( $M = 13.54$  s;  $SD = 2.86$  s). The SGD is not significant, however it is shorter in tendency (early:  $M = 1.76$  s;  $SD = 0.69$  s; late:  $M = 1.59$  s;  $SD = 0.45$  s). The Pearson correlation between the early and late SGDs is  $r = .764$  ( $N = 24$ ). While the mean SGD is not significantly shorter between the early and late test, the AAM 85<sup>th</sup> percentile SGD exhibits a remarkable drop from 2.32 s (early) to 1.86 s (late). The NOG was not mentioned in the hypothesis, but drops also from 11.4 (early) to 9.03 (late).



## Discussion

The TGT results indicate that the short training at the beginning of the experiment was not fully sufficient. There was still considerable learning in progress (shortening of the TGT by 24%). The NHTSA guideline has a block-wise procedure; each task is trained and measured sequentially. This procedure was not chosen for this experiment due to the four measurements methods (baseline, occlusion, TDRT, eye-tracking). For example, the head-mounted eye-tracking and occlusion are mutually exclusive and would (excessively) increase the required calibration for the eye-tracking.

This ongoing learning is undesirable for the evaluation experiment. Nevertheless, within the experiment, the radio tuning is one of the longer and more complex tasks; the 24% shortening should be a worst case. In addition, the ongoing learning is spread across the measurement conditions by the randomness in the experimental procedure; again the defined early/late point in time for the radio tuning is a worst-case condition. For experiments these kind of quality data are typically not available or reported. The eye-tracking procedure of the NHTSA guideline would even rely on a single-trial measurement. In this case, not even quality data between trials can be calculated. In the pass/fail-Table 4.1 (p. 95), the NHTSA TEORT columns reveal that three tasks that failed in the first trial would pass this criterion in the second trial.

Comparisons between the results of this experiment and the results of the radio tuning app in Krause et al. (2015a) are restricted, particularly for length of time dependent values. In Krause et al. (2015a), the radio tuning was consecutively performed three times in the already-started radio application, while in this experiment, the tuning task included starting the application and performing one tuning.

In Krause et al. (2015a), the discussion focused on the SGDs and they were wondering that these were very different between two reported experiments for the same application. The AAM 85<sup>th</sup> percentile was around 2 s for one experiment and 1.55 s for another reported experiment. In this thesis, 2.32 s (early) to 1.86 s (late) were measured (AAM 85<sup>th</sup>). Krause et al. (2015a) used the radio-tuning task for approximately one hour frequently. The counterintuitive hypothesis in issue five was that extensive training in Krause et al. (2015a) may prolong SGDs for the radio-tuning task because test subjects feel safe to look longer when the task is highly trained. This seems unreasonable based on the statistical results presented above. The mean SGD between an early and late point during the experiment displayed no statistical difference. The tendency was in the wrong direction and the 85<sup>th</sup> percentiles even demonstrated a considerable drop.

Another hypothesis, stated in Krause et al. (2015a), was that the long glance strategies are perhaps motivated by a carry-over effect. Krause et al. (2015a) included a number input on a touchscreen keyboard for task training that resulted in SGDs of approximately 2 s (AAM 85<sup>th</sup>). Another mentioned experiment with surprisingly short 1.55 s AAM P85 SGD did not incorporate any touchscreen keyboards. The new evaluation experiment of this thesis included extensive inputs on touchscreen keyboards and again displayed longer SGDs for radio tuning (early: 2.32 s; late: 1.86 s). This conforms to the carry-over hypothesis. If the hypothesis is true, it would be challenging to reliably measure SGDs in experiments. The typical industrial testing includes different tasks. This is also explicitly allowed, e.g., in the NHTSA guideline. The carry-over hypothesis implies that the SGD

result would depend on the type and mixture of the tasks within one experiment, which is highly undesirable for testing.

A further influence could be: Artifacts of subject sampling. The Pearson correlation above ( $r = .764$ ) indicates that glance strategies are individual. The AAM 85<sup>th</sup> percentile can be influenced by a few people with long glances. To separate the influence of carry-over and subject sampling could be a topic for further research.

The radio tuning was also used in Krause et al. (2015c) with occlusion. The average R-metric (TSOT/TTT) was .647. In Krause et al. (2015a), the R was .636 on a tablet and .659 on a smartphone. In the present evaluation experiment, the R was .672. Therefore, the spread of these R results in different experiments, in different settings, with different examiners and different subjects on different devices is  $.672 - .636 = .036$ . Referenced to the middle of this range, the four results lie within  $\pm 3\%$ . This demonstrates the impressive power of relative metrics. The TSOT and TTT are measured under the same conditions. The relative calculation (TSOT/TTT) cancels out most of the experimental disturbances and the result can be used to purely characterize a task.

---

## 5. Conclusion

The conclusion summarizes and merges results and discussions of the experiments. Possible implications and recommendations are then derived for driver distraction testing (i.e., guidelines and standards).

### Summary

Some subtasks of the experiment to construct the prediction model were compared to a former experiment. The descriptive statistics (Section 3.6.1) indicated that a middle-aged group has longer TSOT during occlusion. The TGT were longer, comparable or shorter. The mean and P85 SGDs were longer for the younger group. Inference statistics in the evaluation experiment for one task (Section 4.6.5) found significantly longer TSOT, indistinct outcomes (significant and not significant) for TGT and no statistical differences for mean SGD. However, a descriptive trend for longer SGD in the younger group was found.

The descriptive statistics of the experiment for building the model (Section 3.6.2) indicated that TTT, TGT and SGD increase when tasks are combined with the TDRT measurement method (triple-task setting). An inference statistical analysis in the evaluation experiment reported significantly longer TGT and SGD, while no difference in the DLP driving performance was found when tasks are combined with the TDRT method.

An in-depth analysis of glance metrics during System Response Times was conducted for the experiment to build the prediction model (Section 3.6.3). The results help to understand, estimate and model glance behavior during SRTs. The evaluation experiment demonstrated (Section 4.6.3) that it is possible to lower SGDs by inserting an artificial delay.

Within the evaluation experiment, a test-retest of one task (radio tuning) revealed insights regarding training effects during the experiment (Section 4.6.6). The TGT becomes significantly shorter with training. The NOG also dropped remarkably. The SGD only displayed the tendency to get shorter. The radio-tuning task was used in former experiments with a wide range of different SGD results. It would be reasonable that carry-over effects of glance strategies and/or subject sampling also had an undesirable influence on SGD. Touchscreen keyboards are particularly suspected of encouraging longer glance strategies and transferring this behavior to other tasks.

The predictions of the model were evaluated (Section 4.6.2) and demonstrated reasonable overall results for the different metrics of glance, occlusion, driving and DRT methods (except for one metric: DFH). The (open source) tool and database could be helpful in obtaining a provisional estimate. In no case should the tool be used to replace final subject testing. The model is intended to lower the amount of (unsuitable) tasks that are

---

planned for subject testing and improve new tasks in an early stage of interaction design (e.g., paper prototyping).

## Implications and Recommendations

With the information from Östlund et al. (2005) and Section 2.3, the comparison of SDLPs in driver distraction testing (with a typical task duration of 5–15 s) could be judged as inappropriate. Comparing SDLPs of tasks that have different lengths is questionable due to the fact that SDLP is length dependent and therefore should be correlated to TGT. The argument of disregarding driving metrics (SDLP) due to the correlation to eye-tracking metrics (TGT) would be circular reasoning. For this thesis, the DLP (and median DLP deterioration) worked quite well to assess lateral driving performance.

The occlusion standard ISO 16673 (2007) is, at 15 pages, one of the shortest, most precise and understandable standards of the ISO working group. This probably helped to make the occlusion technique popular. A drawback is that references (e.g., guidelines) specify unique subject sampling or procedures. This renders the fundamental idea of standardization useless, disables comparisons of results and requires several (regional) testings.

The informational appendix of the occlusion standard includes some recommendations and conjectures regarding the glance behavior during System Response Times. The lack of experimental data for the delay topic is reduced by the outcomes of this thesis.

A general benefit of the occlusion technique is that it not concealed by long delays like the eye-tracking SGD metrics. The influence (waiting) is obvious.

The relative R-ratio (TSOT/TTT) seems a powerful tool that cancels out many experimental problems. However, it is not used by guidelines. Overall, relative testing seems uncommon and should be fostered instead of absolute criteria testing. Examples for relative testing are the radio tuning reference (AAM) and the baseline driving within the LCT (ISO 26022, 2010).

In this thesis, one subject again had severe problems operating the DRT (Section 4.5)—a problem that was also observed during a former experiment. In both cases this was revealed later in the data analysis. Therefore, a comment in ISO/DIS 17488 (2014) for the data analyst or even the examiner could be helpful to check for the behavior that people react to after 1 s. It is reasonable that automatically switching off the stimulus after 1 s generally feels a stimulus on its own; perhaps fading out the stimulus can be an advance.

While the occlusion needed 15 pages, the DRT standard is expanded to approximately 80 pages. Whether an engineer without any contact to the related ISO working group would be able to build or perform a DRT properly could be questionable.

DIN EN ISO 15007-1 (2003) may be advanced if the topic of split glances is mentioned. Every recoding of eye-tracking must be started and stopped, which splits glances that are in progress. This happens every time and for every task. The influence depends on setup and experiment. It is a significant issue when assessing short subtasks. The topic was explained and discussed on p. 38. For this thesis, a so called *fractional* approach was chosen. An alternative when handling longer tasks could be to disregard unreasonable

---

short glances, especially when starting or stopping a measurement interval. The split topic should at least be mentioned.

There can be distinctive differences between task-related glances to an IVIS and eyes-off-road metrics. The differences are principally due to short speedometer checks (the subjects are instructed to maintain distance and speed), which can have a considerable impact. Delays can be exploited to evoke speedometer checks and short IVIS check glances. A guideline which relies on eyes-off-road metrics and specifies a test procedure that explicitly states a speed display can be mounted in the driving scene (NHTSA, 2014, pp.35–36 VI.C.3.c) is likely to cause problems regarding test reliability in different laboratories and car setups. Also, the single-trial approach (NHTSA, 2014, p.41 E.9) provides the impression that reliable testing is not prioritized in this guideline; at least for eye-tracking (occlusion: five trials).

The task-related approach (AOI IVIS) seems more robust against uncontrolled disturbances and is likely independent of the car technology (cluster versus head-up-display). It must be also mentioned that eyes-off-road metrics can sometimes have a benefit: In a study for a traffic light assistant (KOLIBRI), a visualization interface with the shortest task-related glances had no clear advantage when eyes-off-road metrics were used for assessment (cf. Krause and Bengler, 2012b,a). The interface motivated the test subjects to combine task-related glances and speedometer checks.

All KOLIBRI reports (e.g., Krause and Bengler, 2014) used histograms and metrics based on the distribution of all glances together. This is similar to the way the AAM derived its glance criterion from literature (Driver Focus-Telematics Working Group, 2006, p. 41, p. 57). It appears a reliable method. It is curious why guidelines changed to procedures that can be heavily affected by accidental (measurement) artifacts, the random individual glance behavior of each single subject, and even the use of only one measurement trial.

The histogram approach also solves another problem: The assessment of continuous tasks, e.g., using a satnav application for route guidance. The NHTSA guideline specifies procedures for *testable tasks* and defines *testable tasks* in a way that seems not to include continuous ongoing tasks. With this logic, satnav usage would be an untestable task. When considering TGT and TEORT metrics for long-travel satnav usage, the cumulative eyes-off-road time would be impressively high but likely irrelevant. Nevertheless, it is recommendable to assess these interfaces too.

When using the online tool, sooner or later questions will emerge concerning whether the model can be extended. The values measured from different methods (e.g., eye-tracking metrics, driving metrics) are paired because they are from single test subjects. The (open source) application and setup to measure the subtasks is documented in this thesis. Therefore, it would be possible to measure additional subjects (e.g., from different age groups) with the same application and add the results to the database. A challenging request would be to add other subtasks while preserving the old ones. One approach could be to test these new subtasks with new subjects and include some of the old subtasks for reference. The reference subtasks may be used to find and map the new test subjects to similar existing test subjects in the data, based on their performance. With this mapping, perhaps the new subtasks can be merged into the database. In other words, the subtasks of two similar test subjects (preexisting and new) are combined to give a new (virtual) test subject in the database.

---

Task testing is done on a regular basis in some laboratories. If the tasks within these tests could be programmed to automatically mark subtasks (similar to the application used in this thesis), subtask data could be easily gathered. If the approach to add subtasks from different subjects based on reference performances is be evaluated, the database could be filled automatically.

An interesting case would be frameworks that restrict the usable GUI widgets and standardize the interface (e.g., Android Auto). If these widgets are tested and saved to a database, the subjective selection of suitable subtasks by a human factors engineer in the predictive modeling would be eliminated; the selection could be done objectively or perhaps even automated.

---

# Bibliography

- 2000/53/EC (1999). Commission Recommendation of 21 December 1999 on safe and efficient in-vehicle information and communication systems: A European statement of principles on human-machine interface (notified under document number C(1999) 4786).
- 2008/653/EC (2008). Commission Recommendation of 26 May 2008 on safe and efficient in-vehicle information and communication systems: update of the European Statement of Principles on human-machine interface (notified under document number C(2008) 1742). <http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32008H0653> (accessed 04/16/2016). vii, 7, 9, 10
- Anderson, G., Doherty, R., and Ganapathy, S. (2011). User Perception of Touch Screen Latency. In Marcus, A., editor, *Design, User Experience, and Usability. Theory, Methods, Tools and Practice: First International Conference, DUXU 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part I*, pages 195–202. Springer Berlin Heidelberg, Berlin, Heidelberg. 16, 19
- Anderson, J. R. and Lebiere, C. (1998). *The Atomic Components of Thought*. Erlbaum, Mahwah, NJ. 35
- Avenoso, A. (2012). European Progress. Presentation at the Driven to Distraction Conference, Toronto <http://www.distracteddriving.ca/presentations/Panel-Legislation&Enforcement-SpeakerAvenoso.pdf> (accessed 08/06/2016). 3
- Baumann, M., Keinath, A., Krems, J. F., and Bengler, K. (2004). Evaluation of in-vehicle HMI using occlusion techniques: experimental results and practical implications . *Applied Ergonomics*, 35(3):197 – 205. The Occlusion Technique. 29
- Bengler, K. and Broy, V. (2008). Animationen im Fahrzeug GUI – Randbedingungen für deren ergonomische Gestaltung. In *Produkt- und Produktions-Ergonomie – Aufgabe für Entwickler und Planer*, 54. Kongress der Gesellschaft für Arbeitswissenschaft, pages 157–161, Dortmund. GfA Press. 17
- Card, S. K. (1981). The Model Human Processor: A Model for Making Engineering Calculations of Human Performance. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 25(1):301–305. 28, 32
- Card, S. K., Moran, T. P., and Newell, A. (1980a). Computer text-editing: An information-processing analysis of a routine cognitive skill . *Cognitive Psychology*, 12(1):32 – 74. 28
- Card, S. K., Moran, T. P., and Newell, A. (1980b). The Keystroke-level Model for User Performance Time with Interactive Systems. *Commun. ACM*, 23(7):396–410. 28

- Card, S. K., Moran, T. P., and Newell, A. (1986). *The Model Human Processor – An Engineering Model of Human Performance*, chapter Chapter 45. Wiley, New York. 28
- Card, S. K., Newell, A., and Moran, T. P. (1983). *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA. 28
- Carsten, O. and Nilsson, L. (2001). Safety Assessment of Driver Assistance Systems. *European Journal of Transport and Infrastructure Research*, 1(3):225–243. 8
- Carter, S. (2010). Mr. Data Converter. Open source tool to convert CSV to JSON <https://github.com/shancarter/mr-data-converter> (accessed 06/20/2016). 55
- Conti, A. S., Kremser, F., Krause, M., An, D., and Bengler, K. (2015). The Effect of Varying Target Sizes and Spaces between Target and Non-target Elements on Goal-directed Hand Movement Times while Driving . *Procedia Manufacturing*, 3:3168 – 3175. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. 23, 31
- DIN EN ISO 15007-1 (2003). Road vehicles Measurement of driver visual behaviour with respect to transport information and control systems Part 1: Definitions and parameters (ISO 15007-1:2002); German version EN ISO 15007-1:2002. 56, 112
- DIN EN ISO 17287 (2003). Road vehicles – Ergonomic aspects of transport information and control systems – Procedure for assessing suitability for use while driving. 20
- DIN EN ISO 9241-1 (1997). Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten Teil 1: Allgemeine Einführung. 13
- Dingus, T. A., Guo, F., Lee, S., Antin, J. F., Perez, M., Buchanan-King, M., and Hankey, J. (2016). Driver crash risk factors and prevalence evaluation using naturalistic driving data. *Proceedings of the National Academy of Sciences*, 113(10):2636–2641. 1, 3
- Driver Focus-Telematics Working Group (2006). Alliance of Automobile Manufacturers. Statement of Principles, Criteria and Verification Procedures on Driver Interactions with Advanced In-Vehicle Information and Communication Systems. <http://www.autoalliance.org/index.cfm?objectid=D6819130-B985-11E1-9E4C000C296BA163> (accessed 04/16/2016). 7, 8, 12, 13, 18, 20, 22, 38, 40, 77, 113
- Eagleman, D. M. (2009). Brain Time. <https://www.edge.org/conversation/brain-time> (accessed 04/24/2016). 16
- Elwart, T., Green, P., and Lin, B. (2015). Predicting Driver Distraction Using Computed Occlusion Task Times: Estimation of Task Element Times and Distributions. Technical Report ATLAS-2015-01 <http://www.atlas-center.org/wp-content/uploads/2013/12/Green-ATLAS-2015-01.pdf> (accessed 11/14/2016). 29, 30
- ESoP draft (2005). European Statement of Principles on the Design of Human Machine Interaction (ESoP 2005) Draft. <http://www.imobilitysupport.eu/library/imobility-forum/working-groups/active/human-machine-interaction/other-reports-5/2416-hmi-wg-esop-hmi-01-jun-2005-1/file>, (accessed 06/26/2016). 11, 12



- 
- Feuerstack, S., Lüdtke, A., and Osterloh, J.-P. (2015). A Tool for Easing the Cognitive Analysis of Design Prototypes of Aircraft Cockpit Instruments: The Human Efficiency Evaluator. In *Proceedings of the European Conference on Cognitive Ergonomics 2015, ECCE '15*, pages 22:1–22:8, New York, NY, USA. ACM. 36
- Google (2016a). Auto App Quality. <https://developer.android.com/distribute/essentials/quality/auto.html> (accessed 05/30/2016). 18
- Google (2016b). Keeping Your App Responsive. <http://developer.android.com/training/articles/perf-anr.html> (accessed 04/26/2016). 17
- Google (2016c). NetworkOnMainThreadException. <http://developer.android.com/reference/android/os/NetworkOnMainThreadException.html> (accessed 04/26/2016). 17
- Gore, B. F. (2011). Man-machine Integration Design and Analysis System (MIDAS) v5: Augmentations, Motivations, and Directions for Aeronautics Applications. In Cacciabue, C. P., Hjalmdahl, M., Luedtke, A., and Riccioli, C., editors, *Human Modelling in Assisted Transportation: Models, Tools and Risk Methods*, pages 43–54. Springer Milan, Milano. 37
- Gore, B. F., Hooley, B. L., Wickens, C. D., and Scott-Nash, S. (2009). A Computational Implementation of a Human Attention Guiding Mechanism in MIDAS v5. In Duffy, V. G., editor, *Digital Human Modeling: Second International Conference, ICDHM 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009. Proceedings*, pages 237–246. Springer Berlin Heidelberg, Berlin, Heidelberg. 37
- Hankey, J. M., Dingus, T. A., Hanowski, R. J., Wierwille, W. W., and Andrews, C. (2001a). In-Vehicle Information Systems Behavioral Model and Design Support: Final Report. FHWA-RD-00-135. 33, 35
- Hankey, J. M., Dingus, T. A., Hanowski, R. J., Wierwille, W. W., and Andrews, C. (2001b). In-Vehicle Information Systems Behavioral Model and Design Support IVIS Demand Prototype Software User's Manual. FHWA-RD-00-136. 33
- Harvey, C. (2011). *Modelling and evaluating drivers' interactions with in-vehicle information systems (IVIS)*. PhD thesis, University of Southampton. 32
- Harvey, C. and Stanton, N. A. (2013). Modelling the hare and the tortoise: predicting the range of in-vehicle task times using critical path analysis. *Ergonomics*, 56(1):16–33. PMID: 23140467. 28, 31
- Heinrich, C. (2013). Fighting Driver Distraction - Worldwide Approaches. ESV conference 2013, Seoul Korea, Paper Number 13-0290 <http://www-esv.nhtsa.dot.gov/Proceedings/23/files/23ESV-000290.PDF> (accessed 05/28/2016). 8
- Horrey, W. J., Wickens, C. D., and Consalus, K. P. (2006). Modeling drivers' visual attention allocation while interacting with in-vehicle technologies. *Journal of Experimental Psychology: Applied*, 12(2):67–78. 37

- ISO 16673 (2007). Road vehicles – Ergonomic aspects of transport information and control systems – Occlusion method to assess visual demand due to the use of in-vehicle systems. 18, 68, 74, 92, 102, 112, 124
- ISO 26022 (2010). Road vehicles – Ergonomic aspects of transport information and control systems – Simulated lane change test to assess in-vehicle secondary task demand. 26, 42, 112
- ISO/DIS 17488 (2014). Road vehicles – Transport information and control systems – Detection-response task (DRT) for assessing attentional effects of cognitive load in driving. 43, 93, 112
- ITU G.114 (2003). *Recommendation G.114 (05/03): One-way transmission time*. <http://de.onlinecomponents.com/datasheet/rkjxt1e12001.aspx?p=10114295> (accessed 04/17/2016). 15
- JAMA (2004). Japan Automobile Manufacturers Association – Guideline for In-vehicle Display Systems – Version 3.0. 7
- Johansson, E., Engström, J., Cherri, C., Nodari, E., Toffetti, A., Schindhelm, R., and Gelau, C. (2004). Review of existing techniques and metrics for IVIS and ADAS assessment. Project AIDE. Deliverable D2.2.1. 26, 27
- John, B. E. and Gray, W. D. (1995). CPM-GOMS: An Analysis Method for Tasks with Parallel Activities. In *Conference Companion on Human Factors in Computing Systems, CHI '95*, pages 393–394, New York, NY, USA. ACM. 31
- John, B. E., Prevas, K., Salvucci, D. D., and Koedinger, K. (2004a). Predictive Human Performance Modeling Made Easy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04*, pages 455–462, New York, NY, USA. ACM. 35
- John, B. E., Salvucci, D. D., Centgraf, P., and Prevas, K. C. (2004b). Integrating Models and Tools in the Context of Driving and In-vehicle Devices. In *Proceedings of the Sixth International Conference on Cognitive Modeling*, pages 130–135, Mahwah, NJ. Lawrence Earlbaum. 36
- Jorritsma, W., Haga, P.-J., Cnossen, F., Dierckx, R., Oudkerk, M., and van Ooijen, P. (2015). Predicting human performance differences on multiple interface alternatives: KLM, GOMS and CogTool are unreliable. In *6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences*. Elsevier. 30
- Jürgensohn, T. (2007). Control Theory Models of the Driver. In Cacciabue, P. C., editor, *Modelling Driver Behaviour in Automotive Environments: Critical Issues in Driver Interactions with Intelligent Transport Systems*, pages 277–292. Springer London, London. 26
- Kaaresoja, T. and Brewster, S. (2010). Feedback is... Late: Measuring Multimodal Delays in Mobile Device Touchscreen Interaction. In *International Conference on Multimodal*

- 
- Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ICMI-MLMI '10, pages 2:1–2:8, New York, NY, USA. ACM. <http://doi.acm.org/10.1145/1891903.1891907> (accessed 04/16/2016). vii, 10, 15, 19
- Kang, T.-P., Lin, B. T.-W., Green, P., Pettinato, S., and Best, A. (2013). Usability of a Prototype Generation 4 Hyundai-Kia Navigation-Radio: Evidence from an Occlusion Experiment, and SAE J2365 and Pettitt's Method Calculations. Technical Report UMTRI-2013-11. vii, 30, 31, 100
- Knappe, G. (2009). *Empirische Untersuchungen zur Querregelung in Fahrsimulatoren – Vergleichbarkeit von Untersuchungsergebnissen und Sensitivität von Messgrößen*. PhD thesis, Philosophische Fakultät und Fachbereich Theologie, Friedrich-Alexander-Universität Erlangen-Nürnberg. 20, 21
- Kohlisch, O. and Kuhmann, W. (1997). System response time and readiness for task execution the optimum duration of inter-task delays. *Ergonomics*, 40(3):265–280. 14, 19
- Krause, M. (2015a). Android App to control Arduino DRT (USB remote control). Open source tool <http://www.lfe.mw.tum.de/open-source/drt-rc/> or via [redirect link](#) (accessed 09/01/2016). 73
- Krause, M. (2015b). Arduino Occlusion. Open source tool <http://www.lfe.mw.tum.de/arduino-occlusion/> or via [redirect link](#) (accessed 09/01/2016). 43, 73
- Krause, M. (2015c). Bluetooth Rotary Knob for mock-up interfaces on Android. Open source tool <http://www.lfe.mw.tum.de/en/open-source/bluetooth-rotary-knob/> or via [redirect link](#) (accessed 09/01/2016). 43
- Krause, M. (2016a). Android Numpad Application for Evaluation Experiment. Open source tool <https://github.com/MichaelKrause/numpad> or via [redirect link](#) (accessed 09/01/2016). 78
- Krause, M. (2016b). Android Remote Control Application for Evaluation Experiment. Open source tool <https://github.com/MichaelKrause/rc> or via [redirect link](#) (accessed 09/01/2016). 73
- Krause, M. (2016c). Marker helper tool. Open source tool <http://www.lfe.mw.tum.de/en/open-source/marker/> or via [redirect link](#) (accessed 09/19/2016). 91
- Krause, M., Angerer, C., and Bengler, K. (2015a). Evaluation of a Radio Tuning Task on Android while Driving . *Procedia Manufacturing*, 3:2642 – 2649. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. 21, 27, 42, 77, 90, 109, 110
- Krause, M. and Bengler, K. (2012a). Traffic Light Assistant–Evaluation of Information Presentation. *Advances in Human Aspects of Road and Rail Transportation*, page 166. 113
- Krause, M. and Bengler, K. (2012b). Traffic Light Assistant-Driven in a Simulator. In *Proceedings of the 2012 International IEEE Intelligent Vehicles Symposium Workshops*. 113
-

- Krause, M. and Bengler, K. (2014). KOLIBRI - Ampelassistentz für die Landstraße auf einem Smartphone. *Zeitschrift für Verkehrssicherheit ZVS*, 60(30):135–1410. 113
- Krause, M. and Bengler, K. (2015). Suitability for Use while Driving – Introduction for (App) Developers. Technical Report [http://urban-online.org/cms/upload/download/Introduction\\_DrivingApps\\_final\\_.pdf](http://urban-online.org/cms/upload/download/Introduction_DrivingApps_final_.pdf) (accessed 05/25/2016). 7
- Krause, M. and Conti, A. (2015). Arduino Detection Response Task DRT. Open source tool <http://www.lfe.mw.tum.de/arduino-drt/> or via [redirect link](#) (accessed 09/01/2016). 43
- Krause, M., Conti, A., Henning, M., Seubert, C., Heinrich, C., Bengler, K., Herrigel, C., and Glaser, D. (2015b). App Analytics: Predicting the Distraction Potential of In-vehicle Device Applications. *Procedia Manufacturing*, 3:2658 – 2665. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. vii, 38, 39, 40, 61, 63, 73, 75, 89, 106
- Krause, M., Donant, N., and Bengler, K. (2015c). Comparing Occlusion Method by Display Blanking to Occlusion Goggles. *Procedia Manufacturing*, 3:2650 – 2657. 92, 110
- Krause, M. and Prasch, L. (2016). Android Rotary Knob and Touchscreen Widget Application. Open source tool <https://github.com/MichaelKrause/widgets> or via [redirect link](#) (accessed 09/01/2016). 46
- Krause, M., Rissel, A., and Bengler, K. (2014a). Traffic Light Assistant-What the Users Want. *Proceedings of the Seventh International Conference on Advances in Computer-Human Interactions ACHI, Barcelona, Spain*, pages 235–241. 58, 93
- Krause, M., Yilmaz, L., and Bengler, K. (2014b). Comparison of Real and Simulated Driving for a Static Driving Simulator. *Advances in Human Aspects of Transportation: Part II*, 8:29. 3
- Kurokawa, K. (1990). *Development of an evaluation program for automotive instrument panel design*. PhD thesis, Virginia Polytechnic Institute and State University. vii, 32, 33, 34
- Lee, J., Lee, J. D., and Salvucci, D. D. (2012). Evaluating the Distraction Potential of Connected Vehicles. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '12, pages 33–40, New York, NY, USA. ACM. 36
- Lee, J. Y., Gibson, M., and Lee, J. D. (2015). Secondary Task Boundaries Influence Drivers' Glance Durations. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '15, pages 273–280, New York, NY, USA. ACM. 16
- Lee, J. Y., Gibson, M. C., and Lee, J. D. (2016). Error Recovery in Multitasking While Driving. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5104–5113, New York, NY, USA. ACM. 16

- Leuchter, S. (2009). *Software Engineering Methoden für die Bedienermodellierung in dynamischen Mensch-Maschine-Systemen*. PhD thesis, Technische Universität Berlin. 29, 35
- Maier, F. (2013). *Wirkpotentiale moderner Fahrerassistenzsysteme und Aspekte ihrer Relevanz für die Fahrausbildung*. PhD thesis, Technische Universität München. 2
- Mavor, A. S., Pew, R. W., and (U.S.), N. R. C. (1998). *Modeling human and organizational behavior : application to military simulations / Richard W. Pew and Anne S. Mavor, editors* . National Academy Press Washington, D.C. 35
- Maynard, H. B., Stegemerten, G. J., and Schwab, J. L. (1948). *Methods-time measurement*. McGraw-Hill Book Co., New York. 29
- Michon, J. A. (1985). A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? In Evans, L. and Schwing, R. C., editors, *Human Behavior and Traffic Safety*, pages 485–524. Springer US, Boston, MA. 26
- MIL-STD-1472F (1999). Department of Defense Design Criteria Standard Human Engineering. <http://everyspec.com/MIL-STD/MIL-STD-1400-1499/download.php?spec=MIL-STD-1472F.027465.pdf> (accessed 04/24/2016). 14
- MIL-STD-1472G (2012). Department of Defense Design Criteria Standard Human Engineering. [http://quicksearch.dla.mil/qsDocDetails.aspx?ident\\_number=36903](http://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=36903) (accessed 04/16/2016). ix, 13, 14
- Miller, R. B. (1968). Response Time in Man-computer Conversational Transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I, AFIPS '68 (Fall, part I)*, pages 267–277, New York, NY, USA. ACM. 9, 13, 14, 15, 16
- NHTSA (2012). Visual-Manual NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices. Document Citation: 77 FR 11199. <https://www.federalregister.gov/articles/2012/02/24/2012-4017/visual-manual-nhtsa-driver-distraction-guidelines-for-in-vehicle-electronic-devices> (accessed 04/16/2016). 12
- NHTSA (2013). Visual-Manual NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices. FR 04232013. [http://www.nhtsa.gov/staticfiles/nti/distracted\\_driving/pdf/distracted\\_guidelines-FR\\_04232013.pdf](http://www.nhtsa.gov/staticfiles/nti/distracted_driving/pdf/distracted_guidelines-FR_04232013.pdf) (accessed 04/16/2016). 12, 13
- NHTSA (2014). Visual-Manual NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices. V1.01. [http://www.distraction.gov/downloads/pdfs/11302c-Visual\\_Manual\\_Distraction\\_Guidelines\\_V1-1\\_010815\\_v1\\_tag.pdf](http://www.distraction.gov/downloads/pdfs/11302c-Visual_Manual_Distraction_Guidelines_V1-1_010815_v1_tag.pdf) (accessed 04/16/2016). 7, 8, 30, 39, 40, 54, 94, 96, 113
- NHTSA (2016). Docket No. NHTSA-2013-0137 Visual-Manual NHTSA Driver Distraction Guidelines for Portable and Aftermarket Devices. <https://www.regulations.gov/docket?D=NHTSA-2013-0137> Prepublication Version: [http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Distraction\\_Phase\\_2\\_FR\\_Notice\\_11-21-16\\_final.pdf](http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Distraction_Phase_2_FR_Notice_11-21-16_final.pdf) (accessed 11/26/2016). 8

- Nielsen, J. (1993). Response Times: The 3 Important Limits. <https://www.nngroup.com/articles/response-times-3-important-limits/> (accessed 06/26/2016). 11, 13
- Östlund, J., Nilsson, L., Carsten, O., Merat, N., Jamson, H., Jamson, S., Mouta, S., Carvalhais, J., Santos, J., Anttila, V., Sandberg, H., Luoma, J., De Waard, D., Brookhuis, K., Johansson, E., Engström, J., Victor, T., Harbluk, J., Janssen, W., and Brouwer, R. (2004). Deliverable 2 – HMI and safety-related driver performance. *Human Machine Interface And the Safety of Traffic in Europe (HASTE) Project*. [http://ec.europa.eu/transport/roadsafety\\_library/publications/haste\\_d2\\_v1-3\\_small.pdf](http://ec.europa.eu/transport/roadsafety_library/publications/haste_d2_v1-3_small.pdf) (accessed 06/19/2016). 27
- Östlund, J., Peters, B., Thorsl, B., Engström, J., Markkula, G., Keinath, A., Horst, D., Regienov, S. J., Mattes, S., and Foehl, U. (2005). Driving performance assessment methods and metrics. Project AIDE. Deliverable D2.2.5. vii, 20, 21, 23, 24, 26, 112
- Pettitt, M. (2006). *Visual demand evaluation methods for in-vehicle interfaces*. PhD thesis, University of Nottingham. 28, 29
- Popova-Dlugosch, S., Krause, M., and Bengler, K. (2011). To Touch or Not To Touch – Gestaltungshinweise für die Touchscreens im Kraftfahrzeug. In *57. Kongress der Gesellschaft für Arbeitswissenschaft. Mensch, Technik, Organisation – Vernetzung im Produktentstehungs- und -herstellungsprozess*, pages 269–272. GfA-Press, Dortmund. 46
- Purucker, C., Naujoks, F., Prill, A., Krause, T., and Neukum, A. (2014). Vorhersage von Blickabwendungszeiten mit Keystroke-Level-Modeling. In Butz, A., Koch, M., and Schlichter, J., editors, *Mensch & Computer 2014 - Workshopband*, pages 239–248, Berlin. De Gruyter Oldenbourg. 30, 38
- Purucker, C., Naujoks, F., Prill, A., and Neukum, A. (2017). Evaluating distraction of in-vehicle information systems while driving by predicting total eyes-off-road times with keystroke level modeling . *Applied Ergonomics*, 58:543 – 554. 30, 38, 100
- Rassl, R. (2004). *Ablenkungswirkung tertiärer Aufgaben im Pkw Systemergonomische Analyse und Prognose*. PhD thesis, Institute of Ergonomics, Technische Universität München. 15
- Remlinger, W. (2013). *Analyse von Sichteinschränkungen im Fahrzeug*. PhD thesis, Technische Universität München. 2
- SAE J 2944 (2013). Proposed Draft - Operational Definitions of Driving Performance Measures and Statistics. 20, 26
- SAE J2364 (2004). Navigation and Route Guidance Function Accessibility While Driving. 7, 29
- SAE J2365 (2002). Calculation of the Time to Complete In-Vehicle Navigation and Route Guidance Tasks. 29, 30, 31
- Salvucci, D. D. (2005). Distract-R: Rapid prototyping and evaluation of in-vehicle interfaces. In *In Human Factors in Computing Systems: CHI 2005 Conference Proceedings*, pages 581–589. ACM Press. 26, 36, 100

- 
- Salvucci, D. D. (2006). Modeling Driver Behavior in a Cognitive Architecture. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(2):362–380. 36
- Salvucci, D. D. (2009). Rapid Prototyping and Evaluation of In-vehicle Interfaces. *ACM Trans. Comput.-Hum. Interact.*, 16(2):9:1–9:33. 36
- Schneegaß S., Pfleging, B., Kern, D., and Schmidt, A. (2011). Support for Modeling Interaction with Automotive User Interfaces. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '11, pages 71–78, New York, NY, USA. ACM. 29
- Steelman, K. S., McCarley, J. S., and Wickens, C. D. (2011). Modeling the Control of Attention in Visual Workspaces. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(2):142–153. 37
- Stetson, C., Cui, X., Montague, P. R., and Eagleman, D. M. (2006). Motor-sensory recalibration leads to an illusory reversal of action and sensation. *Neuron*, 51(5):651–659. 16
- Stevens, A., Board, A., Allen, P., and Quimby, A. (1999). A safety checklist for the assessment of in-vehicle information systems: a user’s manual. Report PA3536/99. [http://www.trl.co.uk/umbraco/custom/report\\_files/PA3536-99.pdf](http://www.trl.co.uk/umbraco/custom/report_files/PA3536-99.pdf) (accessed 06/26/2016). 8, 12
- Stevens, A. and Cynk, S. (2011). Checklist for the assessment of in-vehicle information systems. MIS005; Transport Research Laboratory. 8, 12
- Summala, H., Nieminen, T., and Punto, M. (1996). Maintaining Lane Position with Peripheral Vision during In-Vehicle Tasks. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 38(3):442–451. 26
- Tullis, T. (1984). *Predicting the Usability of Alphanumeric Displays*. PhD thesis, Rice University. 36
- Urbas, L., Leuchter, S., Schaft, T., and Heinath, M. (2008). Modellgestützte Bewertung der Ablenkungswirkung von neuen interaktiven Diensten im Fahrzeug. In Alkassar, A. and Siekmann, J., editors, *Sicherheit 2008*, pages 329–340, Bonn, Germany. Gesellschaft für Informatik. 29
- Utesch, F. and Vollrath, M. (2010). Do slow computersystems impair driving safety? In *European Conference on Human Centred Design for Intelligent Transport Systems, 2nd, 2010, Berlin, Germany*. 15
- Wickens, C. D., Helleberg, J., Goh, J., Xu, X., and Horrey, W. J. (2001). Pilot Task Management: Testing an Attentional Expected Value Model of Visual Scanning. Technical Report ARL-01-14/NASA-01-7. 36

---

# A. Appendix – Prediction Tool Manual

The following section describes the GUI of the prediction model, available online:  
<http://www.distract.one>.

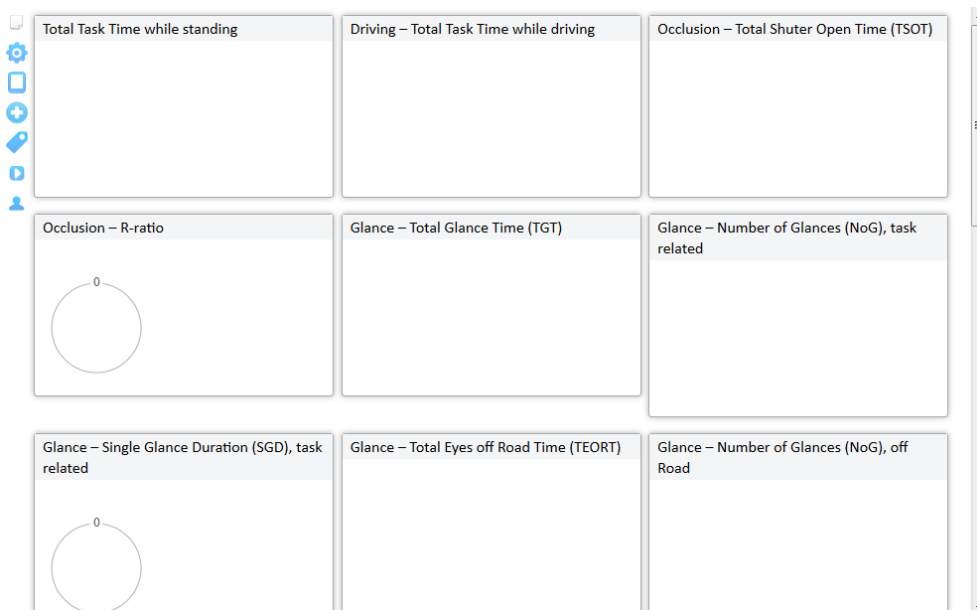


Figure A.1.: Online tool (no subtasks selected)

On the left a toolbar (fixed position, non-scrolling) with seven icons is always accessible (Figure A.1). The tool-tips show appropriate hints.



The ‘New Window’ function opens a new browser tab (or new browser window) with the online tool and no subtask selected. This short-cut is known from typical desktop applications (e.g., ‘New File’ or ‘New Document’ menu item)



The config option opens or closes the config window (see Figure A.2). Within the config window, the five main topics can be enabled or disabled. The Total Task Time while standing is always calculated, the other 12 metrics can be enabled or disabled group-wise with check-boxes. Tool-tips show the grouping. Disabling means the panel for a metric is not shown in the main window (Figure A.1) and the column of the metric is disabled in the subtask selection window (Figure A.4). The occlusion option has an additional box to configure whether delays should be ignored. If enabled, delay subtasks are ignored when calculating the TSOT and R-metric. This option could be helpful when someone is following the ‘System Response Delay’ recommendation in the appendix of ISO 16673 (2007).



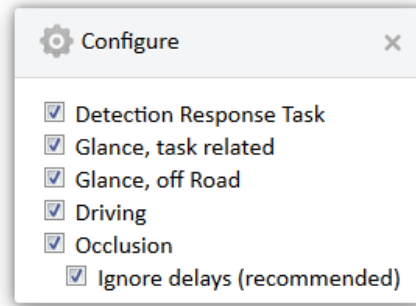


Figure A.2.: Config window



The composed task icon opens the composed task window (Figure A.3). The composed task window is kind of a ‘shopping basket’. Selected subtasks (that compose the task) are collected in the composed task window.

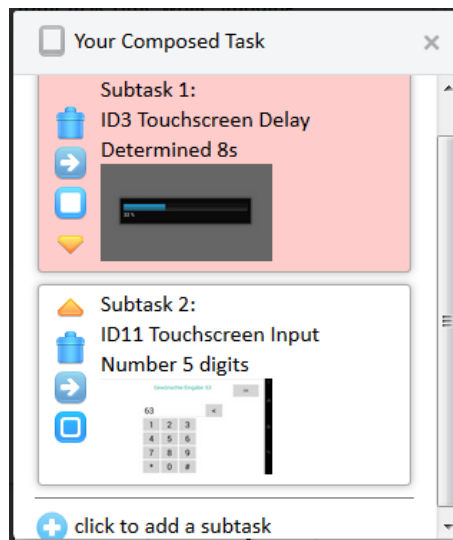


Figure A.3.: Composed task window

The example shows two subtasks in the composed task window (a determined delay of 8 s and the input of five digits via touchscreen). The icons on the left of each subtask can be used for the following actions:



The orange up/down arrows can be used to rearrange the order of the subtasks within the composed task. While the **order is not important** during all calculations, the right order of subtasks can make it easier to model a task or find overlooked subtasks.



The trash can is used to delete a subtask from the collection. Afterward, a dialog (really sure?) tries to catch (unintentional) one-click faults.



The enabled/disabled checkbox can be used to quickly enable or disable a single subtask within a composed task. When watching the calculated task metrics in the main window at the same time, it is possible to obtain an idea of the influence of a subtask on the whole task. A forgotten, disabled subtask can lead to faults while modeling; therefore disabled tasks are clearly marked with a red color in the composed task window.



The plus icon in the toolbar, or on the bottom of the composed task window, can be used to add a subtask. This opens the subtask selection window (Figure A.4)

reset	Symbol-Image	ID	Mode	Type	Subtype	Subsubtype	TTT	TTT	TSOT	
			Filter: All	Filter: All	Filter: All	Filter: All	DRV[s]	[s]	[s]	
			10	Touchscreen	Input	Number	3 digits	4	2.4	2
			11	Touchscreen	Input	Number	5 digits	5.5	4.1	3
			12	Touchscreen	Input	Number	10 digits	9.7	6.3	5.7
			13	Touchscreen	List Selection		first	3.3	2	1.8

Figure A.4.: Add subtask. Subtask selection window

The header of the subtask selection window (Figure A.4) can be used to filter and sort the subtasks. The ‘reset’ button helps to return to default. It can, e.g., be helpful to set the drop down menu of ‘Type’ to ‘List Selection’, so the possible list selections are shown. By clicking on the header (e.g., the Total Glance Time), the subtask can be arranged in ascending/descending order. This can provide a better understanding for metrics and subtasks (educational aspect).



The change icon also opens the subtask selection window, but the subtask is highlighted (Figure A.5). This function has two intended usages:

- It is easily possible to see or inspect the values of a subtask
- A subtask can be changed (e.g., from a 8 s delay to a 4 s delay)

reset	Symbol-Image	ID	Mode	Type	Subtype	Subsubtype	TTT	TTT	TSOT	R	TGT	
			Filter: All	Filter: All	Filter: All	Filter: All	DRV[s]	[s]	[s]		[s]	
			01	Touchscreen	Delay	Determined	2s	2	2	0.79	0.39	0.49
			02	Touchscreen	Delay	Determined	4s	4	4	1.9	0.48	0.76
			03	Touchscreen	Delay	Determined	8s	8	8	3.7	0.47	1.1

Figure A.5.: Change subtask



When moving over the description icon, a tool-tip displays some notes that characterize the subtask (Figure A.6).

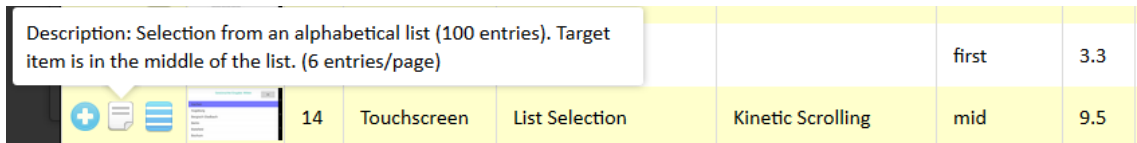


Figure A.6.: Subtask description



A click on the visualization icon opens the glance visualization in a new browser tab or window (Figure A.7).

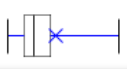


Figure A.7.: Glance visualization

The visualization (Figure A.7) shows the eyes-off-road glance data of the 24 test subjects. Each row (1–24) holds the data of one person. If a subtask was tested in more

than one trial, the visualization can have more tables (for each repeated measurement). A legend on the side clarifies the color coding: Glances to the driving scene are gray. The glance durations away from the driving scene (eyes-off-road-time, EORT) are classified regarding the AOIs detected in this EORT (IVIS: yellow, speedometer: dark blue, combined IVIS & speedometer: purple, unknown targets: black). This can lead to the erroneous assumption that the visualization shows the glance durations to specific AOIs, which is only approximately right. The precise way to understand the graphic is that EORTs are classified by AOI-targets, which can, e.g., additionally include glance durations to unknown targets. Arrows indicate whether a glance has started before the subtask or lasts longer than the subtask (glance split problem). A tool-tip over each segment displays more information about each glance.

reset	Symbol-Image	ID	Mode	Type	Subtype	Subsubtype	TTT	TTT	TS
			Filter: All	Filter: All	Filter: All	Filter: All	DRV[s]	[s]	
		38	Rotary	Input	Number				
		39	Rotary	Input	Number				
		40	Rotary	Input	Number				
		41	Rotary	Input	Alphabetic				
		42	Rotary	Input	Alphabetic	4 chars	14	8.6	6.9



Min	9.3s	Mean	14s
Q1	12s	SD	4.3s
Median	13s	P85	17s
Q3	16s		
Max	26s		

Figure A.8.: Subtask distribution

In the subtask selection window, the mouse can be moved over the metrics to obtain more information (Figure A.8). The tool-tip shows a box plot to acquire an idea of the distribution; a blue cross indicates the 85<sup>th</sup> percentile. A table holds some statistical values (min, max, quartiles, mean, standard deviation and 85<sup>th</sup> percentile). The values in the interface are rounded to two counting digits. This improves clarity and should decrease unrealistic expectations regarding the precision of possible predictions. The file ‘index.html’ holds a global parameter (‘DIGITS’), which can be manually adjusted when needed.

The calculated results of the composed task are visualized in the main window (Figure A.9). Two variants are used for the pie charts: A box plot displays the distribution of the test subjects when the subtask values are combined (blue cross: 85<sup>th</sup> percentile), for some metrics a red line indicates a possible criterion. Statistical values in a table give: min, max, quartiles, mean, standard deviation and 85<sup>th</sup> percentile.

- Values that can be additively summed up (e.g., Total Task Time, Total Glance Time) use a ring for visualization. The sectors indicate the percentage. For example, in a task with two subtasks (TGT mean value 2s and 1s) the 2s subtask has 67%. To obtain the percentages, the mean values of each subtask are used. This is different and separate from the ‘virtual experiment’ calculations.

- Metrics that are based on a weighting mechanism use a complete pie chart. The weighting factor (e.g., Number of Glances) is used for the sector angle. The sector amplitude (i.e. radius) holds the mean subtask value. A gray circle indicates the weighted mean of the subtasks. In the example (Figure A.9), the weighted-mean R-metric of the subtasks was 1. It can be seen that the subtask ‘Typing Alphabetic 4 chars’ (green) has an influence of approximately 30% onto the occlusion metrics and pulls the mean R-ratio (gray circle) with a value of 1.2.

A click on the subtask name in the legends also opens the change window (Figure A.5), which allows changing a subtask to a different subtask or inspecting the values and distributions of a subtask. For some values, the result is bootstrapped several times out of the results from the virtual experiment and compared to a criterion. This can provide an indication regarding how close the result is to a threshold. Like all values, these results are only useful in obtaining an approximate idea or identifying possible problems. The user of the tool should always keep individual plausibility checks in mind.

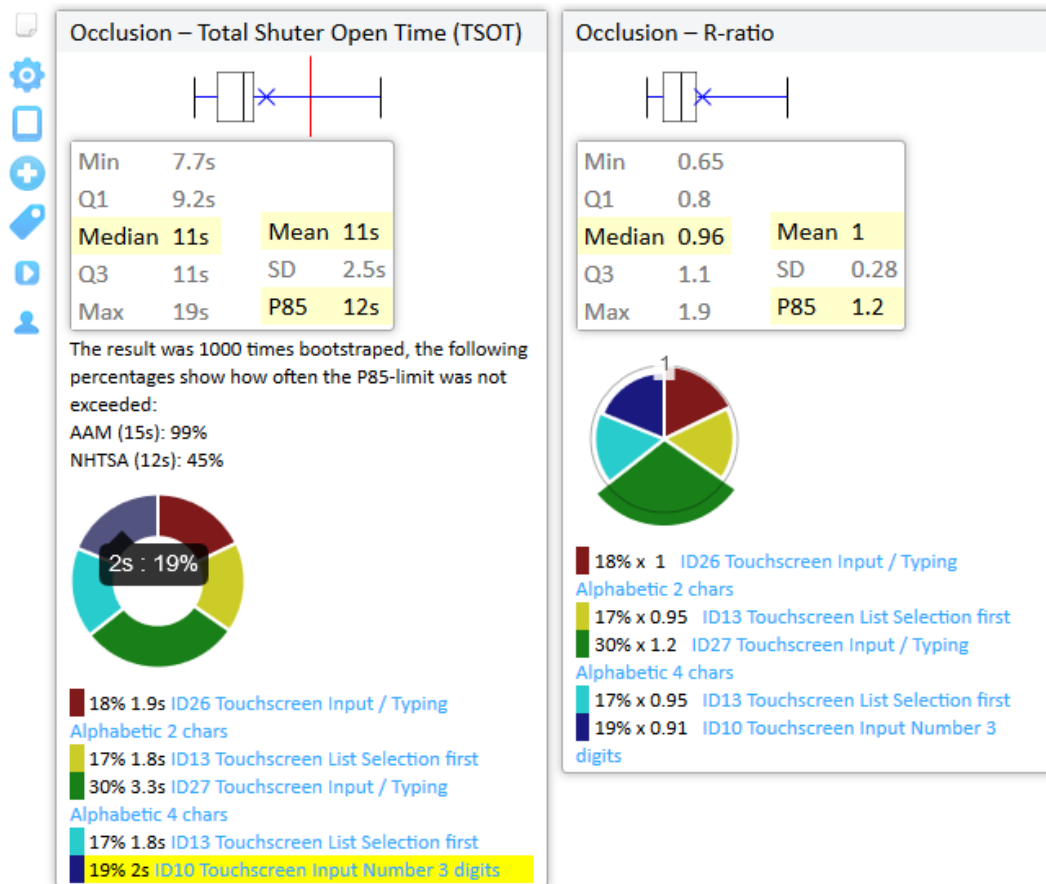


Figure A.9.: Result visualization

---

The remaining three icons in the main toolbar (Figure A.1) are:



When a subtask has been modeled, the link of this icon can be bookmarked or sent to another person. For example, right click on the icon and ‘save as bookmark’ or ‘copy link’ and afterward paste into an email. The link holds all information about the subtasks used (composed task) and the metrics configured. A click on the icon simply opens the link in a new browser tab or window. There the URL can be also copied from the URL bar.

If the task was modeled locally and not online, the link can be look like:

*file:///C:/MyPC/index.html?version=1&drt=true&occ=true&glancetr=true&glanceor=true&drv=true&ignoreDelay=true&s=1030200&S=1110300&*

When the URL should be sent to another person the part before the ‘?’ (*file:///C:/MyPC/index.html*) can be manually replaced by the online tool address: *http://www.distract.one/index.html*



A short video (quick-start tutorial) was recorded and uploaded, which can be reached via this icon. The video is not included when the page is saved locally. It is hosted online.



An about box holds the imprint, license information, some general statements and credits to the used open source libraries.

The page is completely self-contained and holds the database in javascript variables. Therefore, no internet connection is needed when the page is saved locally. The prediction model is available online and licensed as open source. Therefore, it can be tailored to other needs, or parts can be used in other projects. Furthermore, the open source itself is a supplement for this thesis.

---

## B. Appendix – Instructions

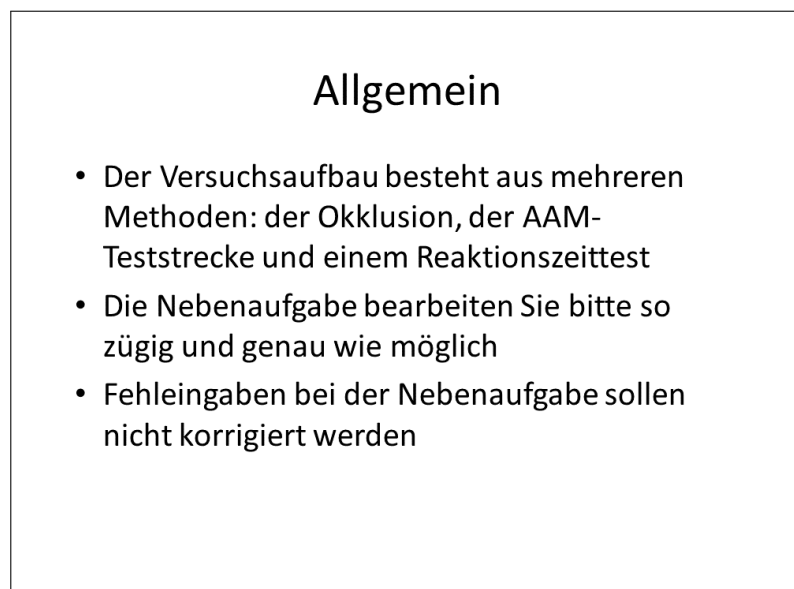


Figure B.1.: Instructions – General

The setup uses different methods: occlusion, AAM driving track and a reaction test

Please work on the secondary task quickly and accurately

Don't correct input errors on the secondary task

---

## Fahraufgabe

Bei der Fahraufgabe fahren Sie mit einem simulierten Auto. Das Fahrzeug hat eine Automatikschaltung, d.h. Sie müssen nur lenken, sowie Gas und Bremse betätigen.

- Bitte **folgen** Sie dem Fahrzeug vor Ihnen und fahren Sie auf dem **rechten Fahrstreifen**.
- Halten Sie bitte einen sicheren Abstand von ca. **50 m**, dies entspricht dem Abstand zwischen den seitlichen Straßenpfosten.
- Versuchen Sie ansonsten bitte die Geschwindigkeit von **80 km/h** zu halten.

Figure B.2.: Instructions – Driving Task (I)

You drive a simulation car with an automated gear shift. You only have to steer, break and accelerate

Please follow the leading vehicle in the right lane

Keep a safe distance of 50 m. That is the distance between reflection posts

Try to keep the speed at 80 km/h

Bitte stellen Sie sich vor, es würde sich um eine echte Autofahrt handeln. Ihre Hauptaufgabe ist es also, sicher zu fahren.

Die Nebenaufgaben, die Ihnen der Versuchsleiter jeweils ansagt, bearbeiten Sie bitte so zügig und genau wie möglich.

Wenn Sie die Nebenaufgabe während des Versuchsdurchlaufs unterbrechen wollen, so machen Sie dies bitte während der Infoscreens.

Figure B.3.: Instructions – Driving Task (II)

Imagine you are really driving. Your main task is to drive safely

Work on the secondary tasks quickly and accurately

If you want to pause the secondary task, please do it during the infoscreen (the instruction screen before each widget)



---

## Okklusion

- Es gibt keine aktive Fahraufgabe
- Es wird alle 1,5sec zwischen klarer und trüber Sicht gewechselt
- Die Aufgaben dürfen und sollen auch bei trüber bzw. verdeckter Sicht fortgesetzt werden
- Die Aufgaben sollen so **zügig und genau** wie möglich durchgeführt werden

Figure B.4.: Instructions – Occlusion

There is no driving task

Every 1.5s, the visibility changes (clear/opaque)

You can and should continue to work on the task when the glasses are opaque

You should work on the task quickly and accurately

## Detection Response Task

- Alle 3-5 Sekunden erhalten Sie einen Vibrationsreiz. Auf diesen sollen Sie so zügig wie möglich per Knopfdruck reagieren.
- Hauptaufgabe bleibt dennoch das Fahren

Figure B.5.: Instructions – Detection Response Task

Every 3–5 seconds you get an vibration stimulus. React as quickly as possible (button press)

The main task is still the driving

---

## **C. Appendix – App Parameters**

```

if(mIsTouch) {/touch

if (mSyllabPacket.stage == 1){/accomodation
mTasks = new ExperimentWidget[] {

//delay determined 2,4,8 seconds
new NumPad((byte) 1, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"23"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPad((byte) 2, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"80333"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPad((byte) 3, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"63"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPad((byte) 4, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"85386"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad((byte) 5, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"73"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad((byte) 6, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"92"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPad((byte) 7, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"80852"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad((byte) 8, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"70"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad((byte) 9, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"42"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//number input 3, 5, 10
new NumPad((byte)10, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"429", "618", "286"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad((byte)11, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"50383", "71579"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad((byte)12, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"049 319 8023"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect((byte)13, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Bielefeld", "Augsburg"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)14, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Koblenz", "Minden"}, 0, ExperimentWidget.DELAY_NONE),
//list select middle, end
new ListSelectNokinetic((byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Krsfeld", "Minden"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelectNokinetic((byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Velbert", "Witten"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker TAP -2, -4, -8
new NumPickerTap((byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"48 (TAP)", "52 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap((byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"46 (TAP)", "55 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerROLL -2, -4, -8
new NumPicker((byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"47 (ROLL)", "51 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)22, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"45 (ROLL)", "54 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)23, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"43 (ROLL)", "59 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum((byte)24, this, mExperimentWidgetLayout, "Slider mit Nummernangabe", new String[]{"50", "35", "70"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis((byte)25, this, mExperimentWidgetLayout, "Slider graphisch", new String[]{"70%", "50%", "35%"}, 0, ExperimentWidget.DELAY_NONE),
//text input 2,4,8 chars
new DirectTextEdit((byte)26, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"ab", "xy", "wa"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit((byte)27, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"dorf", "post"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit((byte)28, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"kontrast"}, 0, ExperimentWidget.DELAY_NONE

};
}
if (mSyllabPacket.stage == 2){/occlusion
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPad((byte) 1, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"24"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPad((byte) 2, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"50764"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPad((byte) 3, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"63"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPad((byte) 4, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"08346"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad((byte) 5, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"87"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad((byte) 6, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"56"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPad((byte) 7, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"97301"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad((byte) 8, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"69"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad((byte) 9, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"54"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//number input 3, 5, 10
new NumPad((byte)10, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"739", "618", "927"}, 0, ExperimentWidget.DELAY_NONE),

```

```

new NumPad((byte)11, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"80415", "17249"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad((byte)12, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"0170 361 602"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect((byte)13, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Bochum", "Augsburg"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)14, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Moers", "Mönchengladbach"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)15, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Ulm", "Würzburg"}, 0, ExperimentWidget.DELAY_NONE),
//list select middle, end
new ListSelectNokinetic((byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Potsdam", "Recklinghausen"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelectNokinetic((byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Stuttgart", "Tübingen"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker TAP -2, -4, -8
new NumPickerTap((byte)18, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"48 (TAP)", "52 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap((byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"54 (TAP)", "46 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap((byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"42 (TAP)", "58 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker ROLL -2, -4, -8
new NumPickerRoll((byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"52 (ROLL)", "48 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)22, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"46 (ROLL)", "54 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)23, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"58 (ROLL)", "42 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum((byte)24, this, mExperimentWidgetLayout, "Slider mit Nummernangabe", new String[]{"55", "20", "85"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis((byte)25, this, mExperimentWidgetLayout, "Slider graphisch", new String[]{"55%", "20%", "85%"}, 0, ExperimentWidget.DELAY_NONE),
//text input 2,4,8 chars
new DirectTextEdit((byte)26, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"ce", "pa", "kl"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit((byte)27, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"kurz", "text"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit((byte)28, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"bankomat"}, 0, ExperimentWidget.DELAY_NONE);
};
}
if (mSllabPacket.stage == 3){//baseline
mTasks = new ExperimentWidget[] {
new NumPad((byte) 1, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"37"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPad((byte) 2, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"60974"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPad((byte) 3, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"51"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPad((byte) 4, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"97824"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad((byte) 5, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"39"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad((byte) 6, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"47"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPad((byte) 7, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"67834"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad((byte) 8, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"72"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad((byte) 9, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"17"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//number input 3, 5, 10
new NumPad((byte)10, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"143", "645", "806"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad((byte)11, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"73928", "48294"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad((byte)12, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"049 310 8026"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect((byte)13, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Berlin", "Bochum"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)14, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Lübeck", "Magdeburg"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)15, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Witten", "Wiesbaden"}, 0, ExperimentWidget.DELAY_NONE),
//list select middle, end
new ListSelectNokinetic((byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Leipzig", "Krefeld"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelectNokinetic((byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Wuppertal", "Ulm"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker TAP -2, -4, -8
new NumPickerTap((byte)18, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"48 (TAP)", "52 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap((byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"54 (TAP)", "46 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap((byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"42 (TAP)", "58 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker ROLL -2, -4, -8
new NumPickerRoll((byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"52 (ROLL)", "48 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)22, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"46 (ROLL)", "54 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)23, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"58 (ROLL)", "42 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum((byte)24, this, mExperimentWidgetLayout, "Slider mit Nummernangabe", new String[]{"90", "25", "50"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis((byte)25, this, mExperimentWidgetLayout, "Slider graphisch", new String[]{"90%", "25%", "50%"}, 0, ExperimentWidget.DELAY_NONE),

```

```

//text input 2,4,8 chars
new DirectTextEdit(byte)26, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"ui", "no", "he"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit(byte)27, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"juni", "haus"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit(byte)28, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"zylinder"}, 0, ExperimentWidget.DELAY_NONE);
};

if (mSllabPacket.stage == 4){//DRT
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPad(byte) 1, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"12"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPad(byte) 2, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"70604"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPad(byte) 3, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"89"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPad(byte) 4, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"38026"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad(byte) 5, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"56"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad(byte) 6, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"20"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPad(byte) 7, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"34902"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad(byte) 8, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"43"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad(byte) 9, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"11"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//number input 3, 5, 10
new NumPad(byte)10, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"934", "101", "429"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad(byte)11, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"75098", "39206"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad(byte)12, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"0171 872 097"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect(byte)13, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Bochum", "Augsburg"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect(byte)14, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Magdeburg", "Koblenz"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect(byte)15, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Velbert", "Wolfsburg"}, 0, ExperimentWidget.DELAY_NONE),
//list select middle, end
new ListSelectKinetic(byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Moers", "Minden"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelectKinetic(byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Wurzburg", "Sollingen"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker TAP -2, -4, -8
new NumPickerTap(byte)18, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"48 (TAP)", "52 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap(byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"54 (TAP)", "46 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap(byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"42 (TAP)", "58 (TAP)"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker ROLL -2, -4, -8
new NumPicker(byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"52 (ROLL)", "48 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker(byte)22, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"46 (ROLL)", "54 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker(byte)23, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"58 (ROLL)", "42 (ROLL)"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum(byte)24, this, mExperimentWidgetLayout, "Slider mit Nummernangabe", new String[]{"20", "45", "85"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis(byte)25, this, mExperimentWidgetLayout, "Slider graphisch", new String[]{"20%", "45%", "85%"}, 0, ExperimentWidget.DELAY_NONE),
//text input 2,4,8 chars
new DirectTextEdit(byte)26, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"ku", "et", "of"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit(byte)27, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"ende", "park"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit(byte)28, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"fabrikat"}, 0, ExperimentWidget.DELAY_NONE);
};
}

if (mSllabPacket.stage == 5){//AAM driving+dikablis
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPad(byte) 1, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"66"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPad(byte) 2, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"90872"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPad(byte) 3, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"24"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPad(byte) 4, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"57303"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad(byte) 5, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"90"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPad(byte) 6, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"14"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPad(byte) 7, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"65278"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad(byte) 8, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"28"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPad(byte) 9, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"96"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//number input 3, 5, 10

```

```

new NumPad((byte)10, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"721", "904", "792"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad((byte)11, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"18365", "40192"}, 0, ExperimentWidget.DELAY_NONE),
new NumPad((byte)12, this, mExperimentWidgetLayout, "Nummerneingabe per Touchscreen", new String[]{"189 101 7350"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect((byte)13, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Bielefeld", "Berlin"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)14, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Mülheim an der Ruhr", "Neuss"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)15, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Trier", "Wolfsburg"}, 0, ExperimentWidget.DELAY_NONE),
//list select middle, end
new ListSelectNokinetic((byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Krefeld", "Minden"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelectNokinetic((byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste", new String[]{"Würzburg", "Wiesbaden"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker TAP -2, -4, -8
new NumPickerTap((byte)18, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"48 (TAP)", "52 (TAP)"}), 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap((byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"54 (TAP)", "46 (TAP)"}), 0, ExperimentWidget.DELAY_NONE),
new NumPickerTap((byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"42 (TAP)", "58 (TAP)"}), 0, ExperimentWidget.DELAY_NONE),
//numpicker ROLL -2, -4, -8
new NumPicker((byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"52 (ROLL)", "48 (ROLL)"}), 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)22, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"46 (ROLL)", "54 (ROLL)"}), 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)23, this, mExperimentWidgetLayout, "Auswahl einer Zahl", new String[]{"58 (ROLL)", "42 (ROLL)"}), 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum((byte)24, this, mExperimentWidgetLayout, "Slider mit Nummernangabe", new String[]{"85", "50", "35"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis((byte)25, this, mExperimentWidgetLayout, "Slider graphisch", new String[]{"85%", "50%", "35%"}, 0, ExperimentWidget.DELAY_NONE),
//text input 2,4,8 chars
new DirectTextEdit((byte)26, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"te", "dl", "vc"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit((byte)27, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"info", "oder"}, 0, ExperimentWidget.DELAY_NONE),
new DirectTextEdit((byte)28, this, mExperimentWidgetLayout, "Eingabe eines Wortes", new String[]{"tagebuch"}, 0, ExperimentWidget.DELAY_NONE);
};
}
} else { // Dreh Druck Steллер / Rotary Knob
if (mSilabPacket.stage == 1){/accomodation
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPicker((byte) 1, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"44"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker((byte) 2, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"52"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker((byte) 3, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"25"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPicker((byte) 4, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"65"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker((byte) 5, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"39"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker((byte) 6, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"47"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPicker((byte) 7, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"59"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker((byte) 8, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"28"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker((byte) 9, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"68"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//num speller 3.5.10
new SpellerNum((byte)10, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"235", "489", "205"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerNum((byte)11, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"87596", "59635"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerNum((byte)12, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"021 369 7512"}, 0, ExperimentWidget.DELAY_NONE),
//char speller 2, 4, 8
new SpellerAbc((byte)13, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"Lp", "BH", "ES"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc((byte)14, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"WAND", "HAUS"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc((byte)15, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"COMPUTER"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect((byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Bielefeld", "Augsburg"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Minden", "Mülheim an der Ruhr"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)18, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Ulm", "Würzburg"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker -2, -4, -8
new NumPicker((byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"48", "52"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"54", "46"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"42", "58"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum((byte)22, this, mExperimentWidgetLayout, "Slider mit Nummernangabe (Dreh-Drücksteller)", new String[]{"50", "90", "35"}, 0, ExperimentWidget.DELAY_NONE),

```

```

//slider vis
new SliderVis(byte)23, this, mExperimentWidgetLayout, "Slider Graphisch (Dreh-Drücksteller)", new String[]{"50%", "90%", "35%"}, 0, ExperimentWidget.DELAY_NONE),
};
}
if (mSlibabPacket.stage == 2){ //occlusion
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPicker(byte) 1, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"55"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker(byte) 2, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"69"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker(byte) 3, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"24"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPicker(byte) 4, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"39"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker(byte) 5, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"58"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker(byte) 6, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"47"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPicker(byte) 7, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"69"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker(byte) 8, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"37"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker(byte) 9, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"59"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//num speller 3.5.10
new SpellerNum(byte)10, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"183","761","964"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerNum(byte)11, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"3593","75986"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerNum(byte)12, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"031 745 7648"}, 0, ExperimentWidget.DELAY_NONE),
//char speller 2, 4, 8
new SpellerAbc(byte)13, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"PE", "AS", "RI"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc(byte)14, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"HANS", "MAUS"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc(byte)15, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"FHRZEUG"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect(byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Bielefeld", "Augsburg"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect(byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Magdeburg", "Moers"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect(byte)18, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Wolfsburg", "Ulm"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker -2, -4, -8
new NumPicker(byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"52","48"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker(byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"46","54"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker(byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"58","42"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum(byte)22, this, mExperimentWidgetLayout, "Slider mit Nummernangabe (Dreh-Drücksteller)", new String[]{"90", "45", "25"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis(byte)23, this, mExperimentWidgetLayout, "Slider Graphisch (Dreh-Drücksteller)", new String[]{"90%", "45%", "25%"}, 0, ExperimentWidget.DELAY_NONE),
};
}
if (mSlibabPacket.stage == 3){ //baseline
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPicker(byte) 1, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"47"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker(byte) 2, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"55"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker(byte) 3, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"26"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPicker(byte) 4, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"59"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker(byte) 5, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"19"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker(byte) 6, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"51"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPicker(byte) 7, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"37"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker(byte) 8, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"46"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker(byte) 9, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"68"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//num speller 3.5.10
new SpellerNum(byte)10, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"145","439","759"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerNum(byte)11, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"34975","12458"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerNum(byte)12, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"0156 7951 43"}, 0, ExperimentWidget.DELAY_NONE),
//char speller 2, 4, 8
new SpellerAbc(byte)13, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"BD", "ZU", "AU"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc(byte)14, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"DORF", "POST"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc(byte)15, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"TANZKURS"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect(byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Augsburg", "Bochum"}, 0, ExperimentWidget.DELAY_NONE),
};
}

```

```

new ListSelect<byte>17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[] {"Koblentz", "Krefeld"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect<byte>18, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[] {"Welbert", "Ulm"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker -2, -4, -8
new NumPicker<byte>19, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"52", "48"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker<byte>20, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"46", "54"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker<byte>21, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"58", "42"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum<byte>22, this, mExperimentWidgetLayout, "Slider mit Nummernangabe (Dreh-Drücksteller)", new String[] {"50", "90", "25"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis<byte>23, this, mExperimentWidgetLayout, "Slider graphisch (Dreh-Drücksteller)", new String[] {"50%", "90%", "25%"}, 0, ExperimentWidget.DELAY_NONE),
};
}

if (mSllabPacket.stage == 4){//DRT
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPicker<byte> 1, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"23"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker<byte> 2, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"33"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker<byte> 3, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"63"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPicker<byte> 4, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"47"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker<byte> 5, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"40"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker<byte> 6, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"62"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPicker<byte> 7, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"59"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker<byte> 8, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"35"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker<byte> 9, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"42"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//num speller 3,5,10
new SpellerNum<byte>10, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[] {"420", "639", "528"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerNum<byte>11, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[] {"84747", "71579"}, 0, ExperimentWidget.DELAY_NONE),
//char speller 2, 4, 8
new SpellerAbc<byte>13, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[] {"MA", "CE", "TI"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc<byte>14, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[] {"HAND", "EURO"}, 0, ExperimentWidget.DELAY_NONE),
new SpellerAbc<byte>15, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[] {"HOCHHAUS"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect<byte>16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[] {"Bochum", "Augsburg"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect<byte>17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[] {"Mannheim", "Lünen"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect<byte>18, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[] {"Welbert", "Trier"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker -2, -4, -8
new NumPicker<byte>19, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"52", "48"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker<byte>20, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"46", "54"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker<byte>21, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"58", "42"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum<byte>22, this, mExperimentWidgetLayout, "Slider mit Nummernangabe (Dreh-Drücksteller)", new String[] {"90", "55", "20"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis<byte>23, this, mExperimentWidgetLayout, "Slider graphisch (Dreh-Drücksteller)", new String[] {"90%", "55%", "20%"}, 0, ExperimentWidget.DELAY_NONE),
};
}

if (mSllabPacket.stage == 5){//AMM driving+dikablis
mTasks = new ExperimentWidget[] {
//delay determined 2,4,8 seconds
new NumPicker<byte> 1, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"71"}, 2000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker<byte> 2, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"56"}, 4000, ExperimentWidget.DELAY_DETERMINED),
new NumPicker<byte> 3, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"24"}, 8000, ExperimentWidget.DELAY_DETERMINED),
//delay indetermined 2,4,8 seconds
new NumPicker<byte> 4, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"49"}, 2000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker<byte> 5, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"62"}, 4000, ExperimentWidget.DELAY_INDETERMINED),
new NumPicker<byte> 6, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"23"}, 8000, ExperimentWidget.DELAY_INDETERMINED),
//delay no indication 2,4,8 seconds
new NumPicker<byte> 7, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"65"}, 2000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker<byte> 8, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"55"}, 4000, ExperimentWidget.DELAY_NOINDICATION),
new NumPicker<byte> 9, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[] {"62"}, 8000, ExperimentWidget.DELAY_NOINDICATION),
//num speller 3,5,10
new SpellerNum<byte>10, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[] {"425", "175", "345"}, 0, ExperimentWidget.DELAY_NONE),

```



```

new SpinnerNum((byte)11, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"95648", "47514"}, 0, ExperimentWidget.DELAY_NONE),
new SpinnerNum((byte)12, this, mExperimentWidgetLayout, "Eingabe einer Zahl per Dreh-Drücksteller", new String[]{"0195 275 457"}, 0, ExperimentWidget.DELAY_NONE),
//char spinner 2, 4, 8
new SpinnerAbc((byte)13, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"AM", "HA", "DE"}, 0, ExperimentWidget.DELAY_NONE),
new SpinnerAbc((byte)14, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"BAW", "RAUN"}, 0, ExperimentWidget.DELAY_NONE),
new SpinnerAbc((byte)15, this, mExperimentWidgetLayout, "Eingabe von Buchstaben per Dreh-Drücksteller", new String[]{"TESAFILM"}, 0, ExperimentWidget.DELAY_NONE),
//list select first page, middle, end
new ListSelect((byte)16, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Bergisch Gladbach", "Bielefeld"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)17, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Neuss", "Oberhausen"}, 0, ExperimentWidget.DELAY_NONE),
new ListSelect((byte)18, this, mExperimentWidgetLayout, "Auswahl aus einer Liste per Dreh-Drücksteller", new String[]{"Solingen", "Wiesbaden"}, 0, ExperimentWidget.DELAY_NONE),
//numpicker -2, -4, -8
new NumPicker((byte)19, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"52", "48"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)20, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"46", "54"}, 0, ExperimentWidget.DELAY_NONE),
new NumPicker((byte)21, this, mExperimentWidgetLayout, "Auswahl einer Zahl per Dreh-Drücksteller", new String[]{"58", "42"}, 0, ExperimentWidget.DELAY_NONE),
//slider num
new SliderNum((byte)22, this, mExperimentWidgetLayout, "Slider mit Nummernangabe (Dreh-Drücksteller)", new String[]{"50", "20", "85"}, 0, ExperimentWidget.DELAY_NONE),
//slider vis
new SliderVis((byte)23, this, mExperimentWidgetLayout, "Slider graphisch (Dreh-Drücksteller)", new String[]{"50%", "20%", "85%"}, 0, ExperimentWidget.DELAY_NONE),
};
}

```

---

## **D. Appendix – Evaluation Results – Extended Data**

All boxplots are drawn from the minimum to the maximum (whiskers) without outlier calculations.

## D.1. Total Task Time Unoccluded

For the Total Task Time unoccluded (also know as TTT static), the TTT of the first and second trial has been averaged for each subject before calculating the boxplots (Figure D.1); therefore  $N = 24$  for each boxplot. The data point to assess the prediction is the median (Table D.1).

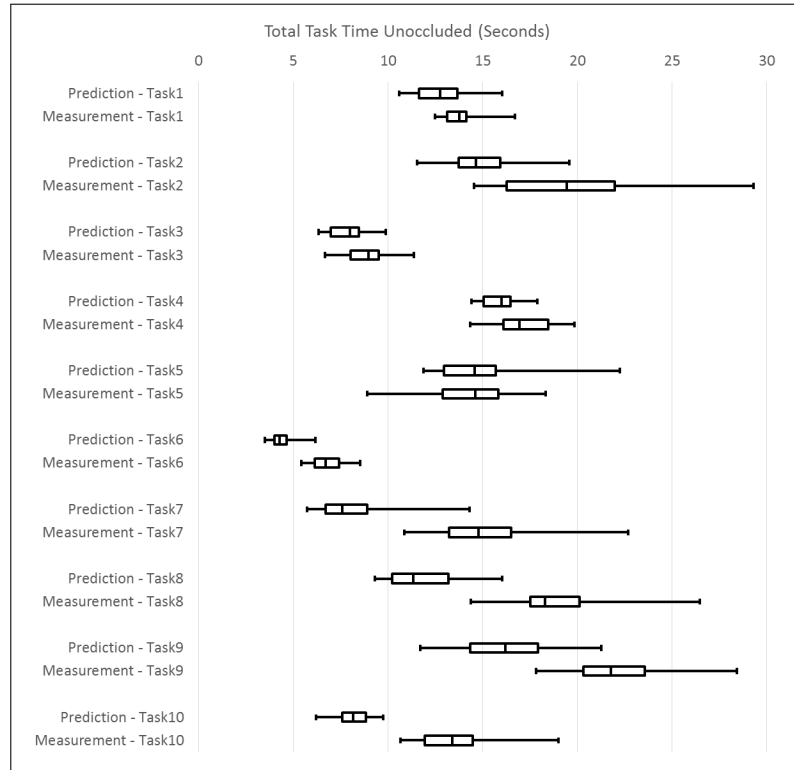


Figure D.1.: Total Task Time unoccluded – Boxplot

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	12.77	13.76	-0.99	-7.21%
Task2	14.67	19.44	-4.77	-24.55%
Task3	7.99	8.98	-0.99	-11.03%
Task4	16.02	16.96	-0.94	-5.53%
Task5	14.60	14.63	-0.03	-0.22%
Task6	4.27	6.71	-2.44	-36.36%
Task7	7.60	14.78	-7.18	-48.58%
Task8	11.35	18.30	-6.95	-37.99%
Task9	16.20	21.77	-5.57	-25.59%
Task10	8.16	13.39	-5.23	-39.03%

Table D.1.: Total Task Time unoccluded – Data table – Median

## D.2. Total Shutter Open Time

For the Total Shutter Open Time, the TSOT of the first and second trial has been averaged for each subject before calculating the boxplots (Figure D.2); therefore  $N = 24$  for each boxplot. The data point to assess the prediction is the median (Table D.2).

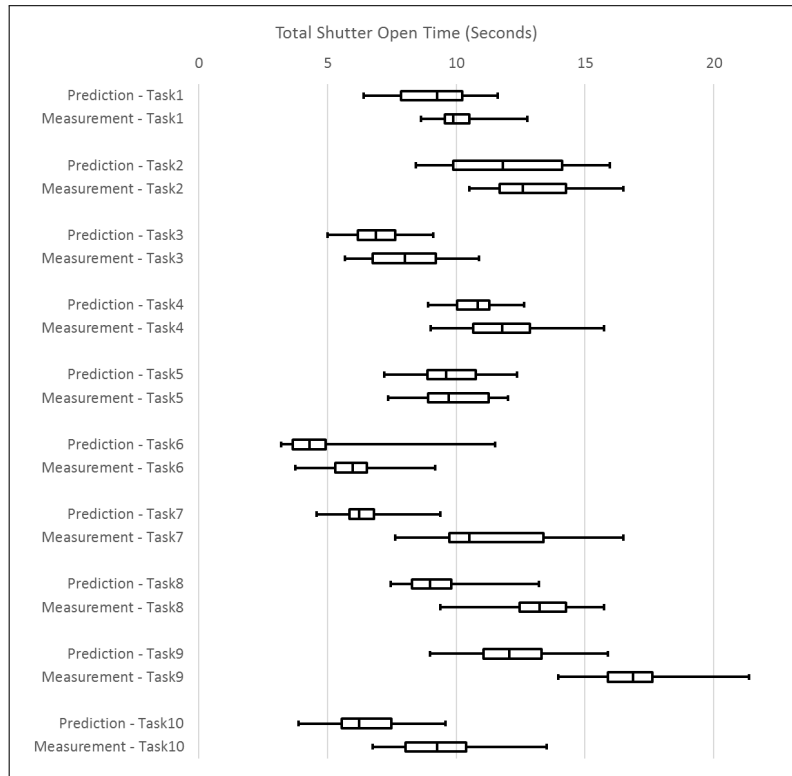


Figure D.2.: Total Shutter Open Time – Boxplot

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	9.27	9.89	-0.62	-6.26%
Task2	11.82	12.59	-0.77	-6.10%
Task3	6.88	8.01	-1.13	-14.14%
Task4	10.83	11.78	-0.94	-8.03%
Task5	9.60	9.71	-0.11	-1.15%
Task6	4.30	5.97	-1.67	-27.94%
Task7	6.22	10.51	-4.29	-40.84%
Task8	8.97	13.24	-4.27	-32.24%
Task9	12.07	16.86	-4.79	-28.42%
Task10	6.23	9.26	-3.03	-32.75%

Table D.2.: Total Shutter Open Time – Data table – Median

## D.3. Occlusion R-ratio

For the R-ratio, the averaged TSOT of the first and second trial is divided by the averaged TTT unoccluded for each subject before calculating the boxplots (Figure D.3); therefore  $N = 24$  for each boxplot. The data point to assess the prediction is the median (Table D.3).

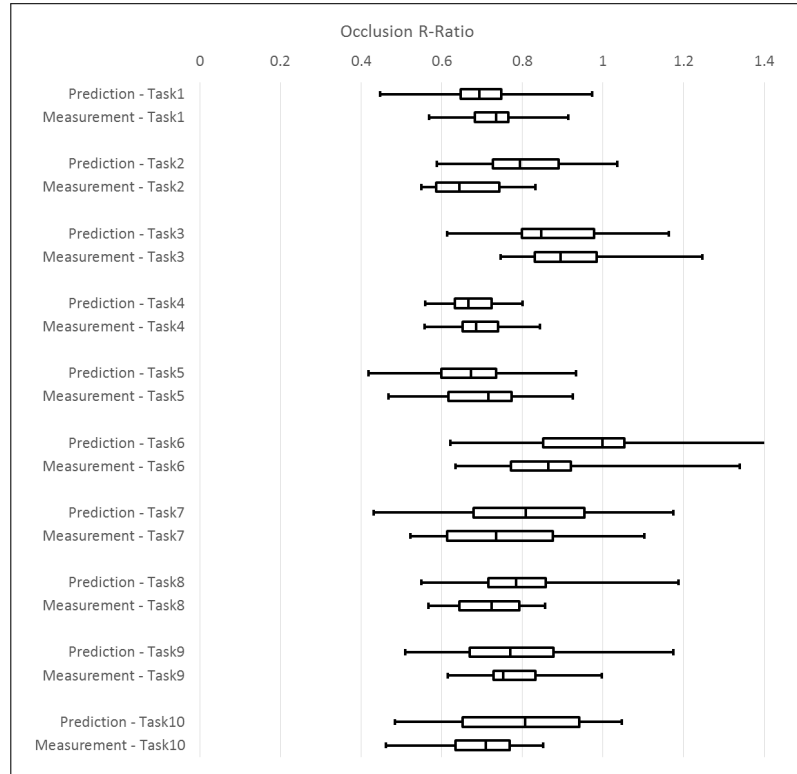


Figure D.3.: Occlusion R-ratio – Boxplot (cut off, outlier 2.8, prediction Task 8)

	Prediction	Measurement	Difference	Relative
Task1	0.693	0.736	-0.042	-5.76%
Task2	0.795	0.644	0.151	23.42%
Task3	0.848	0.896	-0.048	-5.35%
Task4	0.667	0.685	-0.019	-2.70%
Task5	0.673	0.716	-0.043	-5.99%
Task6	0.998	0.864	0.134	15.51%
Task7	0.809	0.735	0.074	10.04%
Task8	0.784	0.724	0.060	8.32%
Task9	0.770	0.753	0.017	2.19%
Task10	0.807	0.710	0.097	13.65%

Table D.3.: Occlusion R-ratio – Data table – Median

## D.4. Total Task Time While Driving

The first and second trials are averaged for each subject before calculating the boxplots (Figure D.4); therefore  $N = 24$  for each boxplot. The data point to assess the prediction is the median (Table D.4).

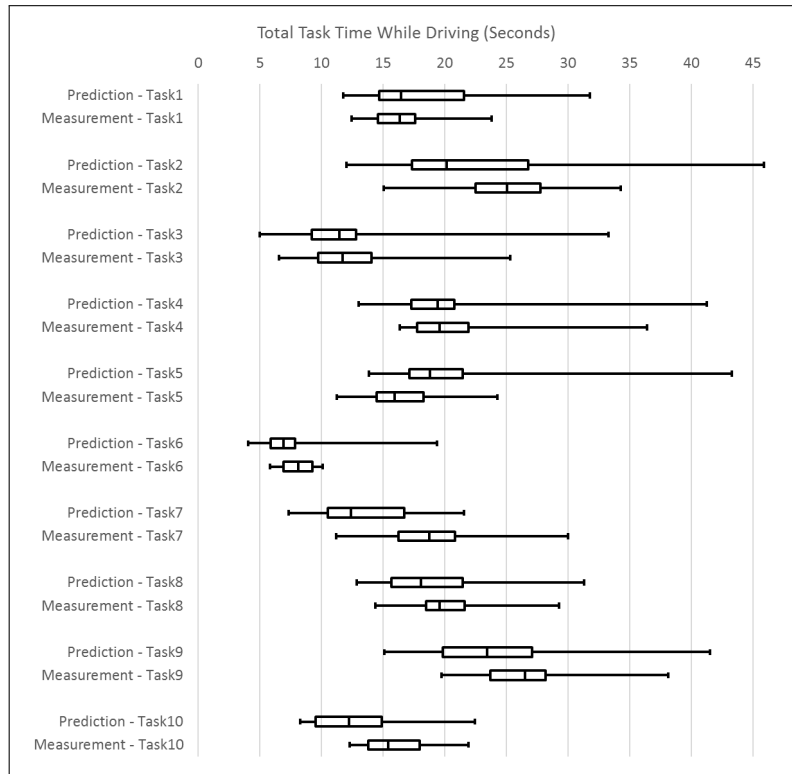


Figure D.4.: Total Task Time while driving – Boxplot

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	16.44	16.36	0.08	0.49%
Task2	20.15	25.06	-4.91	-19.59%
Task3	11.44	11.72	-0.28	-2.39%
Task4	19.44	19.57	-0.13	-0.66%
Task5	18.80	15.92	2.88	18.09%
Task6	6.92	8.13	-1.21	-14.88%
Task7	12.37	18.74	-6.37	-33.99%
Task8	18.04	19.60	-1.56	-7.96%
Task9	23.45	26.50	-3.05	-11.51%
Task10	12.21	15.42	-3.21	-20.82%

Table D.4.: Total Task Time while driving – Data table – Median

## D.5. Total Glance Time to IVIS

The first and second trials are averaged for each subject before calculating the boxplots (Figure D.5); therefore  $N = 24$  for each boxplot. The data point to assess the prediction is the median (Table D.5).

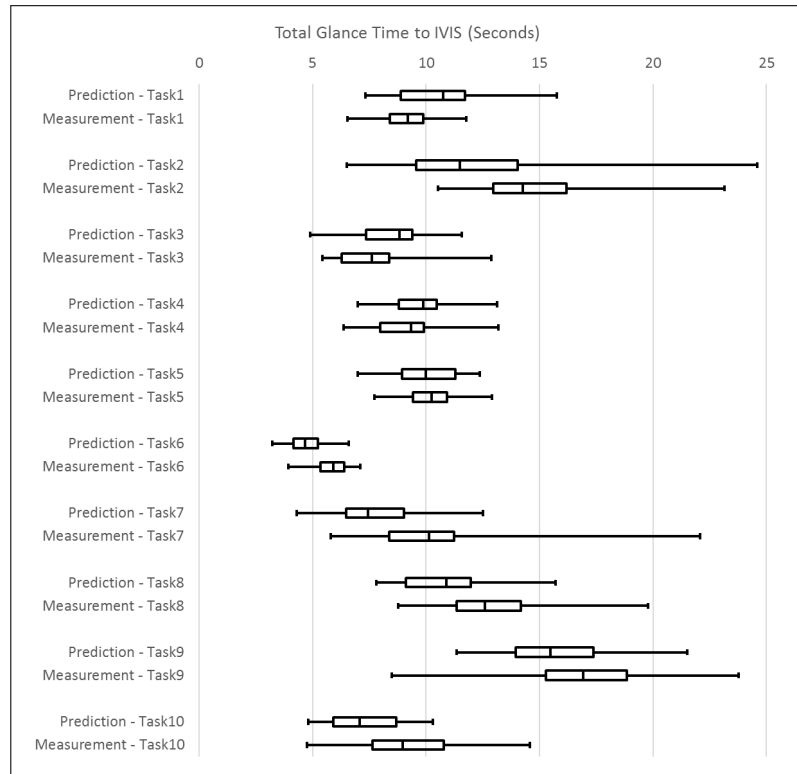


Figure D.5.: Total Glance Time to IVIS – Boxplot

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	10.75	9.19	1.56	16.97%
Task2	11.50	14.27	-2.77	-19.41%
Task3	8.82	7.60	1.22	16.05%
Task4	9.87	9.34	0.53	5.67%
Task5	10.00	10.23	-0.23	-2.25%
Task6	4.68	5.90	-1.22	-20.68%
Task7	7.44	10.14	-2.70	-26.63%
Task8	10.89	12.59	-1.70	-13.50%
Task9	15.47	16.92	-1.45	-8.57%
Task10	7.08	8.97	-1.89	-21.07%

Table D.5.: Total Glance Time to IVIS – Data table – Median

## D.6. Number of Glances to IVIS

The first and second trials of the fractional glances are averaged for each subject before calculating the boxplots (Figure D.6); therefore  $N = 24$  for each boxplot. The data point to assess the prediction is the median (Table D.6).

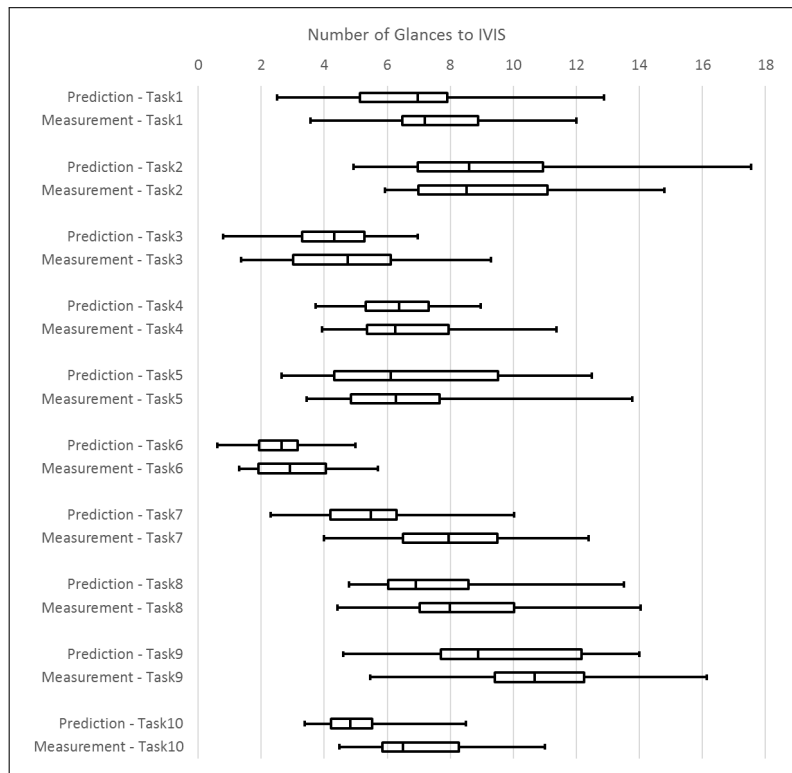


Figure D.6.: Number of Glances to IVIS – Boxplot

	Prediction	Measurement	Difference	Relative
Task1	6.97	7.19	-0.22	-3.08%
Task2	8.60	8.52	0.08	0.92%
Task3	4.33	4.74	-0.42	-8.78%
Task4	6.38	6.26	0.13	2.03%
Task5	6.12	6.27	-0.15	-2.45%
Task6	2.64	2.91	-0.27	-9.28%
Task7	5.47	7.95	-2.48	-31.17%
Task8	6.90	7.98	-1.09	-13.59%
Task9	8.88	10.68	-1.80	-16.89%
Task10	4.83	6.50	-1.68	-25.77%

Table D.6.: Number of Glances to IVIS – Data table – Median



## D.7. Single Glance Duration to IVIS

The SGD is calculated by dividing the TGT by the fractional glances for the first and second trial; then both SGDs (first and second) are averaged for each subject before calculating the boxplots (Figure D.7); therefore  $N = 24$  for each boxplot. The data point to assess the prediction is the median (Table D.7).

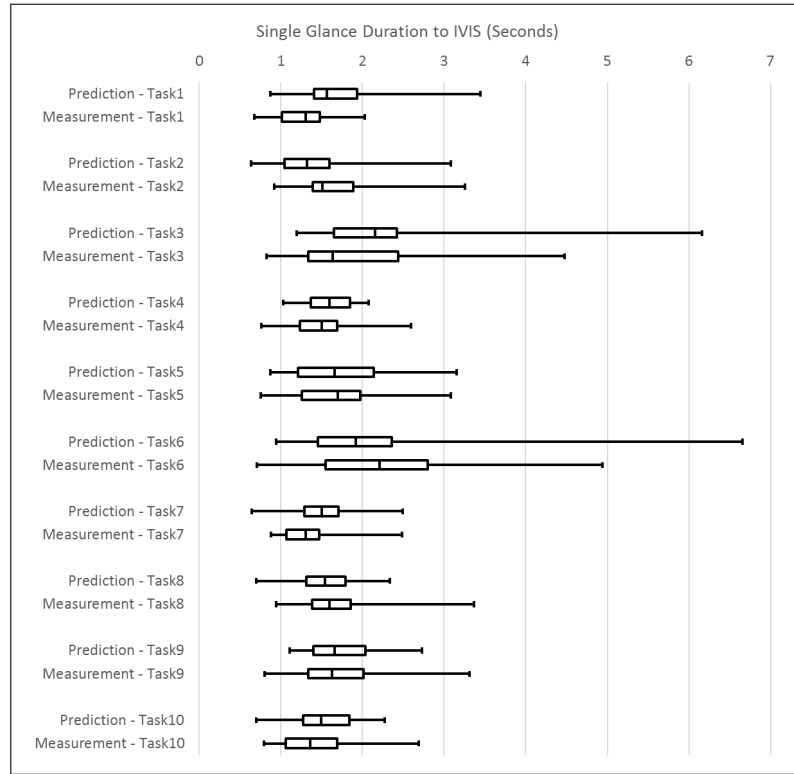


Figure D.7.: Single Glance Duration to IVIS – Boxplot

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	1.567	1.304	0.263	20.12%
Task2	1.319	1.512	-0.193	-12.78%
Task3	2.153	1.636	0.517	31.60%
Task4	1.593	1.500	0.093	6.21%
Task5	1.662	1.698	-0.036	-2.14%
Task6	1.922	2.211	-0.289	-13.06%
Task7	1.498	1.301	0.197	15.15%
Task8	1.544	1.592	-0.048	-3.04%
Task9	1.655	1.625	0.030	1.84%
Task10	1.492	1.360	0.132	9.74%

Table D.7.: Single Glance Duration to IVIS – Data table – Median

## D.8. Total Eyes-Off-Road Time

The TEORT is calculated by averaging the first and second trial;  $N = 24$  for each box-plot (Figure D.8). The NHTSA guideline, which uses TEORT, would use a single-shot measurement (one trial); the evaluation experiment uses two trials and averaging. The data point to assess the prediction is the median (Table D.8).

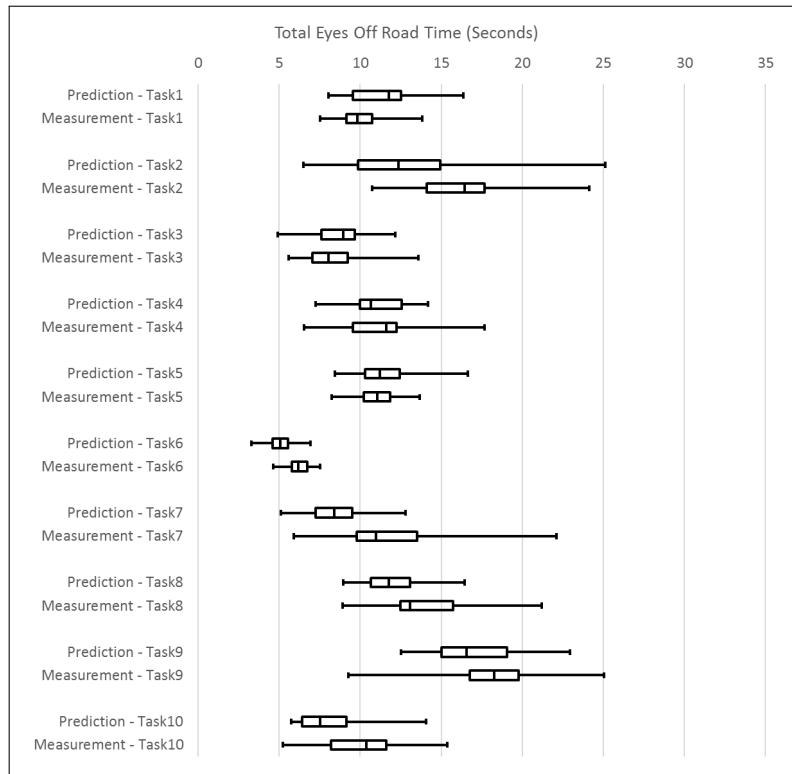


Figure D.8.: Total Eyes-Off-Road Time – Boxplot

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	11.78	9.84	1.94	19.72%
Task2	12.35	16.44	-4.09	-24.88%
Task3	8.95	8.02	0.93	11.60%
Task4	10.64	11.59	-0.95	-8.20%
Task5	11.22	11.07	0.15	1.36%
Task6	5.06	6.19	-1.13	-18.26%
Task7	8.39	10.96	-2.57	-23.45%
Task8	11.75	13.06	-1.31	-10.03%
Task9	16.56	18.27	-1.71	-9.36%
Task10	7.51	10.37	-2.86	-27.58%

Table D.8.: Total Eyes-Off-Road Time – Data table – Median

## D.9. Number of Glances, Eyes-Off-Road

The Number of Glances (eyes-off-road) is calculated by averaging the first and second trial;  $N = 24$  for each boxplot (Figure D.9). The NHTSA guideline, which uses eyes-off-road metrics, would use a single-shot measurement (one trial). The Number of Glances is important to calculate the number of long glances that are allowed. The fractional glance metric is not used for the measurement; fractional glances are counted as full glances. The data point to assess the prediction is the median (Table D.9).

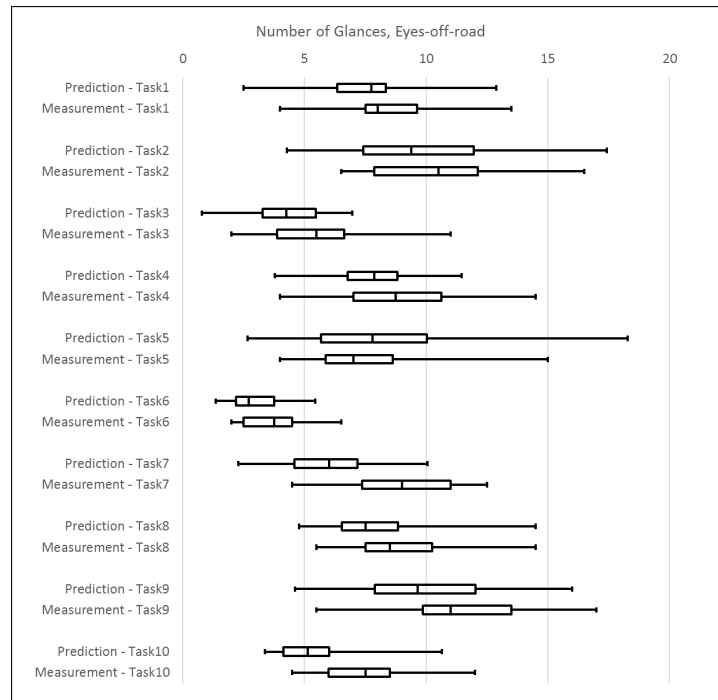


Figure D.9.: Number of Glances, eyes-off-road – Boxplot

	Prediction	Measurement	Difference	Relative
Task1	7.76	8	-0.24	-3.00%
Task2	9.38	10.5	-1.12	-10.69%
Task3	4.24	5.5	-1.26	-22.84%
Task4	7.87	8.75	-0.88	-10.03%
Task5	7.80	7	0.80	11.49%
Task6	2.71	3.75	-1.04	-27.79%
Task7	6.01	9	-2.99	-33.20%
Task8	7.51	8.5	-0.99	-11.69%
Task9	9.66	11	-1.34	-12.16%
Task10	5.13	7.5	-2.38	-31.67%

Table D.9.: Number of Glances, eyes-off-road – Data table – Median

## D.10. Single Glance Duration, Eyes-Off-Road

This Single Glance Duration (eyes-off-road; TEORT/NOG) is based on averaging the outcome of two trials;  $N = 24$  for each boxplot (Figure D.10). The NHTSA guideline, which uses eyes-off-road metrics, would use a single-shot measurement (one trial). The fractional glance metric is not used for the measurements; fractional glances are counted as full glances. The data point to assess the prediction is the median (Table D.10).

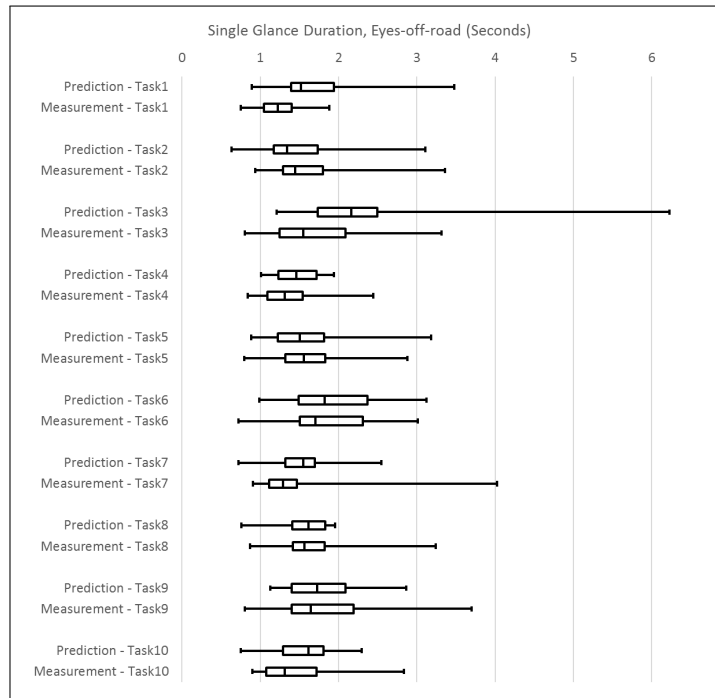


Figure D.10.: Single Glance Duration, eyes-off-road – Boxplot

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	1.517	1.226	0.291	23.71%
Task2	1.345	1.448	-0.103	-7.12%
Task3	2.163	1.548	0.615	39.70%
Task4	1.459	1.310	0.149	11.35%
Task5	1.504	1.557	-0.053	-3.41%
Task6	1.820	1.708	0.112	6.54%
Task7	1.552	1.290	0.262	20.32%
Task8	1.615	1.564	0.051	3.29%
Task9	1.730	1.646	0.084	5.11%
Task10	1.618	1.317	0.302	22.90%

Table D.10.: Single Glance Duration, eyes-off-road – Data table – Median

## D.11. DRT Deterioration

The median DRT reaction time of the first and second trial is averaged. This results is compared to the baseline dual-task reaction time (driving + DRT) of each subject to obtain the deterioration.  $N = 23$  for each boxplot (Figure D.11). The data point to assess the prediction is the median (Table D.11).

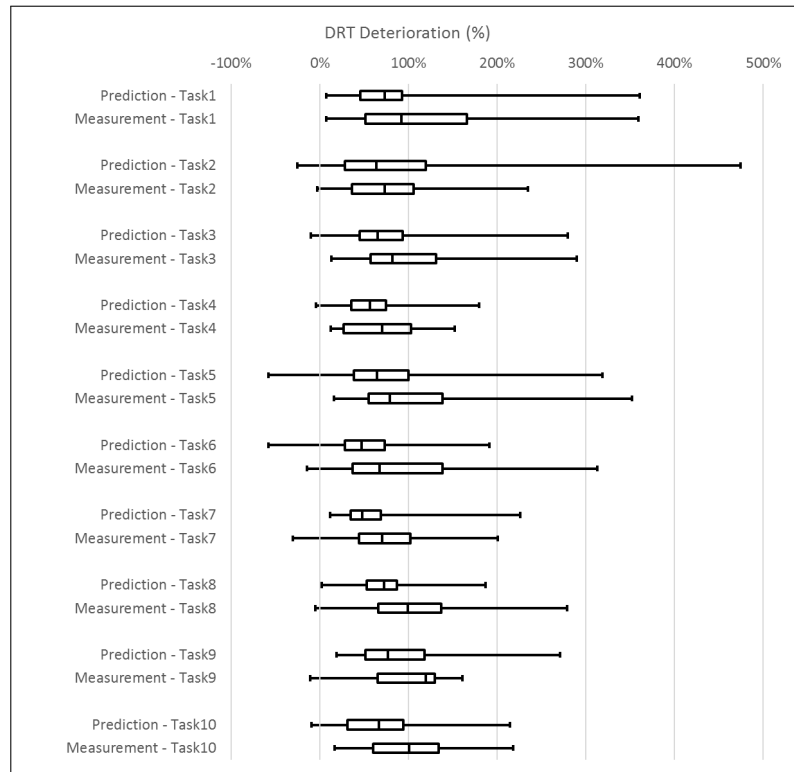


Figure D.11.: DRT deterioration – Boxplot

	Prediction	Measurement	Difference [p.p.]	Relative
Task1	73.2%	92.3%	-19.1	-20.71%
Task2	63.7%	73.3%	-9.6	-13.07%
Task3	65.0%	81.7%	-16.7	-20.47%
Task4	56.6%	70.6%	-14.1	-19.91%
Task5	64.4%	79.2%	-14.7	-18.59%
Task6	46.9%	67.6%	-20.7	-30.66%
Task7	48.1%	70.7%	-22.6	-31.94%
Task8	72.7%	99.6%	-26.8	-26.96%
Task9	77.0%	119.8%	-42.8	-35.75%
Task10	66.4%	100.5%	-34.1	-33.90%

Table D.11.: DRT deterioration – Data table – Median

## D.12. DLP Deterioration

The DLP value of the first and second trial are averaged. This result is compared to the DLP baseline performance (just driving) of each subject to obtain the deterioration.  $N = 24$  for each boxplot (Figure D.12). The data point to assess the prediction is the median (Table D.12).

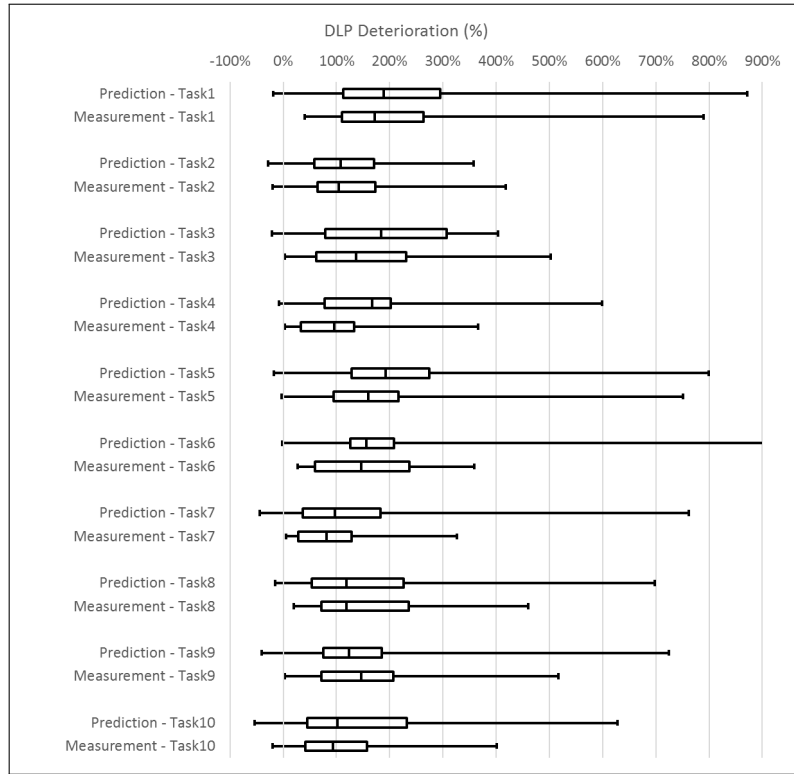


Figure D.12.: DLP deterioration – Boxplot (cut off, outlier 1131%, prediction Task 6)

	Prediction	Measurement	Difference [p.p.]	Relative
Task1	189.2%	171.4%	17.8	10.39%
Task2	107.3%	103.8%	3.5	3.39%
Task3	183.6%	136.5%	47.1	34.51%
Task4	167.2%	96.0%	71.2	74.12%
Task5	191.8%	160.0%	31.8	19.91%
Task6	155.9%	146.5%	9.4	6.41%
Task7	97.4%	81.1%	16.2	20.02%
Task8	118.4%	119.2%	-0.8	-0.64%
Task9	123.6%	146.4%	-22.8	-15.59%
Task10	101.5%	93.9%	7.6	8.07%

Table D.12.: DLP deterioration – Data table – Median

## D.13. DFH Deterioration

The DFH value of the first and second trial are averaged. This result is compared to the DFH baseline performance (just driving) of each subject to obtain the deterioration.  $N = 24$  for each boxplot (Figure D.13). The data point to assess the prediction is the median (Table D.13).

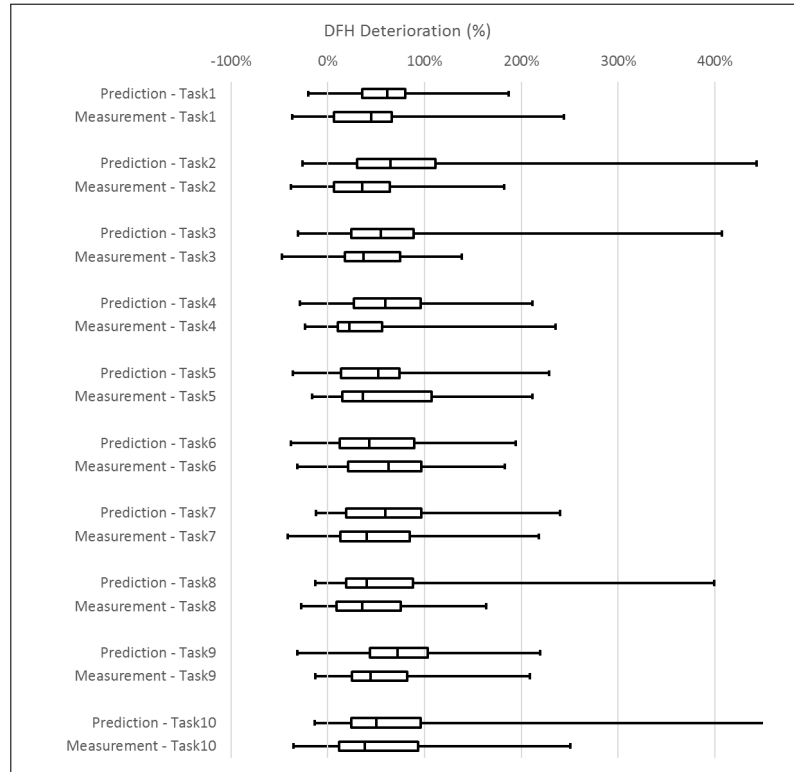


Figure D.13.: DFH deterioration – Boxplot (cut off, outlier 466%, prediction Task 10)

	Prediction	Measurement	Difference [p.p.]	Relative
Task1	61.2%	44.9%	16.3	36.17%
Task2	64.6%	35.3%	29.4	83.22%
Task3	54.6%	36.8%	17.8	48.38%
Task4	59.2%	22.5%	36.7	163.54%
Task5	52.1%	36.5%	15.6	42.64%
Task6	42.7%	62.5%	-19.8	-31.64%
Task7	59.5%	40.5%	19.0	46.94%
Task8	40.0%	35.9%	4.1	11.56%
Task9	72.0%	44.3%	27.7	62.59%
Task10	49.9%	38.2%	11.7	30.60%

Table D.13.: DFH deterioration – Data table – Median

## D.14. 85<sup>th</sup> Percentile Predictions and Bootstrapped Results

For some metrics, the 85<sup>th</sup> percentile is more important than the median or average. Some predictions also include some bootstrapping results. These data are assessed in the following sections. The interpolating Excel quantile function (p=0.85) is used to get the 85<sup>th</sup> percentile of the measurements.

The bootstrapped results are condensed into a percentage concerning the likelihood of fulfilling a criterion. This percentage is subjectively compared to the outcome of the measurements (Table 4.2, p. 97) and reviewed with the signs: (+) ok, (o) reasonable, (-) misleading.

### D.14.1. Total Shutter Open Time – 85<sup>th</sup> Percentile

This is related to the results on p. 144.

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	10.72	11.28	-0.56	-4.99%
Task2	14.83	14.88	-0.05	-0.31%
Task3	7.78	9.91	-2.13	-21.47%
Task4	11.69	13.54	-1.85	-13.66%
Task5	11.03	11.77	-0.74	-6.31%
Task6	5.36	7.18	-1.82	-25.32%
Task7	7.12	13.89	-6.77	-48.74%
Task8	10.68	14.66	-3.98	-27.16%
Task9	14.59	18.50	-3.91	-21.12%
Task10	8.17	10.54	-2.36	-22.43%

Table D.14.: P85 Total Shutter Open Time – Data table

Table D.14 is based on 85<sup>th</sup> percentiles while Table D.2 (p. 144) uses the median.

The prediction of the online tool also bootstraps the predicted results and calculates how often (percentage) these are lower than 15s (AAM) or 12s (NHTSA). Table D.15 reviews this indicator for the AAM limit, and Table D.16 for the NHTSA limit.



	Bootstrap Indicator (%)	Measurement Result	Review (+/o/-)
Task1	100%	ok (11.3)	+
Task2	62.90%	ok (14.9)	o
Task3	100%	ok (9.9)	+
Task4	100%	ok (13.5)	+
Task5	100%	ok (11.8)	+
Task6	100%	ok (7.2)	+
Task7	100%	ok (13.9)	+
Task8	100%	ok (14.7)	+
Task9	65.50%	FAIL (18.5)	o
Task10	100%	ok (10.5)	+

Table D.15.: Bootstrapping Total Shutter Open Time AAM 15s Limit – Data table

	Bootstrap Indicator (%)	Measurement Result	Review (+/o/-)
Task1	100%	ok 23/24	+
Task2	0%	FAIL 8/23	+
Task3	100%	ok 24/24	+
Task4	86.20%	FAIL 13/24	o
Task5	94.80%	ok 24/24	o
Task6	100%	ok 24/24	+
Task7	100%	FAIL 15/24	-
Task8	87.50%	FAIL 6/24	-
Task9	0%	FAIL 0/24	+
Task10	100%	ok 23/24	+

Table D.16.: Bootstrapping Total Shutter Open Time NHTSA 12s limit – Data table

### D.14.2. Total Glance Time – 85<sup>th</sup> Percentile

This is related to the results on p. 147.

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	12.36	10.73	1.63	15.22%
Task2	16.16	16.62	-0.46	-2.74%
Task3	9.61	9.08	0.54	5.89%
Task4	10.82	11.02	-0.20	-1.80%
Task5	11.48	11.04	0.44	3.99%
Task6	5.51	6.70	-1.20	-17.83%
Task7	9.52	13.79	-4.27	-30.98%
Task8	12.77	14.72	-1.95	-13.24%
Task9	18.22	20.76	-2.54	-12.25%
Task10	9.10	11.52	-2.41	-20.95%

Table D.17.: P85 Total Glance Time – Data table

	Bootstrap Indicator (%)	Measurement Result	Review (+/o/-)
Task1	100%	ok (10.7)	+
Task2	94%	ok (16.6)	o
Task3	100%	ok (9.1)	+
Task4	100%	ok (11.0)	+
Task5	100%	ok (11.0)	+
Task6	100%	ok (6.7)	+
Task7	100%	ok (13.8)	+
Task8	100%	ok (14.7)	+
Task9	97.80%	FAIL (20.8)	o
Task10	100%	ok (11.5)	+

Table D.18.: Bootstrapping Total Glance Time AAM 20 s limit – Data Table

Table D.17 is based on 85<sup>th</sup> percentiles while Table D.5 (p. 147) uses the median. The prediction of the online tool also bootstraps the predicted results and calculates how often (percentage) these are lower than 20 s (AAM). Table D.18 reviews this indicator.

**D.14.3. Single Glance Duration to IVIS – 85<sup>th</sup> Percentile**

This is related to the results on p. 149.

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	2.348	1.589	0.759	47.75%
Task2	1.715	2.248	-0.533	-23.70%
Task3	2.611	3.475	-0.864	-24.86%
Task4	1.948	1.765	0.183	10.35%
Task5	2.271	2.362	-0.091	-3.86%
Task6	2.534	3.983	-1.449	-36.37%
Task7	1.877	1.636	0.241	14.70%
Task8	1.878	1.958	-0.080	-4.07%
Task9	2.344	2.456	-0.112	-4.55%
Task10	1.980	1.762	0.218	12.38%

Table D.19.: P85 Single Glance Duration to IVIS – Data table

	Bootstrap Indicator (%)	Measurement Result	Review (+/o/-)
Task1	23.60%	ok (1.59)	–
Task2	79.70%	FAIL (2.25)	o
Task3	0%	FAIL (3.48)	+
Task4	89%	ok (1.77)	+
Task5	6.60%	FAIL (2.36)	+
Task6	0%	FAIL (3.98)	+
Task7	70.10%	ok (1.64)	o
Task8	99.30%	ok (1.96)	o
Task9	8.90%	FAIL (2.46)	+
Task10	78.10%	ok (1.76)	o

Table D.20.: Bootstrapping Single Glance Duration to IVIS AAM 2 s limit – Data table

Table D.19 is based on 85<sup>th</sup> percentiles while Table D.7 (p. 149) uses the median. The prediction of the online tool also bootstraps the predicted results and calculates how often (percentage) these are lower than 2 s (AAM). Table D.20 reviews this indicator.

### D.14.4. Total Eyes-Off-Road Time – 85<sup>th</sup> Percentile

This is related to the results on p. 150.

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	13.31	11.73	1.58	13.47%
Task2	18.41	18.74	-0.33	-1.75%
Task3	10.04	9.68	0.36	3.73%
Task4	12.96	13.23	-0.27	-2.03%
Task5	13.26	12.16	1.10	9.07%
Task6	5.81	6.95	-1.14	-16.41%
Task7	10.3	15.05	-4.75	-31.56%
Task8	14.63	16.03	-1.40	-8.73%
Task9	19.75	22.26	-2.51	-11.28%
Task10	10.1	12.39	-2.29	-18.47%

Table D.21.: P85 Total Eyes-Off-Road Time – Data table

	Bootstrap Indicator (%)	Measurement Result Trial1	Measurement Result Trial2	Review (+/o/-)
Task1	1%	FAIL 19/24	ok 23/24	-
Task2	0.00%	FAIL 2/24	FAIL 2/24	+
Task3	99.70%	ok 23/24	ok 23/24	+
Task4	0.70%	FAIL 15/24	FAIL 16/24	+
Task5	2.30%	FAIL 17/24	ok 21/24	-
Task6	100%	ok 24/24	ok 24/24	+
Task7	98.80%	FAIL 13/24	FAIL 17/24	-
Task8	0%	FAIL 5/24	FAIL 8/24	+
Task9	0%	FAIL 1/24	FAIL 1/24	+
Task10	98.20%	FAIL 17/24	ok 21/24	o

Table D.22.: Bootstrapping Total Eyes-Off-Road Time NHTSA 12s limit – Data table

Table D.21 is based on 85<sup>th</sup> percentiles while Table D.8 (p. 150) uses the median. The prediction of the online tool also bootstraps the predicted results and calculates how often (percentage) these are lower than 12s (NHTSA). Table D.22 reviews this indicator.

**D.14.5. Single Glance Duration, Eyes-Off-Road – 85<sup>th</sup> Percentile**

This is related to the results on p. 152.

	Prediction [s]	Measurement [s]	Difference [s]	Relative
Task1	2.121	1.470	0.651	44.28%
Task2	1.751	2.154	-0.403	-18.70%
Task3	2.731	2.324	0.407	17.50%
Task4	1.830	1.686	0.144	8.55%
Task5	1.875	2.105	-0.230	-10.92%
Task6	2.476	2.493	-0.017	-0.68%
Task7	1.996	1.514	0.482	31.81%
Task8	1.897	1.904	-0.007	-0.35%
Task9	2.351	2.383	-0.032	-1.36%
Task10	1.993	1.803	0.190	10.53%

Table D.23.: P85 Single Glance Duration eyes-off-road – Data table

	Bootstrap Indicator (%)	Measurement Result Trial1	Measurement Result Trial2	Review (+/o/-)
Task1	12.90%	ok 23/24	ok 24/24	–
Task2	24.10%	FAIL 19/24	FAIL 20/24	o
Task3	0%	FAIL 17/24	FAIL 18/24	o
Task4	5.40%	ok 23/24	FAIL 20/24	–
Task5	0%	FAIL 19/24	FAIL 19/24	o
Task6	65.60%	FAIL 16/24	FAIL 15/24	o
Task7	11.70%	ok 23/24	ok 22/24	o
Task8	87.80%	ok 21/24	FAIL 20/24	o
Task9	0%	FAIL 18/24	FAIL 16/24	o
Task10	23%	FAIL 20/24	ok 21/24	o

Table D.24.: Bootstrapping Single Glance Duration eyes-off-road NHTSA 2 s limit – Data table

Table D.23 is based on 85<sup>th</sup> percentiles while Table D.10 (p. 152) uses the median. The prediction of the online tool also bootstraps the predicted results and calculates how often (percentage) these are lower than (mean) 2s NHTSA limit. Table D.24 reviews this indicator.

### D.14.6. DLP and DFH Bootstrap Indicator

The DLP and DFH are compared to a preliminary criteria derived from a former experiment, when tuning a radio (see Section 2.3). The bootstrapping compares how often the task would not exceed the DLP and DFH limits. Table D.25 and Table D.26 review these indicators. The criteria limit is subtracted from the measurement result. Therefore, a positive outcome (p.p.) indicates that the task would be worse than a radio-tuning criterion; a negative number would indicate a better performance. This outcome is correlated to the bootstrap indicators. For the DLP, the Pearson correlation ( $N = 10$ ) of the second and third column is  $r = -0.75$ ; for DFH  $r = 0.14$ . For this reason, the DFH is not reviewed and directly judged as unreliable.

	Bootstrap Indicator (%)	Measurement Result - Limit (117.7%)	Review (+/o/-)
Task1	0.3%	54 p.p.	+
Task2	71.0%	-14 p.p.	+
Task3	6.8%	19 p.p.	+
Task4	7.5%	-22 p.p.	-
Task5	0.1%	42 p.p.	+
Task6	0.2%	29 p.p.	+
Task7	81.8%	-37 p.p.	+
Task8	46.2%	1 p.p.	+
Task9	32.7%	29 p.p.	o
Task10	50.6%	-24 p.p.	-

Table D.25.: Bootstrapping DLP – Data table

	Bootstrap Indicator (%)	Measurement Result - Limit(52.2%)
Task1	29.30%	-7 p.p.
Task2	23.20%	-17 p.p.
Task3	42.40%	-15 p.p.
Task4	42.50%	-30 p.p.
Task5	52.60%	-16 p.p.
Task6	74.10%	10 p.p.
Task7	25.90%	-12 p.p.
Task8	87.30%	-16 p.p.
Task9	1.70%	-8 p.p.
Task10	57.10%	-14 p.p.

Table D.26.: Bootstrapping DFH – Data table