Technische Universität München
Computational Science and Engineering
(Int'l Master's Program)

# Integration of a Component Based Driving Simulator and Design of Experiments on Multimodal Driver Assistance

Master's Thesis

Darya Popiv

Supervisors:

| | |
|---|---|
| 1st Examiner: | Prof. Gudrun Klinker, Ph.D. |
| 2nd Examiner: | Univ.-Prof. Dr. rer. nat. Heiner Bubb |
| Handed in: | 21.12.2006 |

*I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.*

21 December, 2006

# Abstract

Fully automated driving is a future goal of research currently performed in automotive industry. Therefore this thesis deals with driving support systems on different levels of automation. Fully integrated multimodal approaches for such driving systems aim at providing intuitive means for minimally distractive assistance for car drivers. In this thesis, the description of design and implementation of three concepts of driving support systems in the driving simulator is given.

The driving support systems are based on three concepts. Every concept represents assistance on different level of driving automation. A non-automated concept is based on the principle of driving activity without an automation support provided to the driver. A semi-automation support system is represented by the concept of Active Cruise Control, in which driver performs a role of a system’s supervisor and delegates part of the driving tasks to the system. The third concept is the concept of Active Gas Pedal. In terms of this concept driver is offered support on behalf of the driving system, but still is required to perform tasks of driving personally. Also lateral and longitudinal visual assistance is incorporated into the implementation of the three described concepts. To test mentioned concepts, a fixed-base driving simulator was set up, the architecture of which is explained in this thesis as well.

The set-up of the fixed-base driving simulator, its hardware components, corresponding interfacing software applications, and their networking are described. The software system architecture in the driving simulator is explained, and development process of the driving support systems is introduced. Also, needed implementation information is provided for further extensions of the software system.

Finally the experimental design for the user study is described, so that the experiment only needs to get executed.

## Zusammenfassung

Das vollautomatisierte Fahren ist ein zukünftiges Ziel der momentanen Forschungsaktivitäten in der Automobilindustrie. Aus diesem Grund beschäftigt sich die vorliegende Masterarbeit mit Fahrerassistenzsystemen auf unterschiedlichen Automatisierungsebenen. Integrierte multimodale Nutzerschnittstellen sollen den Fahrer optimal unterstützen ohne ihn von der Fahraufgabe abzulenken. Im Rahmen dieser Arbeit werden drei Konzepte für Fahrerassistenzsysteme in einem statischen Fahrsimulator implementiert.

Die entwickelten Systeme greifen auf drei unterschiedlichen Automatisierungsebenen an. Das nicht automatisierte Fahren stellt die Referenzbedingung dar. Das so genannte ACC (Active Cruise Control) stellt ein halbautomatisiertes Fahren dar, bei dem der Fahrer die Längsführung an das Fahrzeug delegiert und sich in die Rolle des Systemüberwachers begibt. Das dritte Konzept ist das aktive Gaspedal. Bei diesem Konzept wird der Fahrer von seinem Fahrzeug bei der Fahraufgabe unterstützt, muss jedoch die nötigen Aktionen selbst durchführen. Ferner werden die Assistenzsysteme von einer optischen Ausgabe in einem Head-up-Display unterstützt.

Der Aufbau des statischen Fahrsimulators, die Hardware-Komponenten, korrespondierende Software Applikationen und die Netzwerkstruktur werden beschrieben. Die Software Systemarchitektur des Fahrsimulators sowie der Entwicklungsprozess der Fahrerassistenzsysteme wird vorgestellt. Ebenso wird ein Ausblick auf weitere nötige Softwareentwicklungen gegeben.

Abschließend wird ein Versuchsdesign zur experimentellen Absicherung der Assistenzsysteme vorgestellt.

# Table of Contents

# List Of Figures

# List of Tables

# 1 Introduction

Driving assistance is gradually approaching the fully automatic level. Assistance systems, in which the driver becomes the supervisor of the system performing driving tasks, are already being embedded into cars. However, present automation systems are not yet good enough to replace the human completely (Hoc and Young, 2006). Therefore main research of this thesis work is focused on driver support improvement rather than driver's full replacement.

In terms of this work, different driving support systems were investigated and implemented. Presented driving support systems operate on different automation levels, and are based on the three concepts.

In terms of the concept "Aus"[1] driving process is performed without any lateral and longitudinal assistance. This concept can be viewed as the baseline to which other concepts operating on higher levels of automation should be compared.

Semi-automated driving is represented by the concept "Active Cruise Control" (ACC). In terms of this concept part of driving tasks is delegated to the driving support system. It is responsible for accelerating and braking (till some allowed deceleration), therefore enabling longitudinal assistance. However, the driver is still responsible for supervision of such a system, and should be able to regain control over own vehicle in critical situations.

The concept "Active Gas Pedal" (AGP) provides decision support by the haptic means to the driver regarding the task of driving. In this concept, the longitudinal assistance is enabled by the resistance point set on the gas pedal. This point defines the advised gas pedal position, which produces the amount of acceleration needed to keep the desired speed and also desired following distance to a leading vehicle. However, it is up to drivers to follow the advice of the system, or to enforce their own action by overpowering the resistance point.

---

[1] German word "Aus" translates as "Off"

To evaluate developed concepts, the driving simulator at Lehrstuhl für Ergonomie (LfE), Technische Universität München (TUM) was set up. The driving simulator consists of hardware components delivered by different firms. After they were assembled, the software had to be introduced to enable the process of driving. First, the analysis of 3rd party interfacing software was performed and the driving simulator's software system architecture done. Afterwards the component-based software framework application *Fahrsim* was developed. It enables the required support for each interfacing software application that serves corresponding hardware component. As result, a person is able to drive in the simulator similar to the driving in normal car.

To perform the research on the three concepts, the implementation of the three driving support systems was embedded into *Fahrsim* application. These will be researched in the upcoming experiment. The main goal of the experiment is to compare driving support systems offered to human on different levels of assisted driving automation.

# 2 Related Work

In this section, driving support systems and their representations that have relevant input on the design of the three concepts presented in this master thesis are described. Research on related work proved necessity and validity of the driving support systems afterwards implemented at the driving simulator. Furthermore it made valuable contribution to the design decisions on haptical and optical realization of driving assistance in terms of the three concepts.

## *2.1 HUD Technology*

Driving support that aims to assist drivers in their awareness about the car's state is of great importance to automotive industry. Different information related to car's state is necessary for drivers to perform their travel, including current speed of the car, rounds per minutes, fuel consumption, etc. Means to inform driver about state of the car are constantly being improved, and in place of already well-known instrument binnacles and in-car displays new devices are introduced, namely Head-Up Displays (HUDs).

Originally coming from military aircraft industry, HUDs are gradually being adopted by automotive industry. A HUD is used to display information regarding to the state of the car by means of projecting relevant symbols at fixed location onto the windshield above the hood. A HUD's objective is to display information related to the task of driving, e.g. the current speed and speed limit, route guidance information, warnings. Corresponding symbols are projected onto the windshield appearing to the driver as transparent icons placed upon the road background. As a result, a driver is able to perceive information without taking eyes off the road. Usage of HUDs drastically increases the "eyes-on-road" time (Gish and Staplin, 1995) while providing helpful information to driver.

As recent investigation of BMW shows, total duration of glances on the speedometer is twice as long compared to the time spent looking up the current speed at the HUD (Joachim Kaufmann, 2004). Overall, the performed experiments proved that displaying information at the HUD distracts drivers much less from their primary task of driving compared to displayed information on in-car displays or instrument binnacles.

Negative effects of HUDs which were implemented without appropriate consideration of symbols' size and color also have been examined. The size of available HUDs is small (for BMW 6th series it is 18 by 9 cm), and therefore the arrangement and amount of displayed information has to be carefully considered. Colors which are used for HUD symbols should be distinguishable from the rest of the world's background; otherwise a driver is likely to misread HUD's information. BMW found orange as suitable color for the text and symbol appearance. Orange has a high luminance and saturation and thus has a high contrast with grey road textures, which is the most appearing background for displayed symbols due to the conventional location of a HUD (HUD is located directly above the hood).

## *2.2 Longitudinal Assistance*

Significant amount of accidents occurs due to the rear-end collisions (Statistisches Bundesamt Deutschland, 2005), and therefore longitudinal assistance is of great importance in automotive research. Main task of longitudinal assistance is to help the driver to keep a wanted speed and following distance to a leading vehicle to avoid rear-end collisions.

On different levels of automation, longitudinal driving support systems are being embedded into cars. Assistances of driving support systems that are relevant to the design of the three concepts are presented in following sections.

### 2.2.1 Active Cruise Control

An ACC system controls acceleration and deceleration of the own vehicle automatically. It maintains driving speed of the car which is manually chosen by the driver (referred to as *wanted speed*), and also keeps the car on a preset distance to a leading vehicle (*desired following distance*) if such vehicle is present. A driver is allowed to press the cancel button or to apply the brake pedal to cause disengagement of the ACC system. The drivers as well can override the acceleration produced by ACC system by applying the accelerator pedal on their own.

Research performed by Brookhuis and de Waard (Brookhuis and Waard, 2006) showed positive results on ACC system acceptance by the drivers. During the training phase subjects failed to reclaim control over the system in 28% of critical situations, which included unexpected large deceleration and illegal changing of the lane by leading vehicles. Nevertheless, once drivers were used to work with the system, they were all able to react in time to reclaim control over their car and therefore avoid collisions when unexpected driving behavior of foreign vehicles occurred.

### 2.2.2 Active Gas Pedal

Another example of longitudinal assistance is an AGP. The principle of an AGP is to create a resistance point on the gas pedal. If the driver keeps their foot on the resistance point, the car will drive at wanted speed and also will remain at desired following distance to a vehicle in front. However, the driver is free to overcome the resistance point by applying slightly more force on the gas pedal (Figure 5, pg.16), as well as to give less gas by not pushing the gas pedal through to the resistance point.

As recent research shows (Lange et al., 2006), introduction of an AGP as driving assistance results in better keeping of wanted speed compared to unassisted drives. Maximum deviation from allowed speed with an AGP is much smaller. Driver’s glance time on the speedometer is also considerably reduced. Subjectively, many drivers enjoyed the use of an AGP and would be glad to use it in their cars.

### 2.2.3 Braking Distance Bar

A braking distance bar which is shown to the driver in conformal HUD provides longitudinal assistance. The purpose of conformal HUD is to project AR schemes on top of the forward field of view. From the driver’s perspective, these AR objects appear to be located on the driving scene. The braking distance bar is represented by a transparent image of a bar which is overlaid upon the road in front of the car, and its movement depends on speed of the car (it appears moving further away with increasing speed) and front wheel angle (it moves to the direction at which the front wheels are turned).

First attempt to represent a braking bar in real cars was done by Bubb (Bubb, 1976) and Assmann (Assmann, 1985). They constructed a special double-ended tubular lamp which was mirrored into the windshield in such a way that the faster car drove the further and smaller the lamp appeared to be in the windshield, thus generating perception of increased distance. Positive results led to further research, and emergence of conformal HUDs gave the opportunity to improve the presentation of a braking bar.

In experiments performed by Tönnis (Tönnis et al., 2006), two presentation schemes for a braking bar were investigated. In one of them a virtual bar is placed upon the road representing the braking distance therefore enabling longitudinal anticipation of the car’s speed. Tunneling lines connecting the car’s front with the sides of the braking bar are used for the second presentation scheme. As results show, drivers enjoyed more and performed better with the first representation (without tunneling lines), which is the reason why in the design of the concepts introduced in this work this scheme was used. Overall drivers drove faster with this longitudinal assistance than without, and safety distances such as the distance to a vehicle in front were not decreased.

Another longitudinal support that can be provided by a virtual bar is anticipation of the desired following distance. Depending on the current speed and changes of the desired following distance, the bar moves closer (lower speed or shorter desired following distance) or further away from the driver (faster speed or longer desired following distance) (Thompson, 2005).

## 2.3 Lateral Assistance

A braking bar and a following distance bar also enable lateral assistance to the driver. The motion of a bar in horizontal plane depends on the car’s front wheel angle. As the investigation made by Tönnis (Tönnis et al., 2006) showed, drivers were able to keep their lane better with this concept of assistance in comparison to non-assisted driving.

# 3 Concept of Assisted Driving

The goal of driving assistance is to improve a driver's capability to perform the main task of driving. Major activities performed by a driver during their journey are navigation, stabilization and maneuvering (Bernotat, 1970). *Navigation* is a process of following a predefined route from starting position to a chosen destination point, *stabilization* is performed while driving the route at wanted speed and keeping safety distances, and *maneuvering* includes such actions of a driver as an appropriate passing of other cars, or driving through intersections. To execute these activities, drivers have to continuously execute a *control circuit* task. They have to perceive input by all senses, process it in their brain and then transcribe the next steps of their driving plan into manipulations of the steering wheel, the gas and the brake pedals (Bubb, 1993). This is a continuous process, and can be seen as an activity which repeats itself in a loop. Therefore, the term *"driver in the loop"* can be used to emphasize that even certain automated assistance might be present, the driver still has to perform all of the tasks of control circuit: perceive, analyze, and act. On contrary, *"driver out of the loop"* is used when drivers are no longer participants in the activity loop, rather their role shifts to supervision of the system and regaining control when needed. The point made by Bainbridge (Bainbridge, 1983) is still valid for modern driving support systems: normal operation can be performed automatically by the system, while abnormal conditions are to be dealt with manually. Important issues to be considered when introducing automated systems that put "driver out of the loop" are whether drivers trust automated vehicles, whether they actually reclaim control if required, and whether they accept supervising an automated vehicle instead of driving (Brookhuis and de Waard, 2006).

## *3.1 Levels of Automation*

The typology of automation used for designing the three concepts of driving support systems represented at this work was introduced by Endsley and Kiris (Endsley and Kiris, 1995). The typology is defined by four levels:

1. perceptive cooperation mode
2. mutual control cooperation mode
    2.1. warning stage
    2.2. action suggestion stage
    2.3. limit stage
    2.4. correction stage
3. functional delegation cooperation mode
4. fully automatic cooperation mode

### 3.1.1 Perceptive Cooperation Mode

In the *perceptive cooperation mode*, the driver is left to interpret the driving support provided by the system. Driver is only informed about current or approaching situation. However, the interpretation of the signals as well as which action to take is completely left to the driver.

### 3.1.2 Mutual Control Cooperation Mode

In the *mutual control cooperation mode* driving support system presents not only information, but also guides the driver through certain stages which help to accelerate the process of performing the action.

At the *warning stage*, the driver is informed about the upcoming situation. Afterwards at the *action suggestion stage* the system suggests what should be done in this particular situation. Then at the *limit stage* the system behavior changes so that the driver is guided by the system towards the desired action which is calculated as the proper one. However, it is still up to the driver to either follow the system’s recommendations, or to overrule it and enforce his own action and therefore enter the *correction stage*.

### 3.1.3 Functional Delegation Cooperation Mode

In the *functional delegation cooperation mode* driver delegates parts of driving tasks to the system. Delegating of driving responsibilities significantly reduces human’s workload, and operator does not participate directly in the control loop anymore. However, if the automation is no longer needed, the system gives manual control back to the driver. At this mode automation systems are said to be adaptive. *Adaptive interfaces* are intelligent systems that monitor the environment for changes in task demand, and regulate the level of automation to maintain an optimal state of system control for the operator at each particular situation (Hoc and Young, 2006).

### 3.1.4 Fully Automatic Cooperation Mode

In the last level, the *fully automatic cooperation mode*, the whole control over the system is taken by automated systems. This mode might be needed in critical situations in which the system recognizes inability of the driver to physically react on time. It might be useful in situations when sudden return of the control back to the driver might cause the car to crash in case the driver fails to overtake the charge immediately.

## *3.2 Three Levels of Driving Assistance*

Full-integrated multimodal driving assistance system which is developed in terms of this master thesis implements first three modes (see pg.9-10) of automation. The system integrates three concepts: concept “Aus”, concept “Active Cruise Control”, and concept “Active Gas Pedal”.

*Concept “Aus”* is realized in the perceptive cooperation mode. This concept represents the most common driving support available to a driver today. *Concept*

*“Active Cruise Control” (“ACC”)* represents driving assistance on the functional delegation cooperation mode, which is gradually embedded into cars, but still means to improve it are widely researched. Also, assistance on the mutual control cooperation mode becomes more and more popular in car production, therefore *concept “Active Gas Pedal” (“AGP”)* is implemented as driving support.

Driver settings of wanted speed and desired following distance are used in implementation of driving support realized in terms of the “ACC” and “AGP” concepts. As it was mentioned before, wanted speed is the speed at which the driver is willing to drive. Usually the driver chooses the value of the wanted speed which correlates with the allowed speed at the driven road segment.

The desired following distance to a leading vehicle is the time that relates to the safe distance to the car in front. In design of the three concepts, the driver is able to choose the time to the next car between 0.9 and 1.5 seconds, which are the standard values used by BMW.

Both wanted speed and desired following distance settings (also referred to as *ACC settings*) can be changed with the help of control devices located on the steering wheel. This choice of control location is with correspondence to the UMTRI Guideline 2: “Controls used most frequently or for critical functions should be close to the predominate position of the hands” (Green et al., 1993).

Visual realization of current driver’s ACC settings is represented in regular and conformal HUDs. In a HUD, the corresponding signs for wanted speed and following distance are shown. If the driver is willing to use 3D optical representation for the desired following distance setting, a bar is shown in conformal HUD (see 2.2.3).

### 3.2.1 Concept “Aus”

The driving process which is performed in terms of concept “Aus” can be depicted as the driving with *fully manual control*. It means the driver always stays in the control loop, and no lateral or longitudinal assistance is presented. The only relevant information displayed in HUD is the current speed of the car (Figure 1, pg.13). This concept is the baseline for the upcoming experiment, to which the efficiency of the “ACC” and “AGP” concepts is to be compared.

### 3.2.2 Concept “Active Cruise Control”

In this concept, the ACC system regulates speed and distance to a leading vehicle in place of the driver. ACC overtakes such tasks of the driver as accelerating and braking (till certain allowed deceleration). The concept of automated braking is known as *Active Brake Pedal*. The ACC system also warns a driver when the situation becomes critical, for example when the driver should overtake control to avoid collision of an emerging or braking vehicle in front.

Driving support realized in terms of the “ACC” concept belongs to the functional delegation cooperation mode, and therefore driver is no longer in the loop when the ACC system is activated.

In addition to current speed, visual information regarding driver’s ACC settings is presented. Two representation models are used to display such supportive information. The first model is a *2D model*, in which only 2D optical information is presented: both wanted speed and desired following distance are displayed in the HUD as symbolic icons (Figure 2, pg.13).

*Figure 1: Concept "Aus" – only current speed is shown in HUD*

*Figure 2: 2D optical information – current speed, wanted speed and desired following distance shown in HUD*

The symbol representing the desired following distance changes if a vehicle in front of own car is detected (Figure 3).

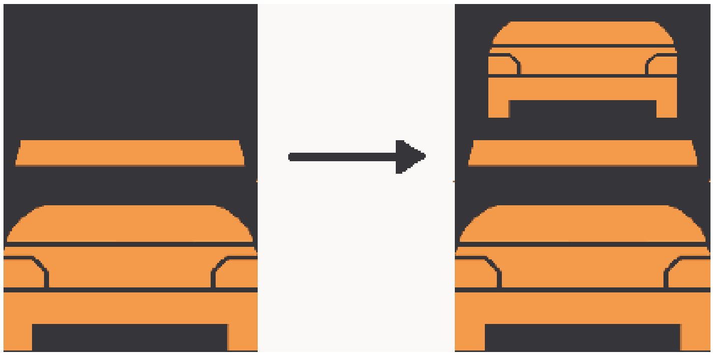


*Figure 3: Warning symbol replacing desired following distance symbol when a vehicle in front of own car is detected*

In the *3D model*, the driver is exposed to 3D optical information. Desired following distance is presented with help of a following distance bar which is shown by means of a conformal HUD (Figure 4).

*Figure 4: 3D optical information – desired following distance is represented by the bar in a conformal HUD*

The driver gets informed if a vehicle in front is detected: the bar changes its color from green to orange.

### 3.2.3 Concept "Active Gas Pedal"

Purpose of an AGP is to assist a driver in keeping the wanted speed and wanted distance to the next car by defining a point of resistance on the gas pedal. Following the point of resistance the driver is capable to stay in the limits of wanted speed, which can be either dictated by the ACC settings of wanted speed, or, if a vehicle in front is detected, by the desired following distance. However, driver is able to push over the point of resistance to accelerate more than suggested by the system.

The "AGP" concept functions on mutual control cooperation mode of automation. The driver stays "in the loop", while the system analyzes the situation and haptically presents the right gas pedal position.

As it was mentioned before (see pg.9-10) the mutual control mode can be divided into four stages. During the warning stage, the driver is shown by visual assistances warning signs, such as when a leading car is being detected in the zone of desired following distance (Figure 3, pg.14). As suggestion, the resistance point is set on the gas pedal, following which driver reaches or keeps once reached the wanted speed. However, driver is free to overcome the resistance point in order to give more gas.

There are two models for visual assistances used in this concept. They were described earlier in 3.2.2. First model is when all relevant information such as desired following distance and wanted speed is displayed in regular HUD, and second, when only wanted speed is displayed in the HUD, while following distance to the car in front is shown with help of a bar in the conformal HUD.

An AGP force appliance algorithm represented on Figure 5, pg.16 is used for designing the concept.

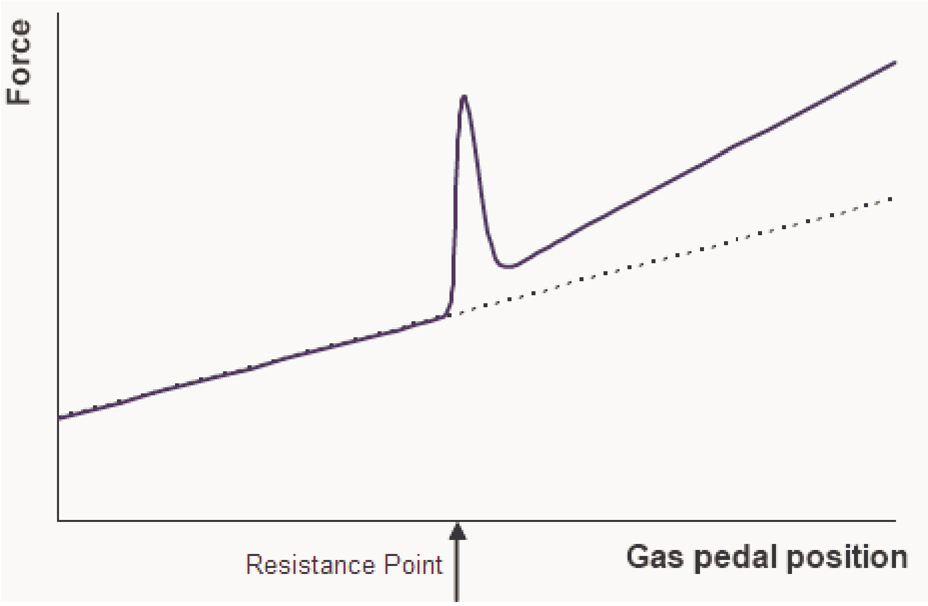


*Figure 5: The force on the gas pedal for the active gas pedal (solid line) compared to a conventional gas pedal (dotted line) (Thompson, 2005)*

Up to the resistance point, the gas pedal operates exactly like a conventional gas pedal. To pass through the resistance point, the required force is slightly greater than normal. It signals the driver that they are driving above the wanted speed or too close to a vehicle ahead (Thompson, 2005).

# 4 Set-up of Fixed-base Driving Simulator

To implement the three concepts of driving assistance, a driving simulator was set up. It was a fixed-base simulator, in which the car does not move, and therefore no physical feed-back of car’s acceleration, pitch, or roll, is provided to the driving person.

The driving simulator consists of different hardware which is operated by interfacing software. Hardware includes the components which are simulating the real car and its parts, i.e. the gas and brake pedals, steering wheel, etc. Software connects and manages hardware components, so that the resulting behavior of the car and the simulated world around gives a person the feeling of real driving.

In the next sections, first the existing hardware is introduced and then integrated 3rd party software is listed. It is followed by description of hardware components’ interfacing software and its networking. The last section illustrates the software application that was designed and implemented to enable driving process at the simulator.

## 4.1 Hardware

Following hardware components were assembled during the set-up phase of the fixed-base driving simulator:

- Car
    - gas pedal,
    - brake pedal,
    - steering wheel with ACC controls
        - control allowing to choose the level of automation support (three buttons: “Aus”, “ACC”, “AGP”),
        - control allowing to set wanted speed (two buttons: “Up” – increases wanted speed on 5 km/h, “Down” – decreases wanted speed on 5 km/h),

        - control allowing to set desired following distance (one button: pressing it results in change of desired following distance between two values: 0.9 sec and 1.5 sec),
    - instrument binnacle
        - speedometer,
        - tachometer,
        - indicator of turn signals,
    - HUD,
- three projection walls, and three corresponding projectors connected to computers generating the world scenery,
- conformal HUD projector (simulation of conformal HUD).

Figure 6 shows the driving simulator and some of its hardware components.

*Figure 6: Driving simulator at LfE, TUM*

The car at the driving simulator is a BMW 6th series convertible. It includes important hardware components which are controlled through interfacing software. There are gas, brake pedals, steering wheel, instrument binnacle, and HUD.

The gas pedal is capable to perform the functionality of an AGP. It was designed at LfE, TUM (Lange et al., 2006).

The steering wheel's hardware and corresponding software interface were done by the third party firm[2]. It has force feed-back functionality to imitate the behavior of a real steering wheel. In addition, ACC control buttons are located on it. These are three buttons for choosing the assistance type ("Aus", "ACC", "AGP"), two buttons to increase or decrease wanted speed, and one button to switch desired following distance (default is 1.5 seconds). To provide the driver with the feedback on which assistance type is chosen, or which button is pressed, light emitting diode (LED) is built in near each button. LED of chosen assistance type is always on, while other corresponding LEDs are blinking when wanted speed or desired following distance is being changed.

The instrument binnacle and its interfacing software were done by third firm[3]. It has speedometer and tachometer. If the assistance "AGP" or "ACC" is on, wanted speed is represented by a narrow arc of light outside the speedometer opposite to the corresponding speed.

In order to simulate the world around the car, three projection walls with three projectors displaying the world scenery are used. Three projection walls surrounding the car provide the driver with approximately 180° field of view. For more on the geometrical set-up of the driving simulator see Appendix A.

---

[2] Simotion, Munich, Germany

[3] Usaneers GmbH, Munich, Germany

A conformal HUD is simulated with the help of an appropriately calibrated video projector. It projects HUD schemes such as a following distance bar onto the central projection wall upon the projection of the world scene.

## *4.2 World and Car Simulation Software in Driving Simulator*

The world and car simulation software is based on *Model/View/Controller* paradigm (Brügge and Dutoit, 2004). The software components responsible for the world simulation in which the car drives are:

- World scenery (*View*);
- Driving Dynamics (*Model*);
- Foreign Vehicles Simulation (*Model*);
- Controlling unit (*Controller*) responsible for binding World, Driving Dynamics, and Foreign Vehicles components

The software which implements above listed components was developed by the third party firm[4].

### 4.2.1 World Scenery

World scenery consists of highway and the landscape surrounding it. It is a rural view with villages, forests, and rivers. The world scenery was designed so that the driver at the simulator gets the realistic impression of driving through highway while performing experiments.

Three viewers are responsible for generating overall picture of world scenery. Each of the viewers creates image with appropriate projection parameters to be shown on right, center, and left projection walls.

---

[4] S&P Simulationstechnik GmbH, Innsbruck, Austria

### 4.2.2 Driving Dynamics

The driving dynamics is responsible for processing the input of the gas pedal, brake pressure, and steering wheel angle to calculate the car related parameters. These parameters are calculated on the base of two-lane driving dynamics model and world's relevant information such as elevation:

- speed;
- velocities in x, y, z directions;
- accelerations in x, y, z directions;
- heading, pitch, roll;
- rpm.

### 4.2.3 Foreign Vehicles Simulation

Foreign vehicle module is responsible for simulating the traffic in the world. It defines the number of vehicles, and their behavior on the world's roads. The behavior and motion of foreign vehicles is defined by following information:

- technical information about foreign vehicle, including size, model, color;
- its driving route, which is defined by the position at which a vehicle starts driving, and which roads it takes at each particular intersection on driving way;
- parameters defining the behavior with respect to other cars, including following distance and overtake probability;
- parameters related to the behavior at the intersections, such as braking distance before the turn;
- the average speed at which foreign car is traveling. There are three modes defining how fast car goes with respect to the speed limit on the given road: slow, normal, fast.

The information calculated at each step for every car in the foreign vehicle module is

- speed and velocities in x, y, z directions;
- accelerations in x, y , z directions;
- heading, pitch, roll;
- angular velocities and accelerations in x, y, z directions;
- position of the car in the world;
- its direction;
- its road number;
- state of the lamps.

The foreign vehicle module also processes the information about own car's position and speed to avoid the collisions from foreign vehicle's side.

### 4.2.4 Controlling unit

Controlling unit is responsible for defining state of the car in simulated world. It asks the information at certain frequency about own car's and foreign vehicles' state, merges it with the simulated world's information, and sends the viewers the data needed for displaying correct world scenery.

The overall set-up of the fixed-base driving simulator at LfE is represented at Figure 7, pg.23.

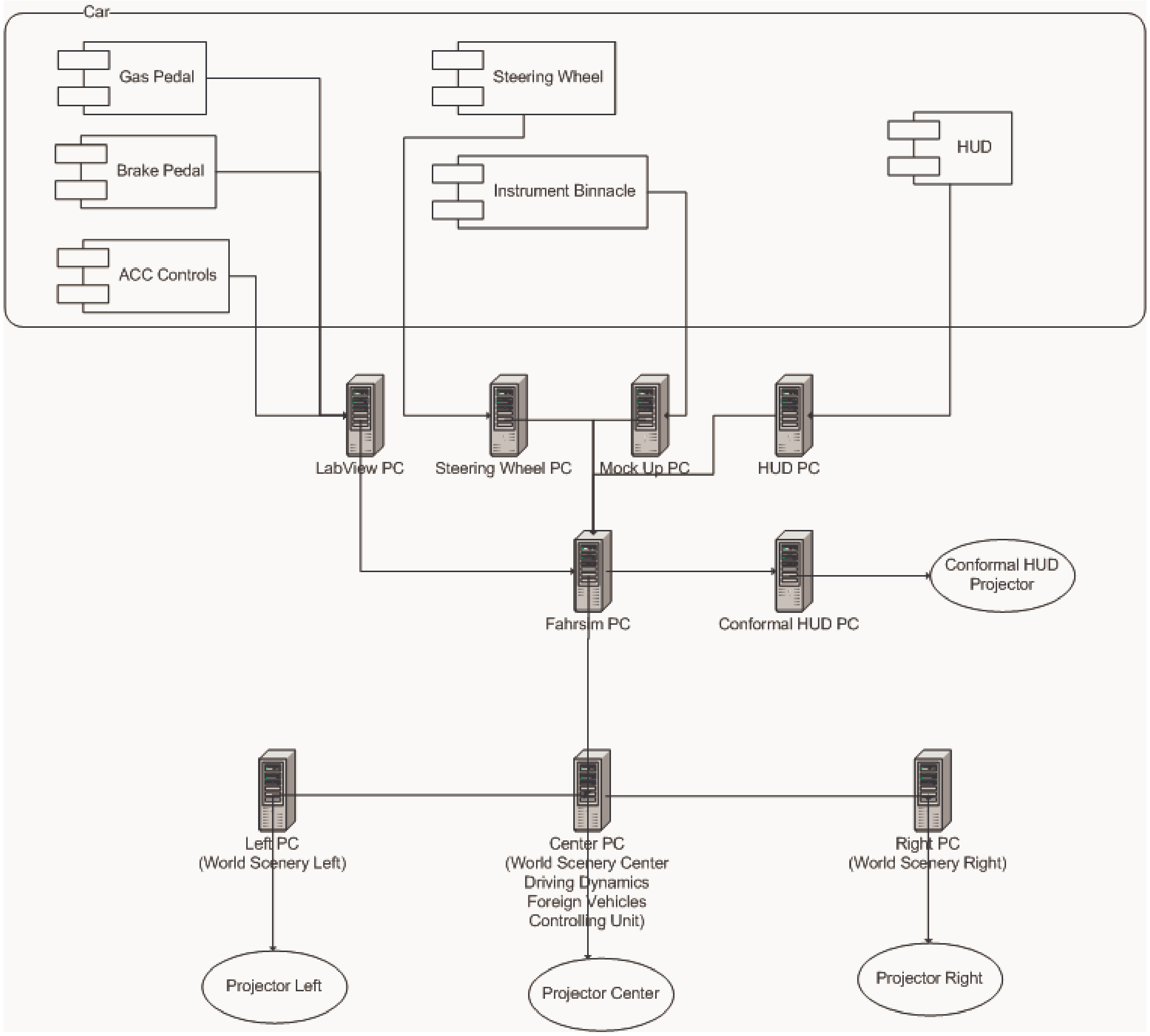


*Figure 7: Hardware integrated at the driving simulator*

## 4.3 Interfacing between Hardware and Software

The following section provides the description of driving simulator's interfacing software applications.

Each of the hardware components has its own interfacing software application. These software applications require input of some kind which depends on the functionality of its hardware component, and they also send information to other interfacing software programs.

Input required by interfacing software can be either direct user input as in the case of gas and brake pedals, or it can be also the data packet with specific information expected from other components. For example, the software application simulating the world updates scenery based on the received data concerning car's actual gas position, brake pressure, and steering wheel angle.

There are also devices that need both kinds of input. The steering wheel requires not only physical turning by a driver, but also the current speed of the car which is calculated by driving dynamic application and sent to it through the network. Current speed of the car is necessary for proper calculation of steering wheel's force feedback presented to the driver.

Two types of protocols are used for transferring data among applications present in the driving simulator: UDP and TCP.

### 4.3.1 Continues Data Streams

UDP is used for transportation of continues data streams, where occasional lost of a packet does not result in system failures. For example, data from the steering wheel serving application are passed with the frequency of 100 Hz. If one of the packages is lost, it won't affect the overall system behavior. Therefore UDP protocol between sending and receiving applications is used, which is faster than TCP and generates less network load.

As already mentioned, the steering wheel and its serving application were developed by a third party firm. The serving software application sends the turning angle and receives speed of the car to simulate the resistance of the steering wheel as if the friction between the tires and the road was present. Sending and receiving of the information is done through UDP protocol.

### 4.3.2 Discrete Events

On contrary, some events occur only once in a while, and loss of a data packet which corresponds to these events results in failures or inconsistent behavior of the system. To ensure delivery of a packet, the reliable TCP protocol is used instead of unreliable UDP. When the user presses the button, e.g. to increase wanted speed or to change desired following distance, corresponding event is issued only once. If the packet would get lost, the system would not react on driver's performed action. The driver would have to press the button once more, until the package responsible for notification of performed event was delivered. However, small failures decrease the driver's acceptance and reliance on the system.

Gas pedal, brake pedal, as well as ACC controls including buttons used to set the degree of driving support automation, wanted speed, and desired following distance, are served by National Instrument (NI) LabVIEW®[5] application. LabVIEW® is a development tool which enables the development of scalable test, measurement, and control applications. The LabVIEW® application implemented at LfE reads the values of the pedals' states from the measurement card, and, through the radio signal, captures the information about pressed buttons. Because information about the state of pedals and pressed buttons is incorporated into one packet, LabView uses TCP protocol to send the data further to the application responsible for distribution of the data some other driving simulator components. One may argue that it was possible to implement TCP for sending events that correspond to pressing buttons and UDP for sending gas and brake. However, TCP was chosen because introducing two different protocols in LabVIEW® application would have complicated the program architecture, and cause difficulties in its maintenance.

---

[5] http://www.ni.com/labview/d/ accessed 17.12.06

The LabVIEW® application receives the voltage values to be set on the gas pedal's. The voltage values correspond to the resistance point on the active gas pedal's servo motor.

The instrument binnacle's application is responsible for functionality of speedometer and tachometer. Data flow is organized with usage the UDP protocol. The received data from driving dynamics application are current speed and rpm.

Data flow between driving dynamics, foreign vehicle simulation, controlling unit, and viewers which are responsible for generating world scenery is performed through UDP sockets.

Because the interfacing software is developed by different firms, some of sent protocol structures differ from expected by the other software. Therefore, additional program was designed and implemented, which serves as communicational link between interfacing software.

## *4.4 Fahrsim: Central Program of Driving Simulator*

After the driving simulator had been set up and its components' interfacing software applications studied, the program *Fahrsim*[6] was developed. Its purpose is to tie the functionality of different devices which could not be connected directly through their interfacing software. The execution of this program together with interfacing software applications enables a person to drive in the simulator.

### 4.4.1 Set-up of the Fahrsim Application

First, a list of requirements which had to be fulfilled by *Fahrsim* program was made:

---

[6] The name for the central program *Fahrsim* comes from abbreviation of the German word Fahrsimulator (driving simulator).

- *modular architecture*: different functionality should be realized in different modules, i.e. communication should be separated from the logic of driving assistance. This enables quick understanding of the program's architecture by following program developers, and furthermore eases the implementation and deployment of new functionality;
- *solid basis for implementation of driving assistance*: except organizing the data flow and therefore enabling the process of driving in the simulator, the program should provide the means to embed functionality of the three concepts. The *Fahrsim* program's interfaces should be well-thought to allow easy integration of driving support assistances;
- *exchangeability of algorithms which implement the assistance*: there can be multiple algorithms behind some of the assistances, i.e. the ones realizing “AGP” and “ACC” concepts. Therefore design of central program should allow easy replacement of the underlying algorithms, so that different algorithms can be tested and the most appropriate chosen.

To have an extensible application which organizes the data flow in the driving simulator, a choice for component-based architecture was made. Component-based software framework has advantages for all parties involved into the development process. It allows reusability of existing components, reconfiguration of the system by plugging modules together, and viewing the system at a high level of abstraction (Bauer et al., 2001). In the component-based architectures, component is a class, or a group of classes, which is responsible for providing predefined services to other components, and receiving needed services in return.

As the first task, the suitable framework for *Fahrsim* application was found. A framework is a software package which eases implementation of certain aspects of an application development. The use of frameworks results in decreased development and deployment time, and accelerates the process of designing program by proposing implementation rules. The rules of framework are defined by a set of design patterns, a set of interfaces and their implementations, and related tools used in development. The driving simulator's framework used as the basis for central program had to be

- *light weighted*: framework should include necessary classes and interfaces applicable for the driving simulator software, but not overwhelming amount of additional classes which provide the functionality that never is needed;
- *easily understandable*: framework should be good documented and number of the most important APIs reasonably small;
- *quickly extensible*: framework should provide useful interfaces and their general purpose (primary) implementations, or abstract implementations, so developers don't have to write them from scratch;
- *flexibly configurable*: simulator software should provide flexible configuration which changes with the driving assistance needed for particular run, e.g. no assistance is required for a person to be able to drive in the simulator, while for the experiment drive such functionality as AGP, ACC should be activated.

Available component-based frameworks Spring, PicoContainer, and Fortress Excalibur were compared in order to find the most suitable.

*Spring*[7] is one of the widely used frameworks. However, for the needs of the driving simulator only part of it would have been used. Such features as integration with *Hibernate*[8] object-relational persistence and query service, *Model View Controller (MVC)*[9] web frameworks are not needed for simulator's central software. In addition, *Spring* imposes requirements on its components which in our case are irrelevant, e.g. its components should be implemented as *JavaBeans*[10].

---

[7] http://www.springframework.org/ accessed 17.12.06
[8] http://www.hibernate.org/ accessed 17.12.06
[9] http://www.torsten-horn.de/techdocs/jee-spring-mvc.htm accessed 17.12.06
[10] http://java.sun.com/products/javabeans/ accessed 17.12.06

*PicoContainer*[11] uses constructor injection as framework's underlying pattern. It means that an application object representing component gets all dependencies via the constructor. Configuration of the simulator's software should be easily changed depending on different assistance to be activated, and constructor injection imposes flexibility restrictions.

*Fortress*[12] is component-oriented framework developed by Apache software organization. Main concepts of Fortress are:

- Inversion of Control Pattern;
- Separation of Concerns Pattern;
- Component Lifecycle.

The detailed description of presented Fortress concepts is given in the following sections.

### 4.4.2 Inversion of Control Pattern

*Inversion of Control Pattern* (IoC) has been stated several years ago. Since then, it is widely used when tight coupling of the components endangers the system functionality. It is often the case in component-oriented software architectures, because the implementation of each component should be more or less independent from the implementation of other component. Moreover, as the practice shows, one instance of the component can be requested by multiple components. As the result, changes in the implementation of specific component should not affect other components that cooperate with it.

So what is needed? Class A must be able to get the reference to the instance of the class B without being concerned with its initialization, and not caring if B is an interface, abstract or concrete class. In IoC frameworks, the initialization and inner implementation of the class B are no longer the concerns of class A. Also

---

[11] http://www.picocontainer.org/ accessed 17.12.06
[12] http://excalibur.apache.org/ accessed 17.12.06

other classes can have references to the same object of the class B. Once the control of the object of the class B does not longer belong to the object of the class A, even though it has the needed reference to it, we are talking of the Inversion of Control Pattern.

Fortress is based on IoC pattern. It sets the references between the components by the principle of loose coupling, which allows the developer to efficiently integrate and test new components, not caring too much of how to implement and inject the needed references to the cooperating components.

### 4.4.3 Separation of Concerns Pattern

*Separation of concerns* is at the core of software engineering in general, and of object-oriented software development in particular (IBM research, 2000).

The principle of separation of concerns pattern is to be able to distinguish the main concerns with which the components should deal. Fortress framework realizes the concerns in the form of interfaces. The framework provides number of basic interfaces, realization of which allows the Fortress correctly initialize and link the components with the respect to inversion of control concept.

Interfaces provided by the Fortress can be extended by the new interfaces. These interfaces arise from the purpose of the specific program. They can represent either the roles which components perform, or the features common throughout the classes, or the aspects which are cross-cutting the functionality of the specific program.

### 4.4.4 Component Lifecycle

To establish the communication between the relevant components, and to manage the behavior of components, containers are used. Containers include groups of components, interaction of which provides specific functionality. In order to create a communicational network, the container first has to initialize required components. Afterwards container calls the methods of each component, which are necessary for component to perform in order to produce the service expected by other components. These methods form the *lifecycle* of a component.

Contact between a container and contained component is done exploiting component lifecycle. A container is required to take a component through certain stages imposed by its lifecycle. These stages are the methods that are to be called on the component by a container. Lifecycle specifies which exactly method should be called and the order in which this should happen.

### 4.4.5 Design Rationale

After the analysis of existing component-based frameworks was done, Fortress was taken as basis for the *Fahrsim* application. Fortress is a well-documented and tested software framework. It enables loose coupling of components and provides means to initiate only needed components for particular run without recompilation of the source code. It also includes lifecycle interfaces needed for the *Fahrsim* components to fulfill requirements of organizing the data flow and implementing driving assistance. Fortress is a relatively easy and light-weighted framework, which allows the developer to focus on design and implementation of wanted functionality. In the next chapter benefits of Fortress and their usage in *Fahrsim* are explained in detail.

## *4.5 Modules of Fahrsim*

There are three main modules in the *Fahrsim* application. Their interaction results in organization of data flow, and provides functionality for the three concepts. Modules are:

1. *manager module*: responsible for running the desired configuration of the program;

2. *socket module*: it is the interface between hardware's serving applications, and it also provides information to assistance modules implementing driving assistance. It receives, processes, and sends data to driving simulator's components;
3. *assistance modules*: consist of the base assistance module which scales the data to meet the requirements imposed by the protocols of the interfacing applications (serves the socket senders, mainly), and also modules which are extensions to a driving standard functionality in which the driving support of the three concepts is implemented.

Implementation of the previously mentioned modules involves extension of interfaces provided by the Fortress framework. Fortress interfaces used in simulator's software are

1. `LogEnabled`: components implementing this interface or extending Fortress's abstract class `AbstractLogEnabled`, can use default logging system introduced by the framework. The Fortress's logging classes are developed based on `Log4J`[13] package. `Log4J` provides ability to change the logging strategy by editing the configuration file without modifying the source code, and also provides the benefits of the hierarchical logging, meaning different level of detail can be activated, i.e. the developers during the debugging phase need detailed description of the errors, and operator which performs the experiments needs only output information of the assistances.
2. `Serviceable`: components implementing this interface should defined method `service`, in which service manager is passed. Service manager includes components, which are expected to be served by the component implementing `Serviceable` interface. This interface facilitates the creation of the communicational network among the components.
3. `Configurable`: components implementing this interface should define method `configure`, in which configuration object created from

---

[13] http://logging.apache.org/log4j/docs/ accessed 17.12.06

configuration xml file is passed. Attributes of a component which alter depending on current setup of the system are no longer needed to be hard coded; they can be changed in corresponding section of the configuration file by the non-programmer operator.

4. `Initializable`*:* components implementing this interface define a method `initialize`. Sockets used to establish communication flow in driving simulator should be properly set up before passing the data. In method `initialize` socket is created, and communication channel is activated.
5. `Startable`*:* components implementing this interface define a method `start`, which allows starting the component without requiring separate thread. It is used in manager module, where the control of the program is passed to the container.

Based on the specific functional needs imposed on components by driving simulator functionality, new interfaces needed to be introduced. These are called so-called *simulator generic interfaces* and are used by Fahrsim components based on their activities and interactions with other components. The description of simulator generic interfaces is given below.

- `Handleable`: allows application components to process data that are received from other application components.
    - Implemented Fortress interfaces: none.
    - New methods: `isValidData` – checks if the data passed to the component is needed for its functionality, `handleData` – performs the activities triggered by passed data.
- `Pushable`: allows components to pass data to other components. This interface provides the ability for the component to interact with components, which implement `Handleable` interface.
    - Implemented Fortress interfaces: `Configurable`, `Serviceable`.

  - New methods: `pushData` – delivers the data to dependent `Handleable` components by calling their `handleData` function with the data as input argument.
- `Retrievable`: allows components to retrieve data from other components.
  - Implemented Fortress interfaces: `Configurable`, `Serviceable`.
  - New methods: `retrieveData` – returns data from some component. The data needed and the component which contains it are specified as input parameters to this method.
- `Returnable`: allows components to return data. Components implementing this interface are coupled with the components implementing Retrievable interface.
  - Implemented Fortress interfaces: none.
  - New methods: `isCorrectDataID` – checks if the data requested can be returned by the component, `returnData` – returns object which represents the data requested.
- `Threadable`: allows components to run in their own thread. Some of the components need their own threads to run throughout the performance of the program, i.e. receiving sockets which are responsible for facilitating continues data flows.
  - Implemented Fortress interfaces: none.
  - Implemented Java interfaces: `Runnable`.
  - New methods: none.

Specific interfaces were also developed for each module. For assistance module `Assistanceable` interface was introduced, for socket module – `Socket`, `Sendable`, `Receivable`. These interfaces mirror the role of components based on their purpose (either to assist, either to be communication link) in *Fahrsim*.

### 4.5.1 Manager Module

The manager module is the entry point of the *Fahrsim* application. In `main` method, a container (see 4.4.4) is set up. The components to be initialized are read from configuration file. Configuration files enable to define which components needed for the particular run of the driving simulator; in these files the user can also reconfigure attributes of initiated components without recompilation of the source code. Also, in manager module the logging is enabled through logging configuration xml file. After reading the setting of current run and creating component's information, the container is then takes its components through their lifecycle stages.

### 4.5.2 Socket Module

The socket module represents the interaction layer between assistance components and outer applications. The socket module is responsible for receiving the data from the driving simulator serving software, passing it to assistance components for further processing, and sending required data to the driving simulator interfacing software.

The interface `Socket` is introduced for socket module. It implements Fortress's `Threadable`, `Serviceable`, `Configurable`, and `Initializable` interfaces. Each socket is supposed to run its own thread. Information of the socket is read from configuration file and passed as configuration object by the container to `configure` method. In `initialize` method the socket is initialized and connection set.

If needed, socket component can have receiver or/and sender components to be served. References to receiver and sender components are created in `service` method. These components are responsible for appropriate parsing of information and passing it further; in case of receiver the data is pushed to assistance components, and in case of sender it is sent to corresponding simulator component's interfacing software. Sender components implement interface `Sendable`, and receiver components – `Receiveable`.

There are seven sockets in the *Fahrsim* application:

- LabView socket, including receiver and sender,
- Driving Dynamic socket, including receiver and sender,
- Traffic socket, only receiver,
- Steering wheel socket, including receiver and sender,
- Mock-Up socket, including receiver and sender,
- HUD socket, including receiver and sender,
- Conformal HUD socket, including receiver and sender.

On Figure 8 the interaction between driving simulator interfacing software applications and *Fahrsim* is shown.

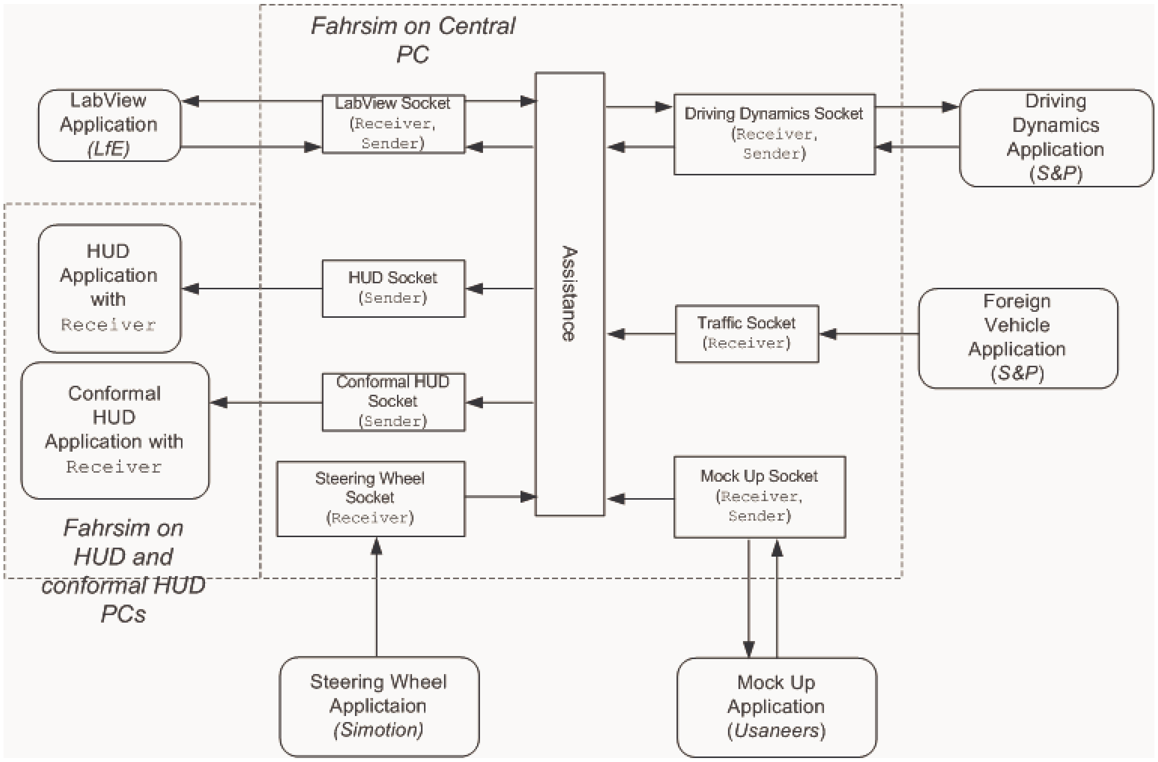


*Figure 8: Data flow between driving simulator interfacing software and Fahrsim application*

LabView application reads gas and brake data from the measurement card. It also receives data about the state of the control buttons: which driving support is activated, whether wanted speed was decreased or increased, and whether desired following distance was changed. These data are passed to the LabView socket, which receives the packet and passes it to its receiver component. LabView receiver component parses data packet, and sends corresponding information to the assistant components which require it:

1. Brake Pedal assistant (base assistance for any run of the simulator, always enabled);
2. Gas Pedal assistant (base assistance for any run of the simulator, always enabled);
3. Level of driving support assistant (assistance enabled when the three concepts need to be introduced to the driver);
4. Wanted Speed assistant (assistance enabled when the three concepts need to be introduced to the driver during the run);
5. Desired following distance assistant (assistance enabled when the three concepts need to be introduced to the driver during the run);
6. Sound assistant (assistant which is enabled in order to give the driver the acoustical feedback of the driving);
7. Data Recorder assistant (assistance enabled during experiments, allows to record data during the run for further analysis).

The sender component of the LabView socket retrieves the data about wanted speed, desired following distance, level of driving support, and voltage to be set on the active gas pedal if AGP or ACC driving support functionality is enabled. It then sends these data to the LabView application.

Driving Dynamics socket receiver becomes data about the state of our own car from driving dynamics application (see section 4.2.2). It parses received packet and sends values to the following assistance components:

1. Active Gas Pedal and Cruise Control assistant (assistance enabled when the three concepts need to be introduced to the driver during the run);
2. HUD socket sender;
3. Conformal HUD socket sender;
4. Data Recorder assistant;
5. Sound assistant.

Driving Dynamics socket sender is responsible for delivering information to driving dynamics application about gas pedal position, brake pressure, and steering wheel angle. It retrieves these data from the base assistance module. If active cruise control is activated, Driving Dynamics socket sender retrieves data about calculated cruise control gas and cruise control brake values, and sends them instead of real values read from measurement card. However, if the real gas position is more than calculated active cruise control gas, the real gas is sent. Here should be also noted that real brake pressure is always 0 when active cruise control is activated, because once the driver presses brakes, full manual control is given beck to him automatically.

Traffic socket only receives the information about the state of other cars (see section 4.2.3), and also additional information about our car, such as its position in the world and on which road and lane it is driving. It pushes the data to the Data Recorder assistant, and also to the AGP and ACC assistant to calculate the distance to the car in front and to activate the desired following distance rule if needed.

Steering wheel receiver pushes the information about current steering wheel angle to Steering Wheel assistant. Sender will be sending additional moment to be applied on the steering wheel in the future (Penka, 2001).

Mock-up receiver is responsible for processing the information about pushed blinker buttons. Sender, in its turn, sends back to mock-up serving application which blinker should be activated.

HUD receiver and sender sockets run on different computers. Form the central *Fahrsim* computer sender socket sends information about which driving support is performed, which one of 2D or 3D optical representations is chosen, whether following distance rule is activated, also current speed, wanted speed, and desired following distance. On HUD computer, HUD receiver captures these data and pushes it to HUD assistant to display corresponding symbols in the HUD.

Conformal HUD sockets function similar to regular HUD sockets. Only the information passed between the central program *Fahrsim* and conformal HUD application differs. The data packet includes current steering wheel angle, current speed, desired following distance, and which representation is to be activated (no bar/green bar/orange bar).

Data flow between socket components and assistance components is shown on Figure 9, pg.40.

### 4.5.3 Base Assistance Module

A pure driving simulator uses only assistances from the base module (Gas Pedal, Brake Pedal, and Steering Wheel assistances). All further assistance modules were integrated later in order to implement the three concepts.

The detailed description of the assistances modules both base and responsible for the three concepts is given in the following Chapter 5.

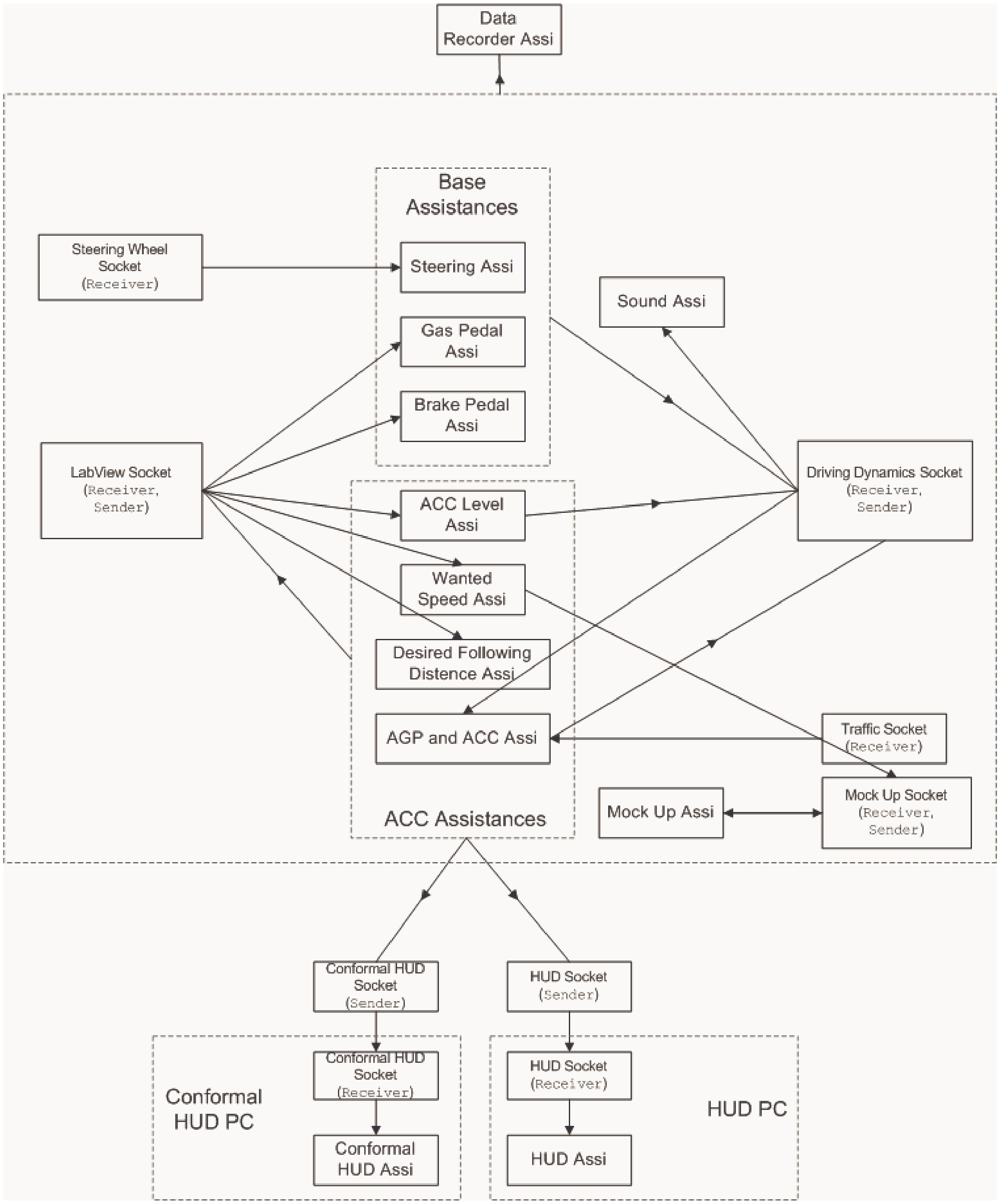


*Figure 9: Data flow between communication and assistance components in Fahrsim application*

# 5 Implementation of Assistance Concept in Fahrsim

As the part of the *Fahrsim* program, new assistance modules realizing the three concepts in the driving simulator were introduced after a person could drive in the simulator.

The assistance modules realizing the three concepts are able to gather data, process it, and trigger the driving support systems when correlated events occur.

## *5.1 Assistance Modules*

Assistance components are divided into several packages based on their functional role in implementation of driving support. The assistance modules are:

1. base assistances
    a. gas pedal assistant
    b. brake pedal assistant
    c. steering wheel assistant
2. medium assistances
    a. level of driving support assistant
    b. wanted speed assistant
    c. desired following distance assistant
3. Conformal HUD assistant
4. Regular HUD assistant
5. Sound assistant
6. Data recorder assistant
7. Active Gas Pedal and Cruise Control Assistant

Primary assistances are gas, brake, and steering wheel assistances. They are included in the base sub package. In order to be able to drive in the simulator, one should include these components and corresponding socket components in the configuration file. During the run of the program, these components receive data from socket objects, filter and scale it in the way other components expect it.

All primary assistances implement `Returnable` interface. It allows Driving Dynamics socket sender to retrieve and then send primary data about gas, brakes, and steering wheel angle expected by driving dynamics application (Figure 9, pg.40).

Brake pedal and steering wheel assistants also implement `Pushable` interface. It results in triggering the events which are dependent on the value of the bake pressure and steering wheel angle. Steering wheel assistant pushes data to conformal HUD socket sender, and Brake pedal assistant pushes current value of brake pressure to Level of Driving Support assistant. If the system is in “ACC” mode, but the driver has pushed the brake pedal, it switches to “Aus”.

Medium assistances are responsible for processing the information regarding wanted speed, wanted distance, and driving mode. They implement `Pushable` and `Returnable` interfaces.

Wanted speed assistant keeps the last set wanted speed, and increases it or decreases it depending on which button was pressed. Afterwards the actual wanted speed value is pushed to Active Gas Pedal and Cruise Control assistant (Figure 10, pg.44), regular HUD socket sender, and Mock-up socket sender (Figure 9, pg.40). Desired following distance assistant functions similar to wanted speed assistant. However, its value is not pushed to Mock-up sender. Level of Driving Support assistant pushes its value only to Active Gas Pedal and Cruise Control assistant, which triggers calculation either voltage to be set on active gas pedal, or active cruise control gas and brakes values (Figure 10, pg.44).

However, some of the components need to retrieve the values from medium assistances. LabView socket sender retrieves values from medium assistances to send it to LabView program. Driving Dynamics sender socket retrieves level of driving support in order to decide if to send real gas, or active cruise control gas and brakes (Figure 9, pg.40).

Conformal HUD assistant is responsible for calculating the placement of following distance bar and displaying it in the right color: green or orange, in case if following distance rule is activated.

Regular HUD assistant displays symbols according to activated level of driving support and optical representation.

In order to give a driver a feeling of real driving, Sound assistant is implemented. It plays the sound of engine, and wind noise. Engine sound is adjusted according the engine speed and throttle position. Wind sound depends only on the speed of the car. Subjective impression of the people driving in simulator is that together with steering wheel feed-back, sound tremendously enhances the feeling of real driving.

Evaluation of experiments is performed based on the data recorded during the run. Information about the state of the gas, brakes, steering wheel angle, and safety distances is necessary in order to analyze the impact of particular driving support. Therefore Data Recorder assistance is implemented. It implements both `Retrievable` and `Handleable` interfaces.

Active Gas Pedal and Cruise Control assistant is responsible for calculating voltage for active gas pedal, if “AGP” mode is on, or active cruise control gas and brake values in case of “ACC” mode. This assistant also provides the information if the following distance rule is activated, and if so, what is the distance to the car in front.

In order to perform calculations, Active Gas Pedal and Cruise Control assistant becomes information from Driving Dynamics socket receiver (our own vehicle data), Wanted Speed assistant, and Desired Following Distance assistant (Figure 10, pg.44). Calculated values of active gas pedal voltage, or active cruise control gas and brakes are retrieved by LabView socket sender, and, if ACC or AGP support is enabled, by Driving Dynamics socket sender (Figure 9, pg.40). Data concerning the activation of following distance rule is retrieved by regular and conformal HUD socket senders (Figure 9, pg.40).

The interaction needed for implementation of the three concepts between assistant components is shown in the Figure 10.

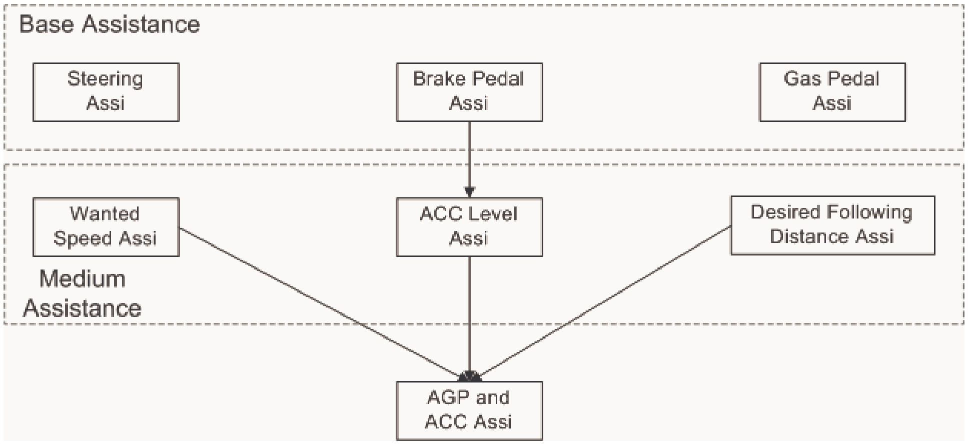


*Figure 10: Data flow between assistance components in assistance modules*

## 5.2 Active Gas Pedal and Active Cruise Control Implementation Algorithm

The algorithm behind calculation of gas pedal’s required position is based on *control theory*. It aims to find at each point of time such position for a gas pedal that will result in reaching wanted speed at desired rate, and, once wanted speed is reached, to keep it as stable as possible.

### 5.2.1 Control Theory

Control theory deals with dynamical systems. A *dynamical system* is a system that changes its state over time. A *controller* is responsible to set the inputs to dynamical systems implemented with respect to control theory. The controller modifies the input in order to obtain the desired effect on the system’s output. The dynamic system operated by a controller is called *control system*. Depending on the processed variables used by the controller, two types of control systems are defined:

1. open-loop control system;
2. closed-loop control system.

In the *open-loop control system*, the controller does not use the output of the system as one of its parameters which is processed in order to find the right input to the system at the next time step. On contrary, in the *closed-loop control system* the output is used as one of the parameters processed by the controller. The basic representation scheme of the closed-loop control system is given by Figure 11.

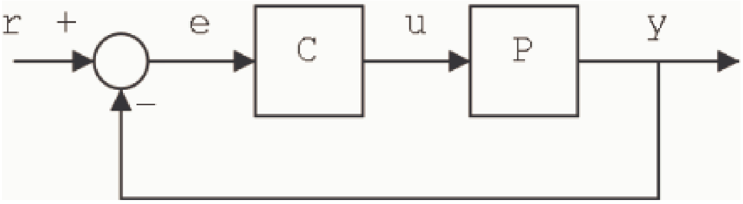


Figure 11: Basic closed-loop control system

On Figure 11, the output value *y* is subtracted from reference value *r*, in this way error *e* is determined. The error term *e* is then processed by the controller *C*, and resulting value *u* is used as an input to the system *P*. The goal is to reach the difference between *y* and *r* as small as possible.

The *stability* of the system is another important issue in control theory. It is usually referred to as *Bounded Input/Bounded Output (BIBO)* stability. It states that for any bounded input over any amount of time, the output will also be bounded.

For any stable closed-loop system, the response can be classified as one of three types of damping that describes the output in relation to the resting state of the system. *Resting state* of the system is the state, when the outputs starting from $i$+1 ($i \in \mathrm{N}$) iteration are equal among themselves for a constant input. The types of the system based on the behavior of the response to a *step input* are:

1. underdamped;
2. overdamped;
3. critical damped.

In *underdamped* systems, the response oscillates within a decaying envelope before reaching the resting state. In *overdamped* systems, the response does not overshoot the resting state value. The more underdamped or overdamped the system is, the longer it takes to reach the resting state.

*Critical damped* systems are the systems which reaches the resting state the fastest with at most one overshoot. Usually it is derived by tuning the parameters of the overdamped system, until the *settling time* (time for a system to reach the resting state) decreases, and no more than one overshoot appears.

One of the most-used closed-loop controllers is *Proportional-Integral-Derivative (PID)* controller. The input to the system *u(t)* which is generated by PID controller is derived by the equation (1)

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \dot{e}(t), \qquad (1)$$

where $e(t)$ is the error representing the difference between the desired output of the system and the measured output of the system; $\dot{e}(t)$ is the derivative of the error with respect to time; $K_P$, $K_I$, $K_D$ - proportional, integral, and derivative coefficients, correspondingly. The system is usually brought to the desired closed loop dynamics by tuning these three coefficients.

It is necessary to describe the effects of each equation’s term on the system behavior. Stability of the system is guaranteed by presence of the proportional term. To anticipate and reject a step disturbance, the integral term is used. If the behavior of system shows that additional damping of the output is needed, the derivative term is introduced.

For implementation of the active gas pedal and the active cruise control as driving support in the driving simulator, PID controller is used.

### 5.2.2 Algorithm of Finding Required Gas Pedal Position

The algorithm is presented on the diagram below (see Figure 12).

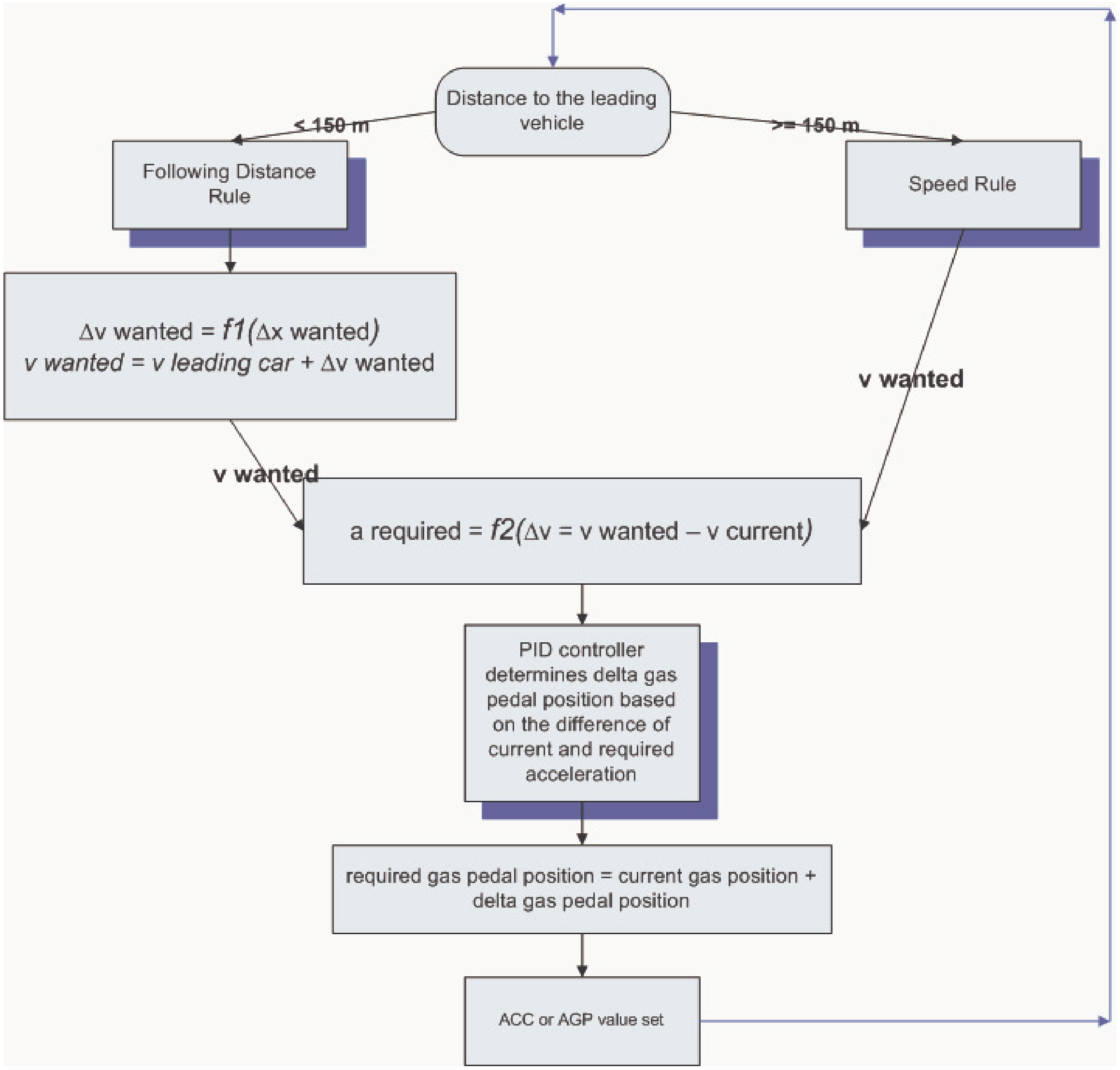


*Figure 12: Algorithm of finding required gas pedal position*

The following distance rule and speed rule, which allow determination of wanted speed for our own car, are described in the following section. Afterwards the function of required acceleration is presented. It is dependent on the difference between current speed, and found wanted speed. Once required acceleration is found, PID controller is processing it in order to find delta gas pedal position, which is then added to the current gas pedal position. Resulting value is used as the input for the system. PID controller is also described in the following sections.

### Wanted Speed: Speed Rule and Following Distance Rule

Two rules that define wanted speed for the car are introduced in implementation of “AGP” and “ACC” concepts:

1. speed rule;
2. following distance rule.

If no vehicle is detected in front of the car, the *speed rule* defines wanted speed which car is supposed to maintain. The value of wanted speed is set manually by the driver.

The *following distance rule* is activated, when the vehicle in front of our car driving at the same lane is detected, and the distance in between is less than some threshold (in our implementation less than 150 m). The goal of the system is to reach and keep the preset desired following distance to a leading vehicle:

*Required Following Distance = Wanted Distance [s] * Speed of Vehicle In Front [m/s].* (2)

In our implementation, wanted distance chosen by the driver can be either 0.9 or 1.5 seconds.

After the difference between current distance to the leading car and required following distance is determined, piece-wise linear function represented on Figure 13 is used to find intermediate value *delta wanted speed*.

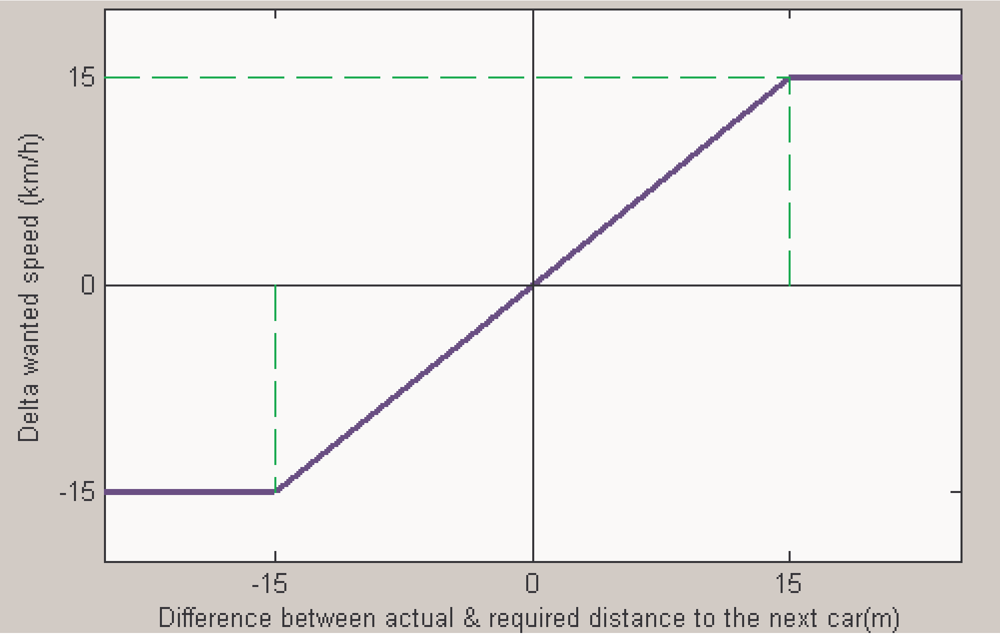


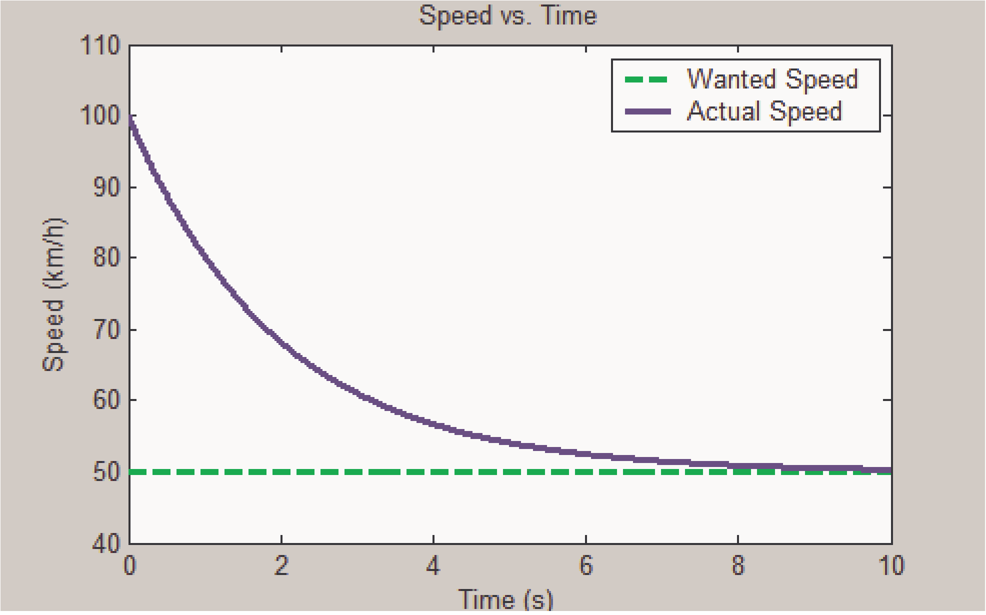
*Figure 13: Piece-wise linear function which maps the difference between actual and required distance to the leading car (m) to delta wanted speed*

After delta wanted speed for the following distance rule is found, wanted speed is calculated by addition of delta wanted speed and the speed of the leading car.

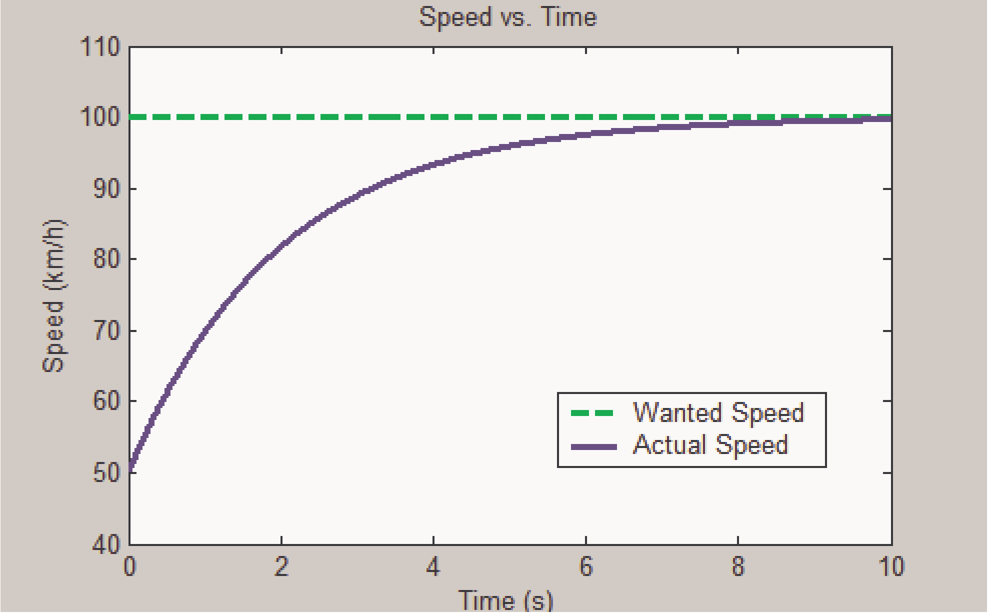
Wanted speed either taken with respect to speed rule, or with respect to the following distance rule, is further used to find desired acceleration (see next section), which is then used to find desired gas pedal position.

### Required Acceleration

Exponential decay function was chosen as the model representing how the speed of the car should change over time to reach the wanted speed. On Figures 14a and 14b, pg.50, the graphs show how current speed approaches wanted speed over the course of time.



(a)



(b)

*Figure 14: Current speed approaches wanted speed in exponential manner over the time; in the case (a) initial current speed is greater than wanted speed, and in (b) initial speed is lower than wanted speed*

The model function for both cases is

$$v_{wanted} - v(t) = c_1 e^{-c_2 t}, \tag{3}$$

where $v(t)$ – the speed of the car, $v_{wanted}$ – wanted speed to reach, $c_1, c_2$ – shaping coefficients, where $c_2>0$ for both cases, and $c_1<0$ when the wanted speed is lower than current speed, and $c_1>0$ otherwise. As the result of differentiation of the equation (2) with respect to time

$$\frac{dv}{dt} = c(v_{wanted} - v(t)), \tag{4}$$

is derived, where $\frac{dv}{dt}$ is the acceleration, and $c$ – some coefficient. As it can be deducted from (4), the required acceleration is linearly dependent on the difference in wanted speed and current speed. Taking into account allowed maximum and minimum acceleration (deceleration), the resulting function for computing required acceleration is represented by the Figure 15.

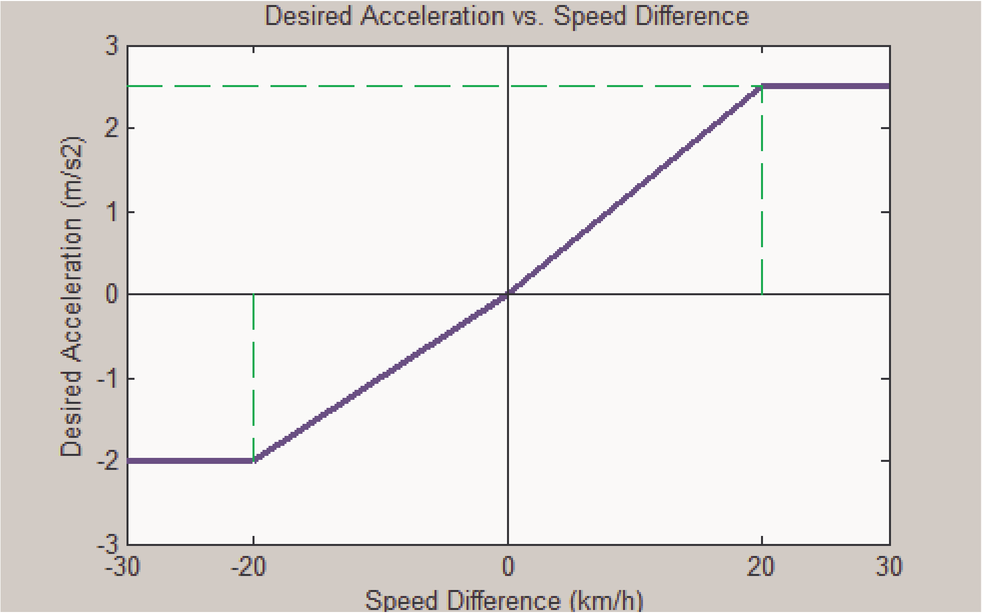


*Figure 15: Piece-wise linear function which maps difference between wanted and current speeds to required acceleration*

The difference in current and wanted speed, after which full acceleration (or full deceleration) is proposed, was found experimentally. In its derivation was considered, that the driver should not experience steep changes in acceleration, but, on the other hand, required acceleration should be enough to reach the wanted speed in reasonably fast time. In our case, if the difference in wanted and current speed is larger than 20 km/h, the maximum allowed acceleration or deceleration is proposed.

After required acceleration is found, it is used by PID controller as the reference value for calculation of the delta gas pedal position.

### PID Controller

Active cruise control and active gas pedal concepts are described in the sections 3.2.2 and 3.2.3. The main goal of implementation of these concepts was to find the gas position at each time step, which would result in reaching the wanted speed at the desired rate without lasting oscillations, and keeping reached speed afterwards.

A control system provides the possibility to implement the desired functionality. The aim was to develop critical damped system using PID controller design with introducing the wanted speed as the resulting resting state of the system.

The parameters of the control system realizing active driving support need to be described. The acceleration is taken as the system output. The error term used by the controller is the difference in desired acceleration (reference value) and current acceleration (output of the system). Gas pedal position is the input to the system which should be found with the help of the controller. The actual parameter calculated by the controller is the change in gas pedal position which is then added to the current gas pedal position, and the resulting value is taken as the input to the system at the next step.

*Proportional-Derivative* (PD) design is the underlying concept of the implemented controller. It is similar to PID design, only the integral term is omitted. Presence of only proportional term experimentally was shown to be sufficient if the speed of our car is below or slightly above the wanted speed (in our implementation, the difference in speeds does not exceed 1 km/h). The proportional coefficient is calculated by equation (5):

$$K_p = \frac{\alpha_{gas}}{a_{\max}}, \tag{5}$$

where $\alpha_{gas}$ is the angle (in degrees) of gas pedal position which produces appropriate acceleration when the current speed is close to the wanted speed and brings the system to the rested state not causing oscillations (in our implementation the angle is equaled to 4°), and $a_{\max}$ is maximum acceleration (in our implementation 2.5 m/s$^2$).

If the difference between driven and wanted speed is larger than threshold (in our implementation 1 km/h), the derivative term is introduced into the equation. In this case the controller amplifies the output parameter by derivative term with coefficient experimentally found to be suitable.

In active cruise control system, found gas position is send to driving dynamics directly. If drivers override the system, the actual gas pedal position is sent instead.

In the case of AGP, found gas pedal position is mapped to the voltage value which should be set on active gas pedal's motor to create resistance point at the required position. Voltage value is retrieved from Active Gas Pedal and Cruise Control assistant by LabView socket sender, and is sent to LabView PC which in its turn sets the value on gas pedal's motor.

### 5.2.3 Active Brake Pedal

*Active brake pedal* concept is used in “ACC” mode. If the gas pedal position is at rest, but the desired acceleration is still smaller than received current acceleration, the system is allowed to brake automatically. For the driver comfort and safety reasons, the deceleration produced by applying automatic braking should not be lower than -2.5 m/s$^2$.

The modified design of PID controller is used in the implementation of active brake pedal functionality. Experimentally it was shown that presence of only proportional term is sufficient for the system to be at critical damping state. The error term used in PID is the difference between current acceleration and desired acceleration.

# 6 Set Up of the Experiment

Effects of the three implemented concepts (“Aus”, “ACC”, “AGP”) on driving behavior and driver’s workload will get investigated in an upcoming experiment. The main questions to be answered by the results of the experiment are:

- How do different levels of driving system’s automation (concepts “ACC” and “AGP”) influence driver’s performance compared to unassisted driving (concept “Aus”)?
- What level of automation (see pg.8-9) drivers are ready to admit?
- What are the effects on driving performance of using only a regular HUD? Combination of a regular HUD and a conformal HUD?

In the following sections describe the requirements which are imposed on the experiment and provide information needed to conduct the experiment. Therefore this chapter provides a tutorial for the experimenters of the upcoming experiment.

## *6.1 Participants*

In order to perform and evaluate the experiment, 24 participants are needed for statistical purposes (Bortz and Jürgen, 1985). Overall 12 women and 12 men should perform the drives. The age of participants should be evenly distributed between 18 and 70.

## *6.2 Experimental Design*

This section describes variables that have to be calculated during the analysis of performed experiments.

### 6.2.1 Independent Variables

A single-session within-subjects design (also known as repeated-measures design) is used, with all drivers experiencing the three concepts described in section 3.2.1, 3.2.2, and 3.2.3. The independent factors are the concept (“Aus”, “ACC”, “AGP”) and model of optical representation of information related to the driving (2D model or 3D model). The description of the factors and their levels is given in Table 1.

*Table 1: Independent factors*

| **Factor** | **Level** | **Description** |
|---|---|---|
| Concept | 1. “Aus” | Driver is given full manual control over the car. |
| | 2. “ACC” | Driver delegates regulation of speed and following distance to the vehicle in front to the assistance system. The system is responsible to reach and keep wanted speed and desired following distance set by the driver. The driver is also able to override the system. |
| | 3. “AGP” | Driver is offered longitudinal assistance in keeping wanted speed and desired following distance which is realized through haptical feedback on the gas pedal. |
| Optical representation of information | 1. 2D Model | Current speed, wanted speed, and desired following distance are all shown in HUD. |
| | 2. 3D Model | Current speed and wanted speed are both shown in HUD. Desired following distance is portrayed in the conformal HUD by the bar. |

### 6.2.2 Dependent Variables

Dependent variables consist of subjective and objective measures. The following sections define and describe these measures in detail.

#### 6.2.2.1 Objective Measures

Objective measures reflect the driver's performance based on data recorded during the run of an experiment. It consists of values that provide information on lateral and longitudinal behavior of the driver, as well as information on driver's distraction during the experiment through glance measures.

***Driving Performance Measures***

Driving performance measures reflect longitudinal and lateral behavior of the participant during experimental drives. Main longitudinal measures to be calculated and analyzed are:

- *average speed:* the average speed (km/h) at given road segment with posted speed limit;
- *median speed difference:* the average speed minus posted speed limit (km/h) at a given road segment;
- *speed deviation:* standard deviation of the vehicle speed (km/h);
- *considerable speed exceedence:* proportion of time during which the vehicle's speed is greater than posted speed limit for more than 10 km/h;
- *gas pedal angle variation*: standard deviation of the gas pedal angle;
- *average Time-To-Collision (TTC):* average time required for two vehicles (for the experiment: our vehicle and vehicle in front) to collide if they continue driving at their present speed and on the same path (s) (Hayward, 1972);
- *average following distance to the vehicle in front* (s);
- *minimal TTC:* the minimal value of calculated TTCs; indicates how imminent an actual collision has been (Van der Horst and Hogema, 1993).

Lateral measures are related to the lane keeping performance. Measures to be taken from the experiment's recorded data are:

- *average lane position:* average lane position relative to center of own lane of the vehicle during performed drive (m);
- *lane deviation:* standard deviation of the lane position (m);
- *average Time to Line Crossing (TLC):* average time required for the vehicle, following its trajectory and keeping its present speed, to cross the lane (s);
- *median TLC*
- *steering angle variation:* standard deviation of the steering wheel angle (deg)
- *large steering corrections:* adoption of steering wheel reversals. A large steering correction is defined as a continuously increasing or continuously decreasing segment where the change in steering wheel angle is greater than 3.7 degrees (Wierwille and Gutman, 1978);
- *rate of large steering corrections:* number of large steering corrections per second.

The values, based on which the upper described variables should be calculated, will get recorded with the data recorder of the driving simulator's system. Data recorder is one of the components of the *Fahrsim* program, and it is started once the experimental drive begins.

### *Glance Performance Measures*

To analyze driver's distraction and workload, glance behavior measures are used. These measures include median saccade angle, number of saccades per minute, medium fixation duration, percentage of time spent looking at the speedometer, and percentage of time spent looking at the areas that are not relevant to driving, e.g. trees, houses, etc.

The glance behavior of the driver is recorded with appropriately calibrated DIKABLIS eye-tracking system[14].

#### 6.2.2.2 Subjective Measures

Subjective measures should be taken as the complimentary information to objective measures. This includes questionnaires about subjective impression of the experienced driving support: how helpful a participant found particular driving assistance, how comfortable they felt themselves using it, etc. The participant is also to be asked to evaluate their driving performance: how well they could keep the wanted speed and desired following distance to the vehicle in front, how good they could keep the lane position throughout the drive. The comparison to objective results enables determination of how objective results match personal experience during the drive. It is necessary for evaluation of the acceptance degree and comfort level of the driving support system.

In addition, NASA TLX[15] questionnaire is given to participants. This allows performance of subjective workload assessments on drivers who experience various automated driving support assistance. The results of NASA TLX questionnaire have to be compared with glance measures. Correlation between eye movement and experienced workload must be investigated to determine the dependency of eye behavior depending on the participant's strain while performing the task.

## *6.3 Procedure*

During the first conversation with the participant, basic demographic questions (age and gender) are to be asked. Afterwards the appointment (approximately 2 hours) should be scheduled in the driving simulator. Participants are to be tested individually.

---

[14] Ergoneers GmbH, Manching, Germany
[15] NASA Task Load Index (TLX) V1.0 Users Manual

The person conducting the experiment is responsible for meeting the participant. Afterwards the demographic questionnaire (Appendix B) together with introduction to the experiment and explanation on the three concepts are to be given to the subject. Once the corresponding forms are filled in, necessary preparations for the eye-tracking system should be performed. The participant is explained the purpose of the DIKABLIS glasses; the following phase is the calibration of the eye-tacking system with help of the participant. Once the eye-tracking system is set up, the subject is asked to perform a test drive.

The purpose of a test drive is to get the participant used into driving in the fix-based simulator. Duration of this phase depends on how the participant feels themselves comfortable while driving in the simulator; it should be taken into account by the conductor of the experiment that this phase usually takes from 10 up to 25 minutes.

After participant feels themselves comfortable while driving at fix-based simulator, the actual experimental drives start. The subject is to perform 5 drives in randomized order. After each experimental drive, the participant is asked to how well they could operate the system and what disadvantages and advantages of driving support they found during their drive. The corresponding question form with full list of questions is provided in Appendix C. After all 5 drives are performed, the subject is to fill in a final form (Appendix D) to summarize their impression about the experienced driving support and to rate the concepts in terms of how comfortable they feel using them, and how helpful in terms of speed and lane maintenance they found it.

# 7 Conclusion and Future Work

The closing chapter summaries the work performed in terms of this master thesis. It gives a summary of fulfilled tasks and briefly lists future work to be done to evaluate the already implemented driving assistance system and to extend driving support at the driving simulator for future experiments.

## *7.1 Summary*

Design and implementation of driving assistance systems in the driving simulator was the main task fulfilled by the master thesis work. First, three concepts of driving assistance were designed. These are the concepts “Aus”, “ACC”, and “AGP”. They operate on different levels of automation. Concept “Aus” represents driving without automation support, and no longitudinal or lateral assistance presented to the driver. As opposite to non-automated driving concept, “ACC” was introduced. It provides driver with automation support by overtaking the driving task of accelerating and braking (till allowed limits), thus putting the driver out of the control loop. Under terms of this concept, the driver regains control over driven vehicle by pressing cancel button, or by using the brake pedal. Concept “AGP” lies in between the other two concepts: the driving support system provides longitudinal support by suggesting the position of the gas pedal, but the driver is free to follow advices of the system, or to enforce their own decision. In the “AGP” concept, operator always stays in the control loop of the driving system.

In terms of the three concepts, two optical information representation models were developed. A 2D model provides driver with optical information about car’s state by displaying corresponding symbols in the HUD. In the 3D model a conformal HUD is used for displaying a following distance bar. In this case the driver is provided with additional visual, spatially embedded (AR) lateral and longitudinal assistance.

To implement and perform experiments on the three concepts, a driving simulator was set up. It assembles hardware and software components provided by different firms. These components were properly installed and communication among them was established to be able to drive in the simulator. I had to study 3rd party interfacing software, and develop my own management application *Fahrsim* which organizes the data flow in the driving simulator.

The developed application is based on the free available framework Fortress designed by the Apache organization. Besides enabling normal driving in the driving simulator, it also implements the three concepts of assisted driving.

To implement ACC and AGP functionality, first functionality algorithms were designed. These are based on control theory, and are realized with the help of a PID controller. Parameters for algorithms were found and adjusted in accordance with control theory recommendations.

## 7.2 Future Work

As future work, the experiments on full-integrated multi-modal driving assistance system must be performed. They should be conducted properly, evaluated, and results should be taken into account for further extension of the driving simulator and further embedment of assistance concepts.

Functionality of existing driving support, namely the following distance bar shown in the conformal HUD, should be extended. The following distance bar's movement is based on one-lane driving dynamics model. However, at the driving simulator, the two-lane model is the model for own car's behavior. Therefore necessary changes to the movement algorithm have to be realized.

The steering wheel currently does not provide haptic lateral assistance to the driver. For future experiments on the supported driving the steering wheel should present haptic feedback to the driver when the car is about to leave the lane.

# References

Assmann, E. (1985) *Untersuchung über den Einfluss einer Bremsweganzeige auf das Fahrverhalten*. Dissertation, TU-München, Munich, Germany.

Bainbridge, L. (1983). Ironies of Automation. *Automatica*, 19(6), 775-779.

Bauer, M. et al. (2001). *Design of a Component-Based Augmented Reality Framework*. Paper presented at Proceedings of the Second IEEE and ACM International Symposium on Augmented Reality, 2001.

Bernotat, R. (1970). Anthropotechnik in der Fahrzeugführung. *Ergonomics*, 13, 353-377.

Bortz, Jürgen (1985): Lehrbuch der Statistik für Sozialwissenschaftler. Springer, Berlin, Heidelberg, New York, Tokyo.

Brookhuis, K., de Waard, D. (2006). *The consequences of automation for driver behaviour and acceptance*. Paper presented at IEA Congress, Maastricht, The Netherlands, 9-14 July 2006.

Brügge, B., Dutoit, A. H. (2004). *Objektorientierte Softwaretechnik*. Prentice Hall.

Bubb, H. (1976) *Untersuchung der die Anzeige des Bremsweges im Kraftfahrzeug*. In Forschungsbericht aus der Wehrtechnik BMWg, FBWT-7.

Bubb, H. (1993). Systemergonomische Gestaltung. *Ergonomie* (Schmidtke, H., Editor), Hanser Verlag, Munich, Germany.

Endsley, M. R., Kiris, E. O. (1995). The out-of-the-loop performance problem and level of control in automation. *Human Factors*, 37(2), 381-394.

Gish, K.W., & Staplin, L. (1995). *Human Factors Aspects of Using Head Up Displays in Automobiles: A Review of the Literature* (No. DOT HS 808 320). Washington, DC: Office of Crash Avoidance Research, National Highway Traffic Safety Administration.

Green, P., Levison, W., Paelke, G., Serafin, C. (1993) *Preliminary Human Factors Guidelines for Driver Information Systems* (Technical Report No. FHWA-RD-94-087). Ann Arbor, MI: The University of Michigan Transportation Research Institute.

Kaufmann, J. (2004). *Head-up Display: Neue Technik für mehr Verkehrssicherheit*, from http://www.zdnet.de/enterprise/tech/auto/0,39026506,39125753,00.htm, accessed 18.12.06.

Hayward, J. Ch. (1972). *Near miss determination through use of a scale of danger*. Report no. TTSC 7115, The Pennsylvania State University, Pennsylvania.

Hoc, J. M., Young, M. S. (2006). *Driver support systems: how to cooperate?* Paper presented at IEA Congress, Maastricht, The Netherlands, 9-14 July 2006.

IBM Research. (2000). *Multi-Dimensional Separation of Concerns: an Overview*, from http://www.research.ibm.com/hyperspace/MDSOC.htm, accessed 18.12.06.

Lange, C., Tönnis, M., Bubb, H., Klinker, G. (2006) *Einfluss eines aktiven Gaspedals auf Akzeptanz, Blickverhalten und Fahrperformance*. Konferenzband der VDI/VW Konferenz, Germany.

Penka, A. (2001). *Vergleichende Untersuchung zu Fahrerassistenzsystemen mit unterschiedlichen aktiven Bedienelementen*. Unpublished diplom thesis. TU-München, Munich, Germany.

Statistisches Bundesamt Deutschland. (2005). *Verkehrsunfälle 2005,* from http://www.destatis.de/themen/d/thm-verkehr.php , accessed 17.12.2006.

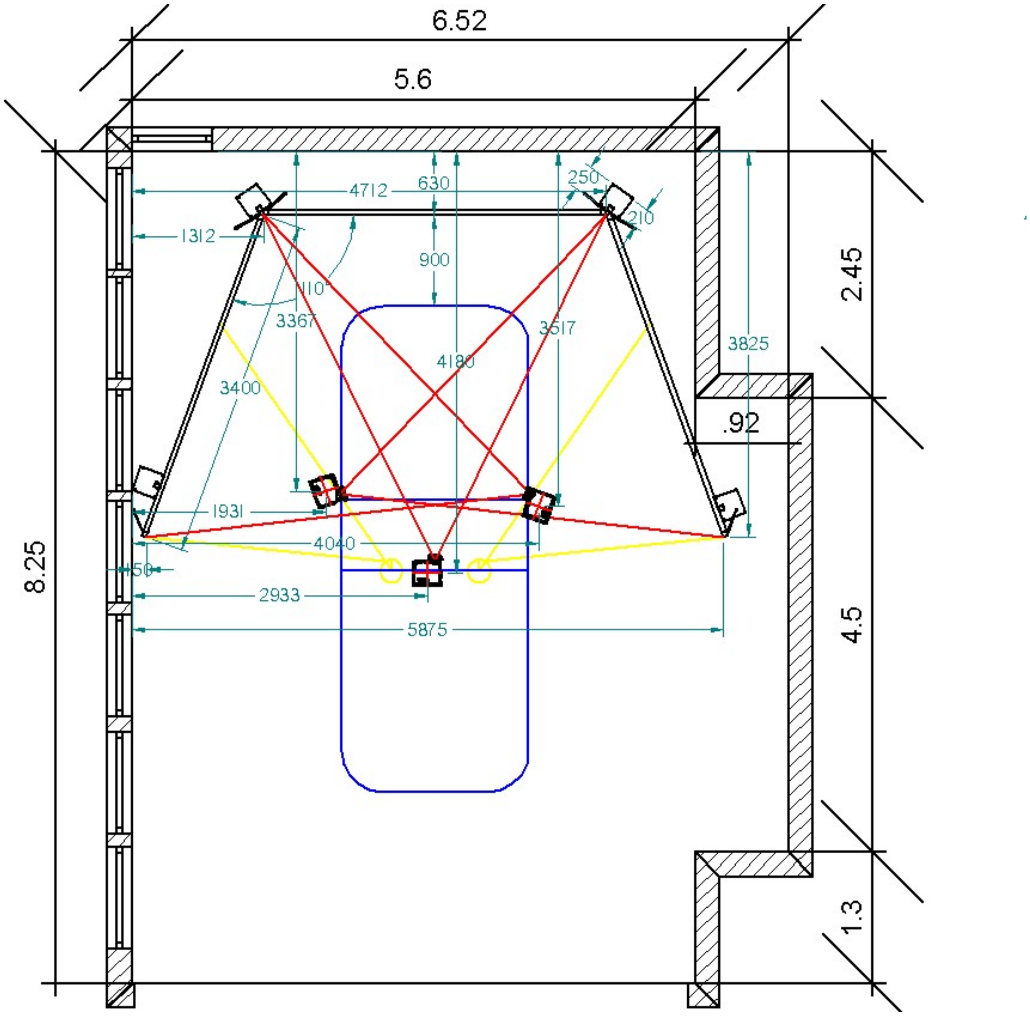
Thompson, L. (2005). *Development and Evaluation of a Driver Assistance Concept for Speed and Lane Assistance*. Unpublished master thesis, TU-München, Munich, Germany.

Tönnis, M. et al. (2006) *Visual Longitudinal and Lateral Driving Assistance in the Head-Up Displays of Cars*. TU-München, Munich, Germany.

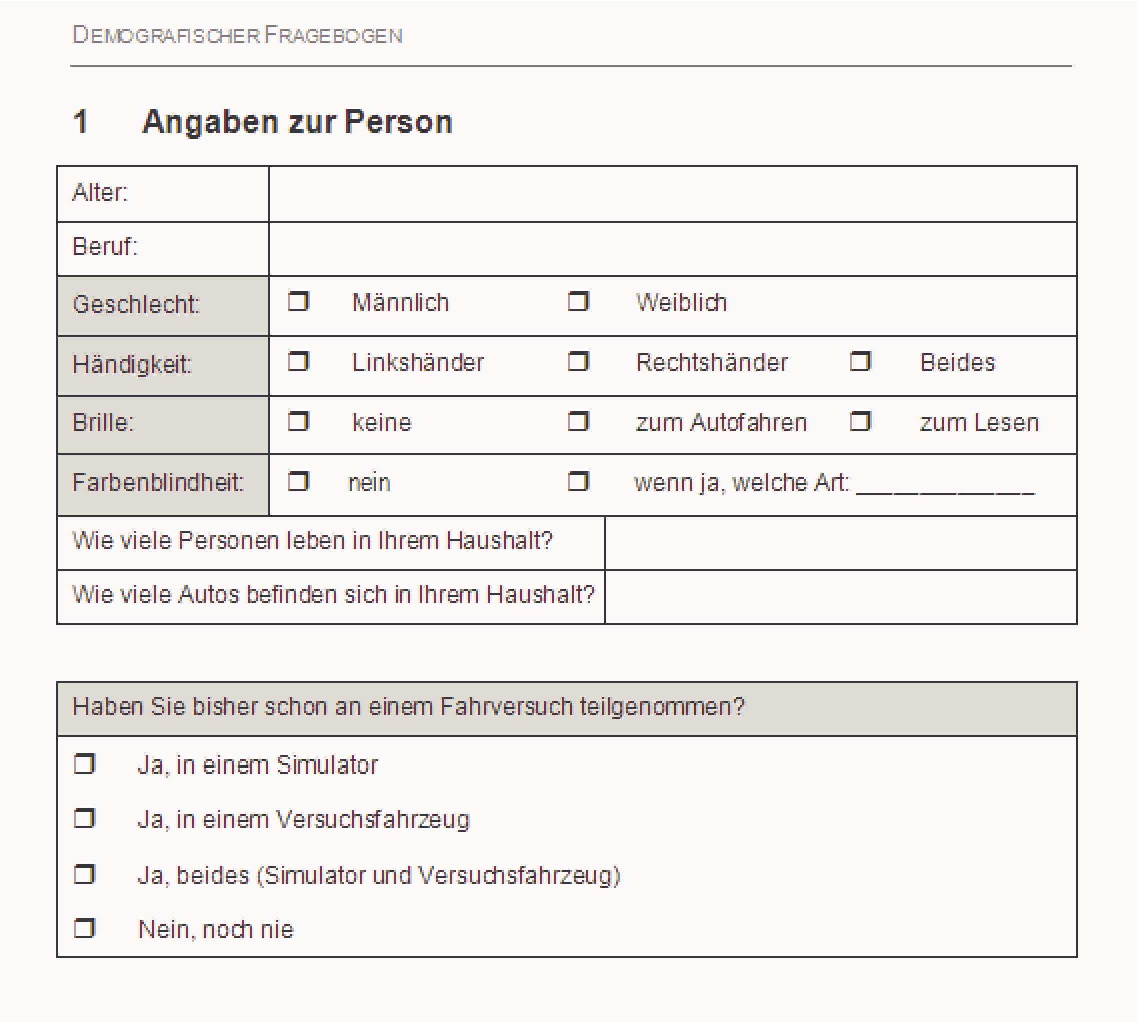
Van der Horst, R., Hogema, J. (1993). *Time-To-Collision and Collision Avoidance Systems*. Paper presented at 6th ICTCT workshop, Salzburg, Austria.

Wierwille, W. W., Gutman, J. (1978). Comparison of primary and secondary task measurements as a function of simulated vehicle dynamics and driving conditions. *Human Factors*, 20, 233-244.

# Appendix A: Geometrical Set-up of the Driving Simulator at LfE, TUM



# Appendix B: Demographic Questionnaire

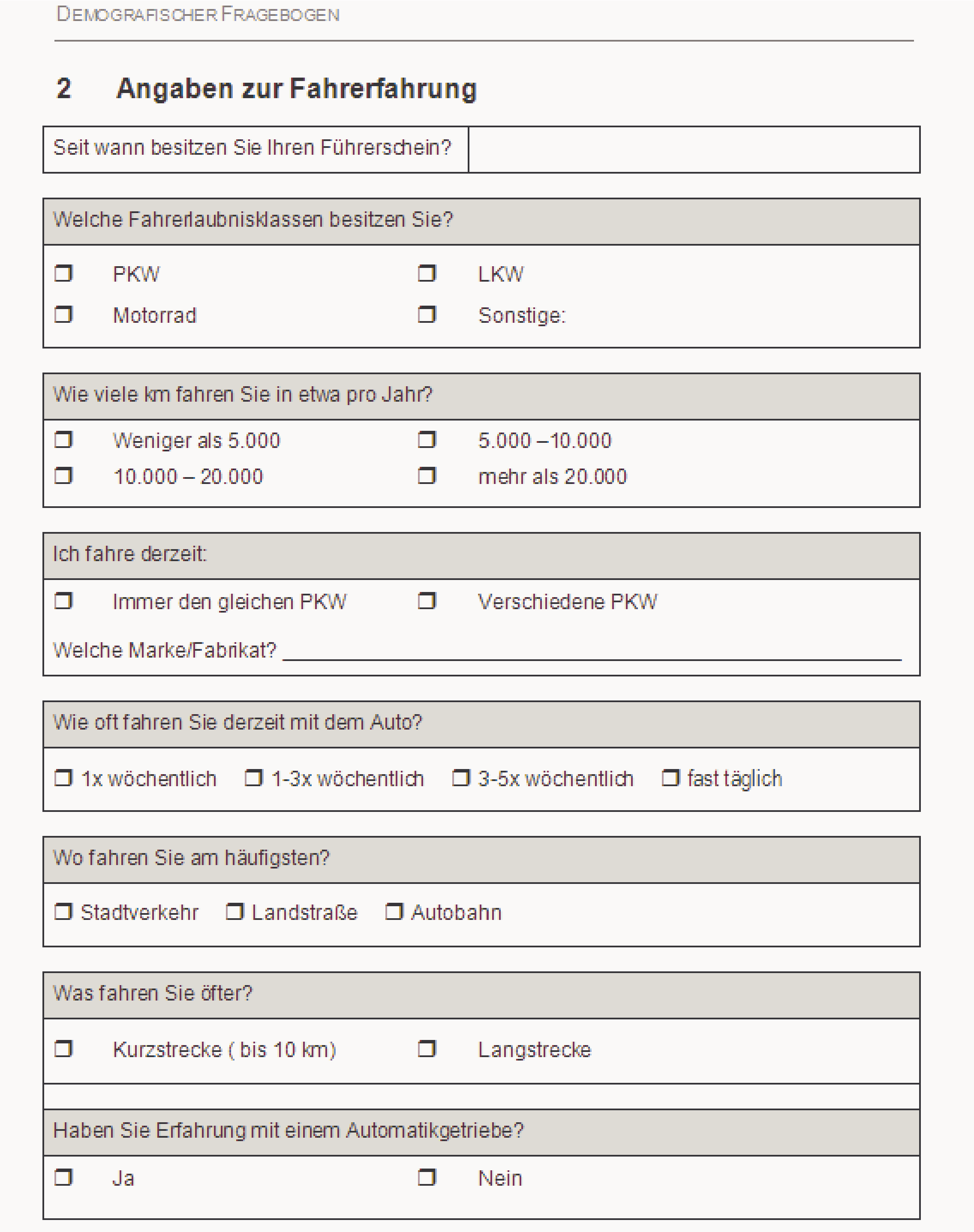


DEMOGRAFISCHER FRAGEBOGEN

## 1 Angaben zur Person

| | | | |
|---|---|---|---|
| Alter: | | | |
| Beruf: | | | |
| Geschlecht: | ☐ Männlich | ☐ Weiblich | |
| Händigkeit: | ☐ Linkshänder | ☐ Rechtshänder | ☐ Beides |
| Brille: | ☐ keine | ☐ zum Autofahren | ☐ zum Lesen |
| Farbenblindheit: | ☐ nein | ☐ wenn ja, welche Art: ____________ | |
| Wie viele Personen leben in Ihrem Haushalt? | | | |
| Wie viele Autos befinden sich in Ihrem Haushalt? | | | |

| Haben Sie bisher schon an einem Fahrversuch teilgenommen? |
|---|
| ☐ Ja, in einem Simulator |
| ☐ Ja, in einem Versuchsfahrzeug |
| ☐ Ja, beides (Simulator und Versuchsfahrzeug) |
| ☐ Nein, noch nie |



DEMOGRAFISCHER FRAGEBOGEN

## 2 Angaben zur Fahrerfahrung

| | |
|---|---|
| Seit wann besitzen Sie Ihren Führerschein? | |

Welche Fahrerlaubnisklassen besitzen Sie?

☐ PKW ☐ LKW

☐ Motorrad ☐ Sonstige:

Wie viele km fahren Sie in etwa pro Jahr?

☐ Weniger als 5.000 ☐ 5.000 –10.000

☐ 10.000 – 20.000 ☐ mehr als 20.000

Ich fahre derzeit:

☐ Immer den gleichen PKW ☐ Verschiedene PKW

Welche Marke/Fabrikat? ______________________________

Wie oft fahren Sie derzeit mit dem Auto?

☐ 1x wöchentlich ☐ 1-3x wöchentlich ☐ 3-5x wöchentlich ☐ fast täglich

Wo fahren Sie am häufigsten?

☐ Stadtverkehr ☐ Landstraße ☐ Autobahn

Was fahren Sie öfter?

☐ Kurzstrecke ( bis 10 km) ☐ Langstrecke

Haben Sie Erfahrung mit einem Automatikgetriebe?

☐ Ja ☐ Nein

## 3 Angaben zur Fahrertypeinschätzung

| Im Vergleich zu anderen Autofahrern würde ich mich folgendermaßen einschätzen: | | |
|---|---|---|
| Sehr erfahren | ☐ ☐ ☐ ☐ ☐ ☐ | Sehr unerfahren |

| Ich würde meinen Fahrstil beschreiben als: | | |
|---|---|---|
| Sportlich/dynamisch | ☐ ☐ ☐ ☐ ☐ ☐ | Ruhig/ausgeglichen |

| Bei hoher Verkehrsdichte verhalte ich mich: | | |
|---|---|---|
| Eher offensiv | ☐ ☐ ☐ ☐ ☐ ☐ | Eher defensiv |

| Meine Kontrolle über das Fahrzeug schätze ich wie folgt ein: | | |
|---|---|---|
| Ich beherrsche mein Fahrzeug in jeder Situation | ☐ ☐ ☐ ☐ ☐ ☐ | In manchen Situationen habe ich Schwierigkeiten |

| Mit einem fremden Fahrzeug zurechtzukommen … | | |
|---|---|---|
| fällt mir leicht | ☐ ☐ ☐ ☐ ☐ ☐ | bereitet mir Schwierigkeiten |

| Ich bin an technischen Dingen um das Auto … | | |
|---|---|---|
| sehr interessiert und informiert | ☐ ☐ ☐ ☐ ☐ ☐ | nicht interessiert |

| Autofahren bedeutet für mich: | | |
|---|---|---|
| Spaß | ☐ ☐ ☐ ☐ ☐ ☐ | ein notwendiges Übel |

| Beim Autofahren … | | |
|---|---|---|
| bleibe ich meistens entspannt und locker | ☐ ☐ ☐ ☐ ☐ ☐ | fühle ich mich oft gestresst |

DEMOGRAFISCHER FRAGEBOGEN

Im Alltag achte ich auf eine Sprit sparende Fahrweise ...

überhaupt nicht ☐ ☐ ☐ ☐ ☐ ☐ sehr stark

Geschwindigkeitsbeschränkungen halte ich strikt ein:

überhaupt nicht ☐ ☐ ☐ ☐ ☐ ☐ immer

Ich kann mich für Technik ... begeistern

überhaupt nicht ☐ ☐ ☐ ☐ ☐ ☐ immer

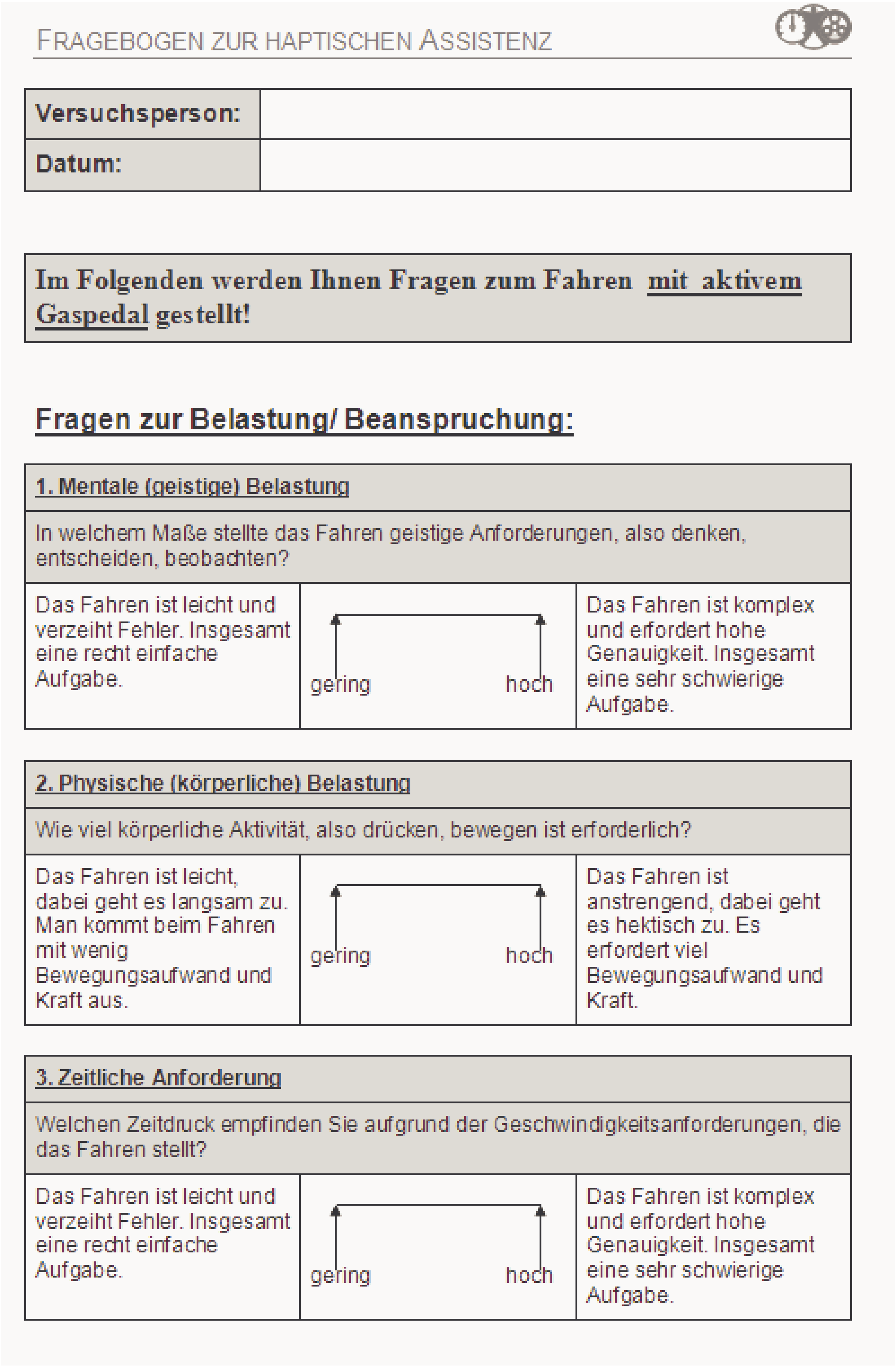
Ich probiere gerne neue technische Geräte aus

trifft zu ☐ ☐ ☐ ☐ ☐ ☐ trifft nicht zu

Ich spiele Computerspiele ...

überhaupt nicht ☐ ☐ ☐ ☐ ☐ ☐ sehr oft

# Appendix C: Questionnaire given after each Drive



FRAGEBOGEN ZUR HAPTISCHEN ASSISTENZ

| | |
|---|---|
| **Versuchsperson:** | |
| **Datum:** | |

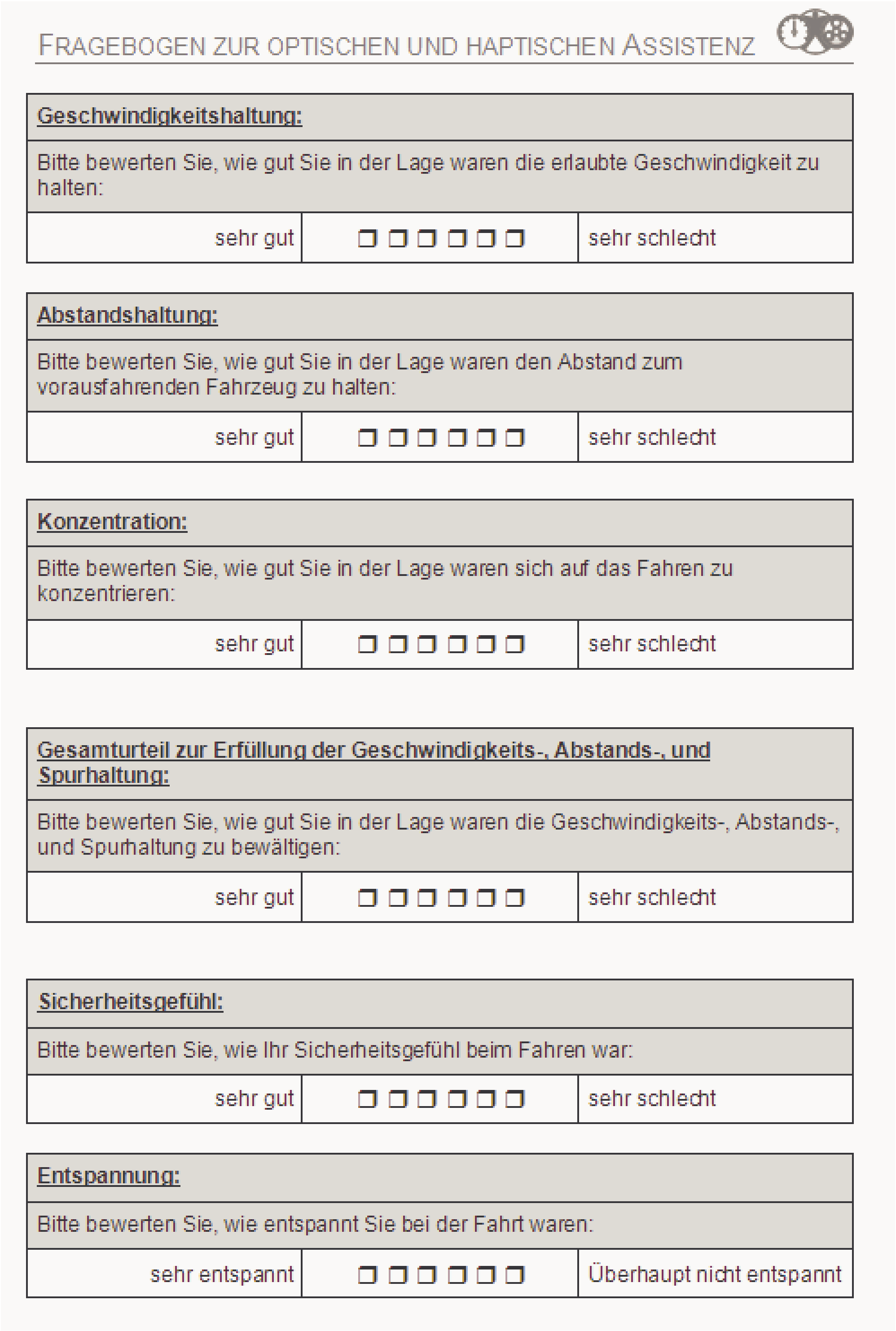
**Im Folgenden werden Ihnen Fragen zum Fahren mit aktivem Gaspedal gestellt!**

## Fragen zur Belastung/ Beanspruchung:

| | | |
|---|---|---|
| **1. Mentale (geistige) Belastung** | | |
| In welchem Maße stellte das Fahren geistige Anforderungen, also denken, entscheiden, beobachten? | | |
| Das Fahren ist leicht und verzeiht Fehler. Insgesamt eine recht einfache Aufgabe. | gering hoch | Das Fahren ist komplex und erfordert hohe Genauigkeit. Insgesamt eine sehr schwierige Aufgabe. |

| | | |
|---|---|---|
| **2. Physische (körperliche) Belastung** | | |
| Wie viel körperliche Aktivität, also drücken, bewegen ist erforderlich? | | |
| Das Fahren ist leicht, dabei geht es langsam zu. Man kommt beim Fahren mit wenig Bewegungsaufwand und Kraft aus. | gering hoch | Das Fahren ist anstrengend, dabei geht es hektisch zu. Es erfordert viel Bewegungsaufwand und Kraft. |

| | | |
|---|---|---|
| **3. Zeitliche Anforderung** | | |
| Welchen Zeitdruck empfinden Sie aufgrund der Geschwindigkeitsanforderungen, die das Fahren stellt? | | |
| Das Fahren ist leicht und verzeiht Fehler. Insgesamt eine recht einfache Aufgabe. | gering hoch | Das Fahren ist komplex und erfordert hohe Genauigkeit. Insgesamt eine sehr schwierige Aufgabe. |



## FRAGEBOGEN ZUR OPTISCHEN UND HAPTISCHEN ASSISTENZ

**Geschwindigkeitshaltung:**

Bitte bewerten Sie, wie gut Sie in der Lage waren die erlaubte Geschwindigkeit zu halten:

| | | |
|---|---|---|
| sehr gut | ☐ ☐ ☐ ☐ ☐ ☐ | sehr schlecht |

**Abstandshaltung:**

Bitte bewerten Sie, wie gut Sie in der Lage waren den Abstand zum vorausfahrenden Fahrzeug zu halten:

| | | |
|---|---|---|
| sehr gut | ☐ ☐ ☐ ☐ ☐ ☐ | sehr schlecht |

**Konzentration:**

Bitte bewerten Sie, wie gut Sie in der Lage waren sich auf das Fahren zu konzentrieren:

| | | |
|---|---|---|
| sehr gut | ☐ ☐ ☐ ☐ ☐ ☐ | sehr schlecht |

**Gesamturteil zur Erfüllung der Geschwindigkeits-, Abstands-, und Spurhaltung:**

Bitte bewerten Sie, wie gut Sie in der Lage waren die Geschwindigkeits-, Abstands-, und Spurhaltung zu bewältigen:

| | | |
|---|---|---|
| sehr gut | ☐ ☐ ☐ ☐ ☐ ☐ | sehr schlecht |

**Sicherheitsgefühl:**

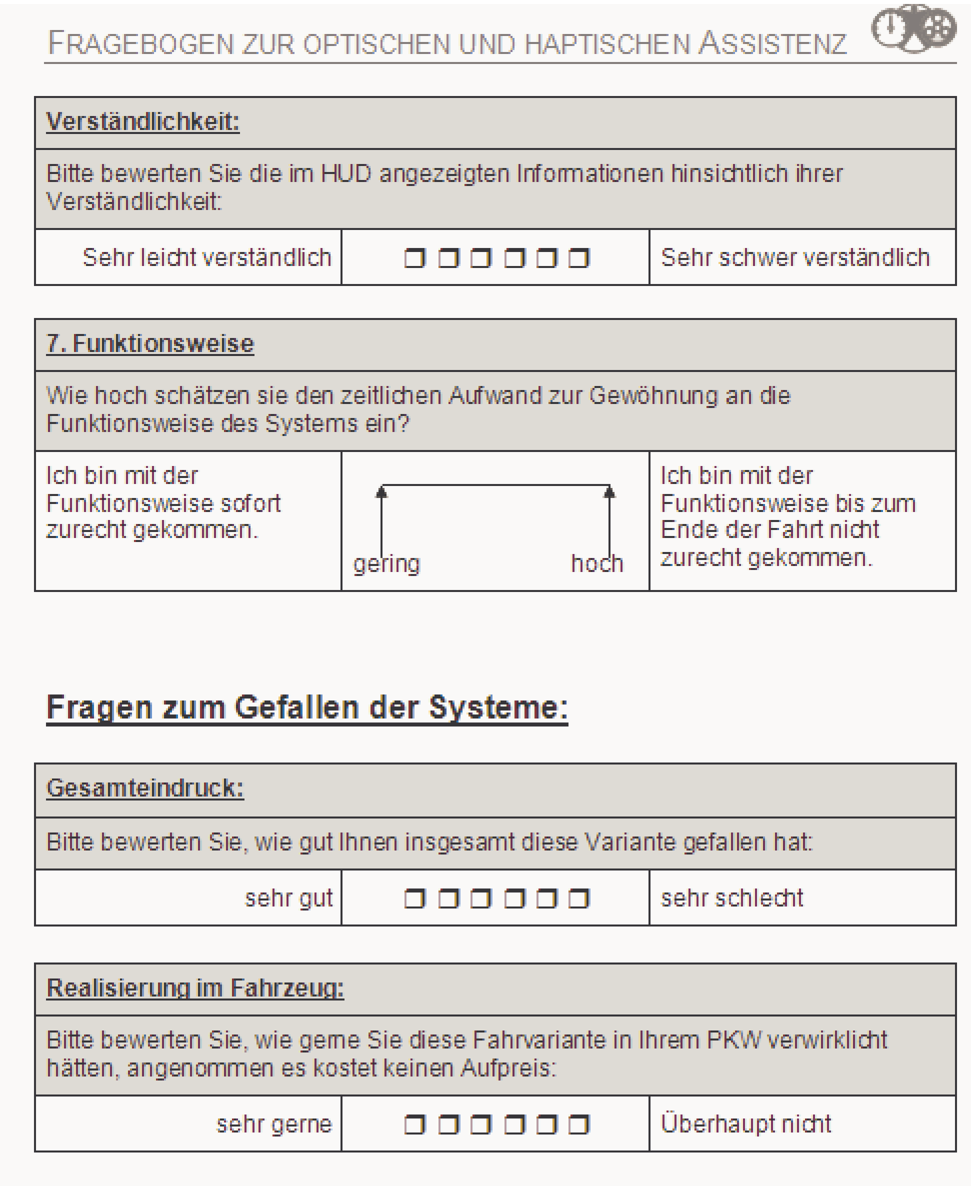
Bitte bewerten Sie, wie Ihr Sicherheitsgefühl beim Fahren war:

| | | |
|---|---|---|
| sehr gut | ☐ ☐ ☐ ☐ ☐ ☐ | sehr schlecht |

**Entspannung:**

Bitte bewerten Sie, wie entspannt Sie bei der Fahrt waren:

| | | |
|---|---|---|
| sehr entspannt | ☐ ☐ ☐ ☐ ☐ ☐ | Überhaupt nicht entspannt |



## FRAGEBOGEN ZUR OPTISCHEN UND HAPTISCHEN ASSISTENZ

| | | |
|---|---|---|
| **Verständlichkeit:** | | |
| Bitte bewerten Sie die im HUD angezeigten Informationen hinsichtlich ihrer Verständlichkeit: | | |
| Sehr leicht verständlich | ☐ ☐ ☐ ☐ ☐ ☐ | Sehr schwer verständlich |

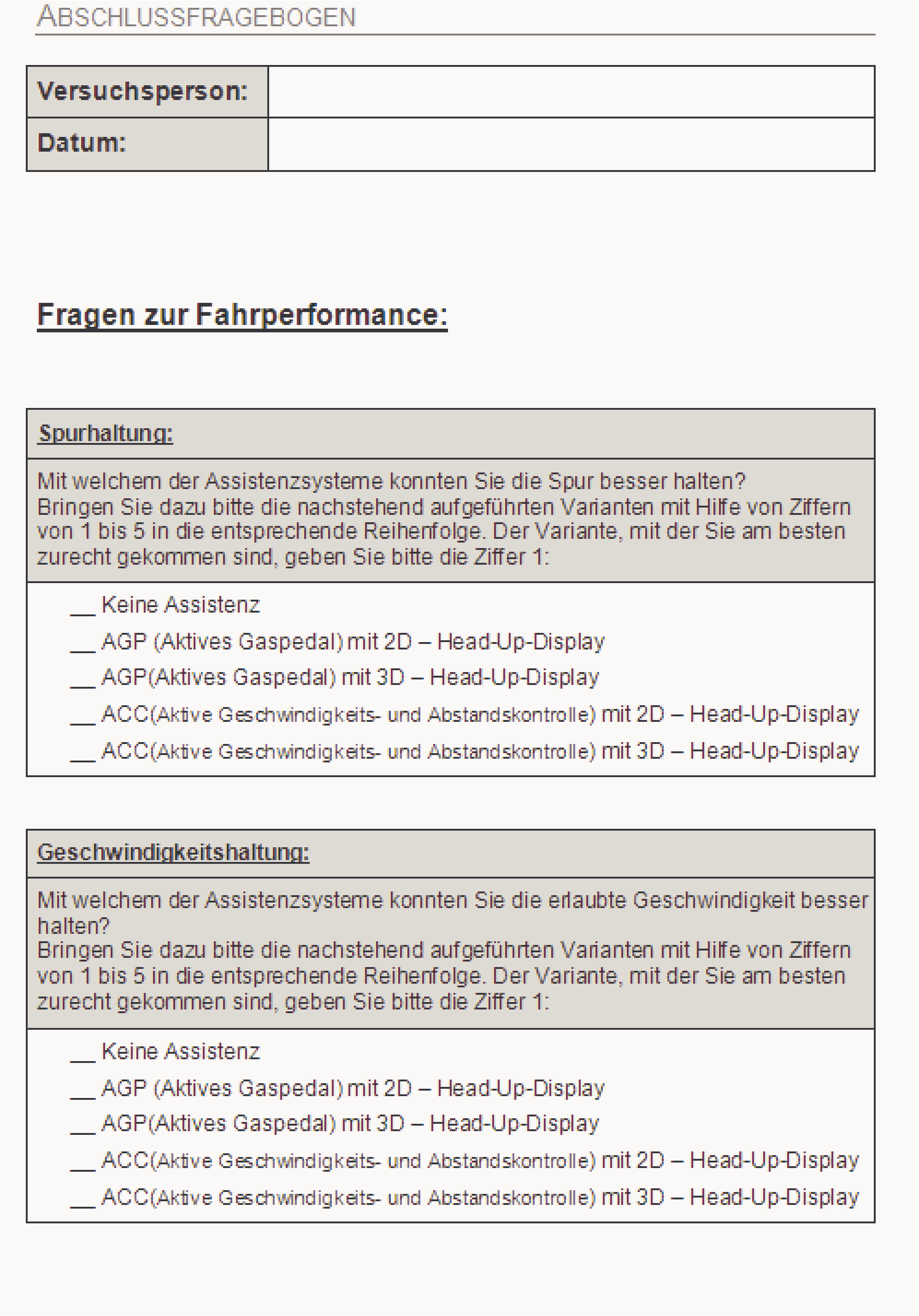
| | | |
|---|---|---|
| **7. Funktionsweise** | | |
| Wie hoch schätzen sie den zeitlichen Aufwand zur Gewöhnung an die Funktionsweise des Systems ein? | | |
| Ich bin mit der Funktionsweise sofort zurecht gekommen. | gering hoch | Ich bin mit der Funktionsweise bis zum Ende der Fahrt nicht zurecht gekommen. |

## Fragen zum Gefallen der Systeme:

| | | |
|---|---|---|
| **Gesamteindruck:** | | |
| Bitte bewerten Sie, wie gut Ihnen insgesamt diese Variante gefallen hat: | | |
| sehr gut | ☐ ☐ ☐ ☐ ☐ ☐ | sehr schlecht |

| | | |
|---|---|---|
| **Realisierung im Fahrzeug:** | | |
| Bitte bewerten Sie, wie gerne Sie diese Fahrvariante in Ihrem PKW verwirklicht hätten, angenommen es kostet keinen Aufpreis: | | |
| sehr gerne | ☐ ☐ ☐ ☐ ☐ ☐ | Überhaupt nicht |

# Appendix D: Final Questionnaire



ABSCHLUSSFRAGEBOGEN

| | |
|---|---|
| **Versuchsperson:** | |
| **Datum:** | |

**Fragen zur Fahrperformance:**

**Spurhaltung:**

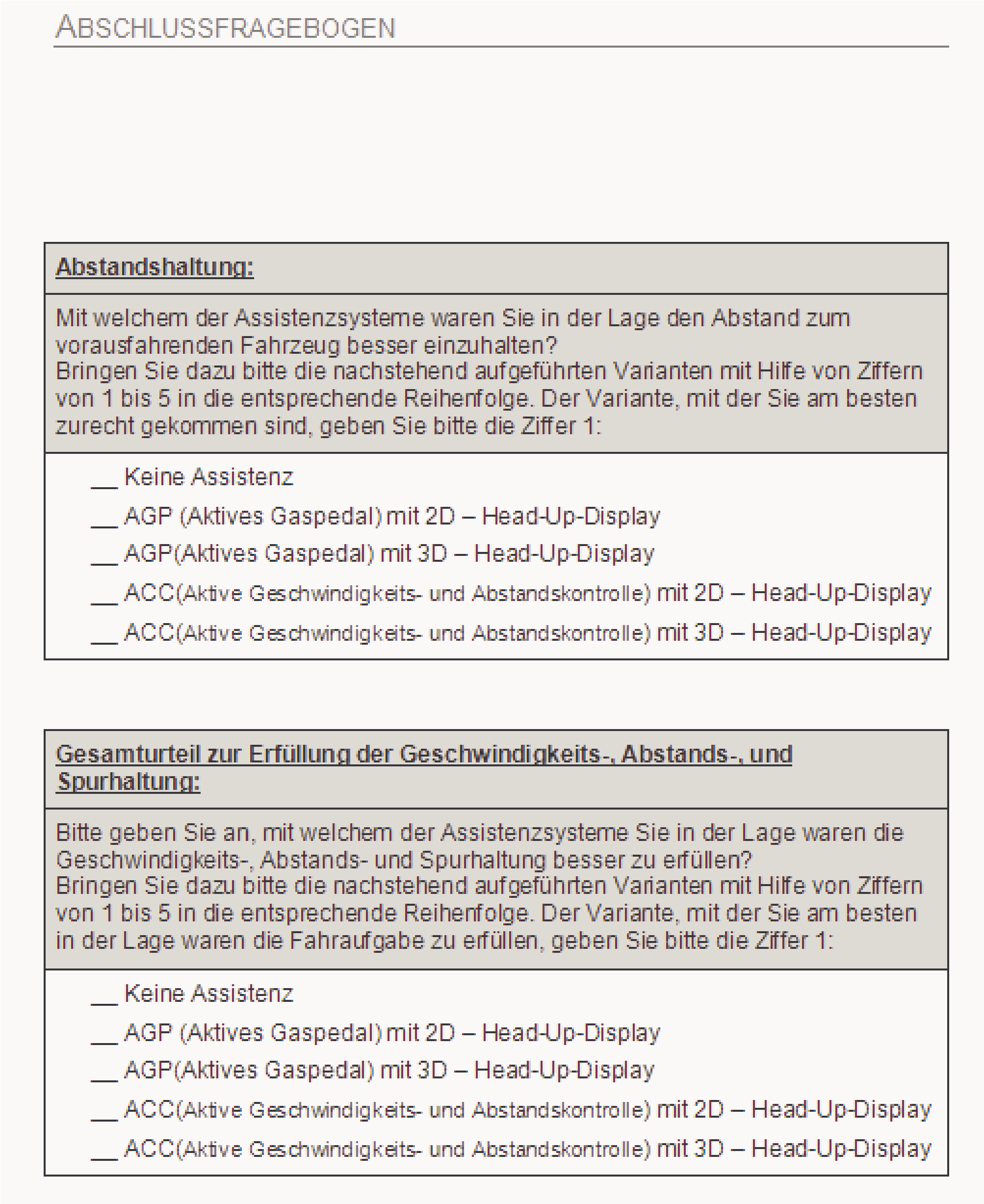
Mit welchem der Assistenzsysteme konnten Sie die Spur besser halten?
Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der Sie am besten zurecht gekommen sind, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display

**Geschwindigkeitshaltung:**

Mit welchem der Assistenzsysteme konnten Sie die erlaubte Geschwindigkeit besser halten?
Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der Sie am besten zurecht gekommen sind, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display



## ABSCHLUSSFRAGEBOGEN

**Abstandshaltung:**

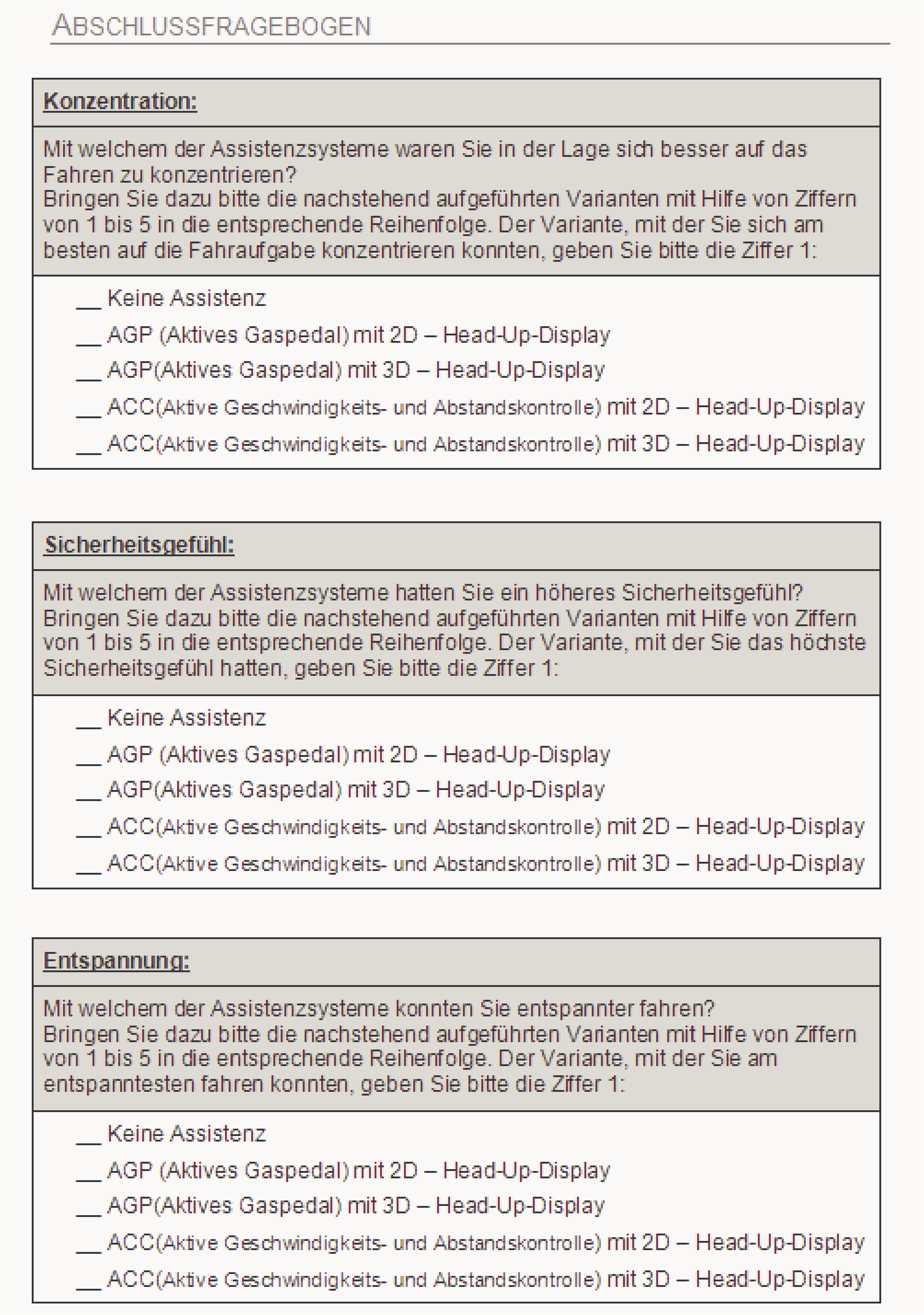
Mit welchem der Assistenzsysteme waren Sie in der Lage den Abstand zum vorausfahrenden Fahrzeug besser einzuhalten?
Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der Sie am besten zurecht gekommen sind, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display

**Gesamturteil zur Erfüllung der Geschwindigkeits-, Abstands-, und Spurhaltung:**

Bitte geben Sie an, mit welchem der Assistenzsysteme Sie in der Lage waren die Geschwindigkeits-, Abstands- und Spurhaltung besser zu erfüllen?
Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der Sie am besten in der Lage waren die Fahraufgabe zu erfüllen, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display



## ABSCHLUSSFRAGEBOGEN

**Konzentration:**

Mit welchem der Assistenzsysteme waren Sie in der Lage sich besser auf das Fahren zu konzentrieren?
Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der Sie sich am besten auf die Fahraufgabe konzentrieren konnten, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display

**Sicherheitsgefühl:**

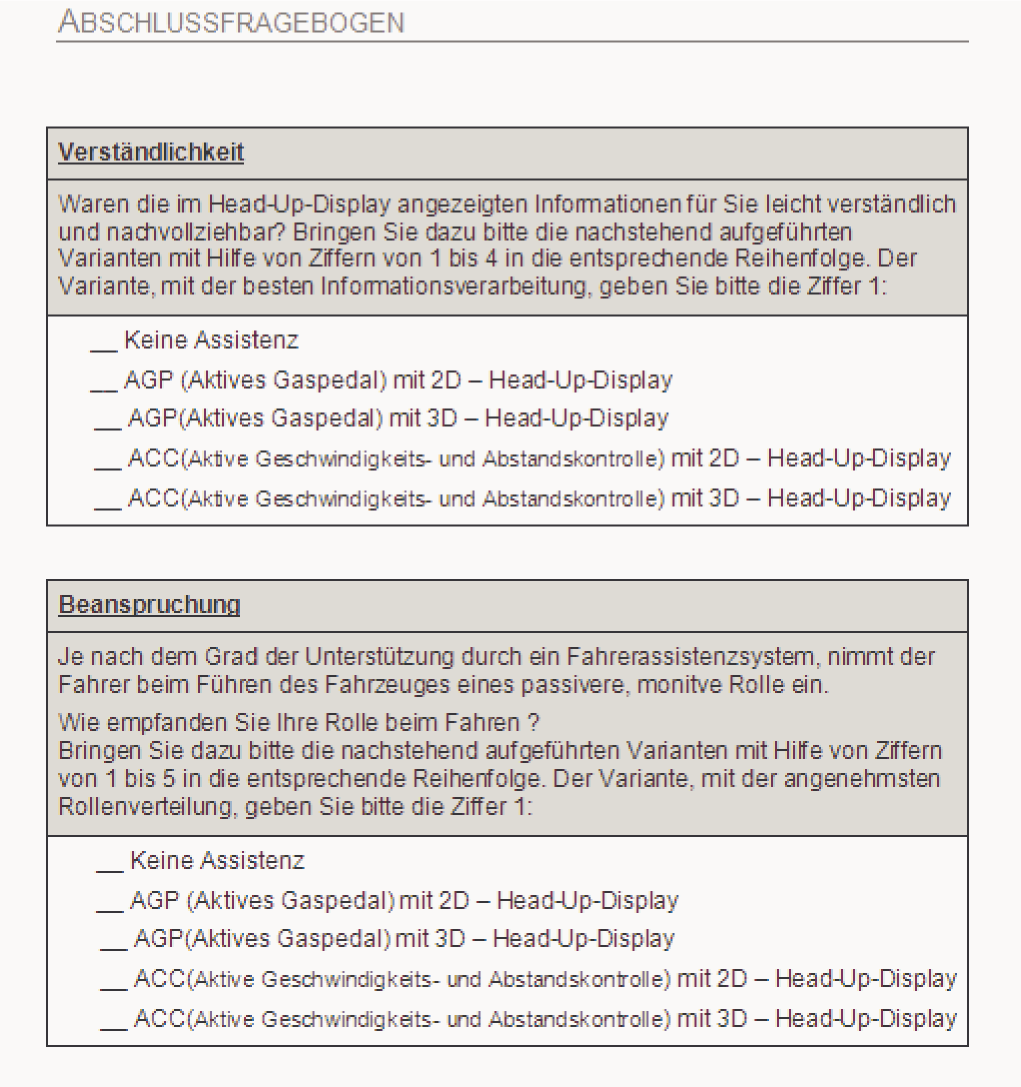
Mit welchem der Assistenzsysteme hatten Sie ein höheres Sicherheitsgefühl?
Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der Sie das höchste Sicherheitsgefühl hatten, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display

**Entspannung:**

Mit welchem der Assistenzsysteme konnten Sie entspannter fahren?
Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der Sie am entspanntesten fahren konnten, geben Sie bitte die Ziffer 1:

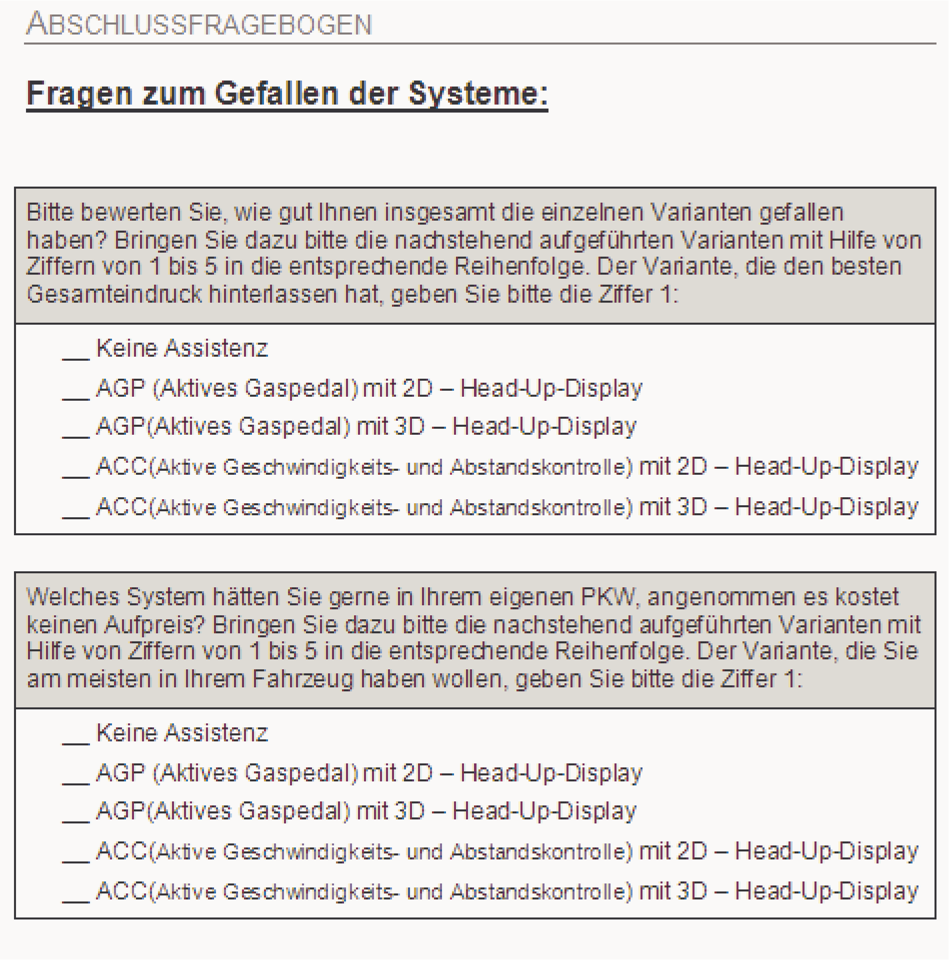
__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display



## ABSCHLUSSFRAGEBOGEN

| |
|---|
| **Verständlichkeit** |
| Waren die im Head-Up-Display angezeigten Informationen für Sie leicht verständlich und nachvollziehbar? Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 4 in die entsprechende Reihenfolge. Der Variante, mit der besten Informationsverarbeitung, geben Sie bitte die Ziffer 1: |
| __ Keine Assistenz<br>__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display<br>__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display<br>__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display<br>__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display |

| |
|---|
| **Beanspruchung** |
| Je nach dem Grad der Unterstützung durch ein Fahrerassistenzsystem, nimmt der Fahrer beim Führen des Fahrzeuges eines passivere, monitve Rolle ein.<br>Wie empfanden Sie Ihre Rolle beim Fahren ?<br>Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, mit der angenehmsten Rollenverteilung, geben Sie bitte die Ziffer 1: |
| __ Keine Assistenz<br>__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display<br>__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display<br>__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display<br>__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display |



ABSCHLUSSFRAGEBOGEN

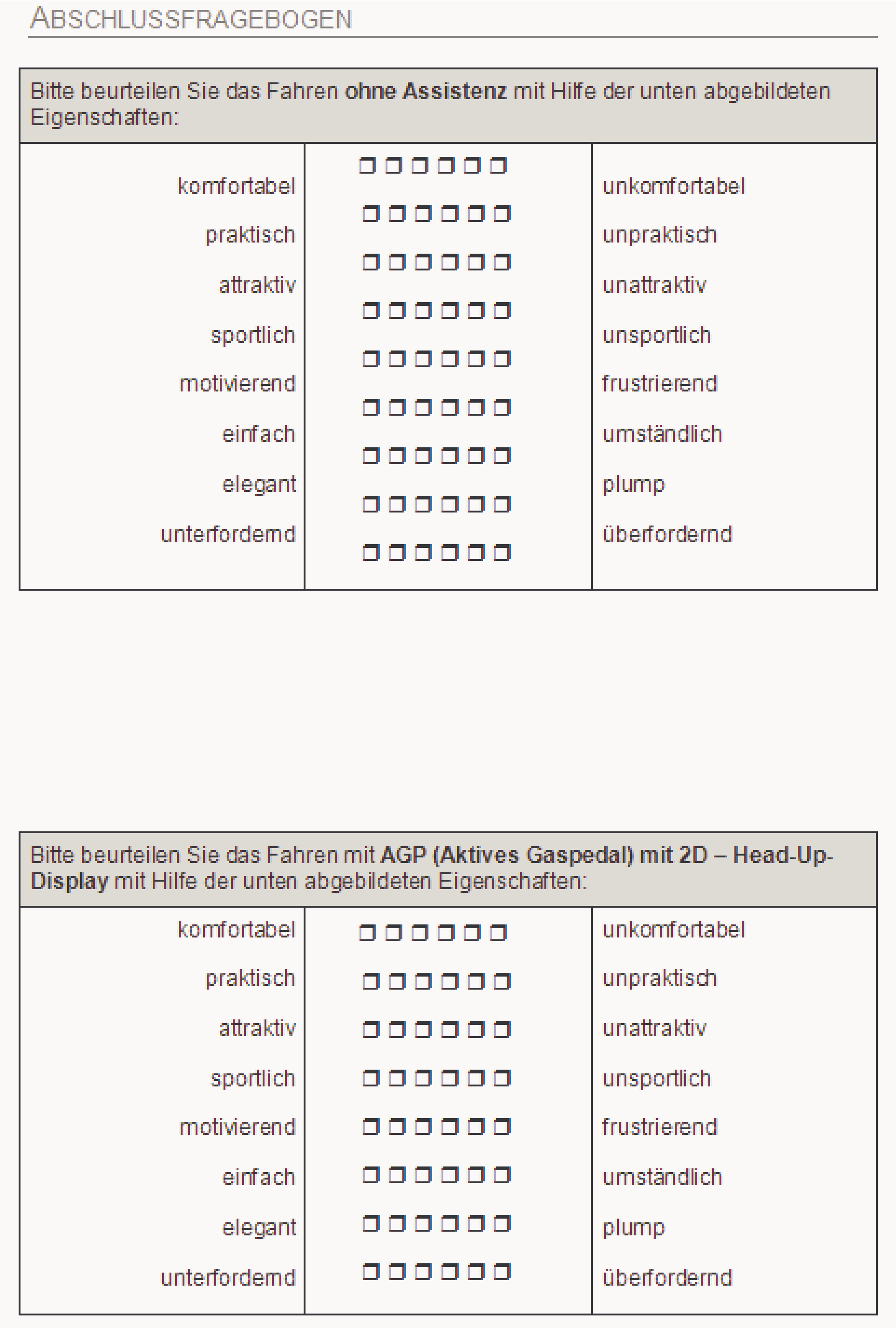
## Fragen zum Gefallen der Systeme:

Bitte bewerten Sie, wie gut Ihnen insgesamt die einzelnen Varianten gefallen haben? Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, die den besten Gesamteindruck hinterlassen hat, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display

Welches System hätten Sie gerne in Ihrem eigenen PKW, angenommen es kostet keinen Aufpreis? Bringen Sie dazu bitte die nachstehend aufgeführten Varianten mit Hilfe von Ziffern von 1 bis 5 in die entsprechende Reihenfolge. Der Variante, die Sie am meisten in Ihrem Fahrzeug haben wollen, geben Sie bitte die Ziffer 1:

__ Keine Assistenz
__ AGP (Aktives Gaspedal) mit 2D – Head-Up-Display
__ AGP(Aktives Gaspedal) mit 3D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display
__ ACC(Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display



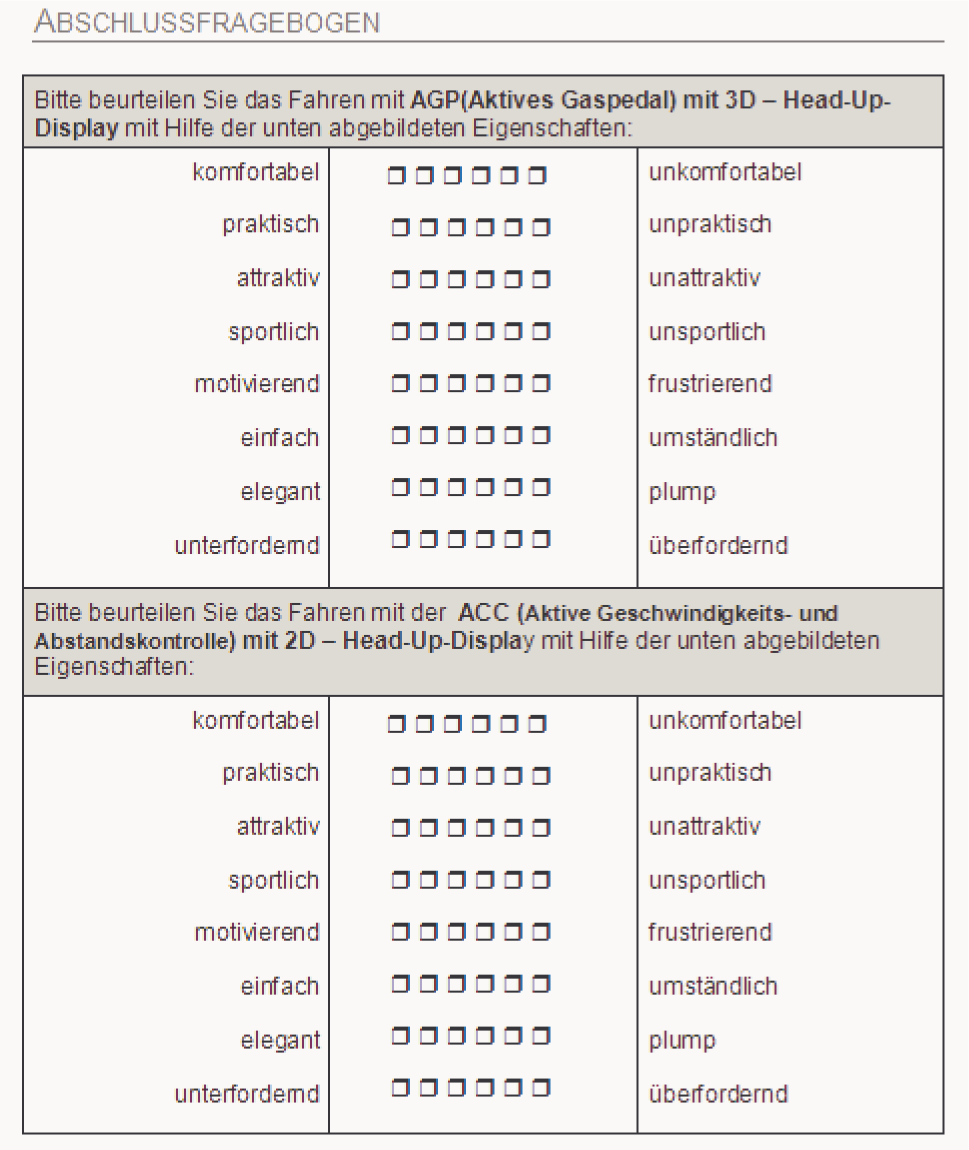
## ABSCHLUSSFRAGEBOGEN

Bitte beurteilen Sie das Fahren **ohne Assistenz** mit Hilfe der unten abgebildeten Eigenschaften:

| | | |
|---|---|---|
| komfortabel | ☐ ☐ ☐ ☐ ☐ ☐ | unkomfortabel |
| praktisch | ☐ ☐ ☐ ☐ ☐ ☐ | unpraktisch |
| attraktiv | ☐ ☐ ☐ ☐ ☐ ☐ | unattraktiv |
| sportlich | ☐ ☐ ☐ ☐ ☐ ☐ | unsportlich |
| motivierend | ☐ ☐ ☐ ☐ ☐ ☐ | frustrierend |
| einfach | ☐ ☐ ☐ ☐ ☐ ☐ | umständlich |
| elegant | ☐ ☐ ☐ ☐ ☐ ☐ | plump |
| unterfordernd | ☐ ☐ ☐ ☐ ☐ ☐ | überfordernd |
| | ☐ ☐ ☐ ☐ ☐ ☐ | |

Bitte beurteilen Sie das Fahren mit **AGP (Aktives Gaspedal) mit 2D – Head-Up-Display** mit Hilfe der unten abgebildeten Eigenschaften:

| | | |
|---|---|---|
| komfortabel | ☐ ☐ ☐ ☐ ☐ ☐ | unkomfortabel |
| praktisch | ☐ ☐ ☐ ☐ ☐ ☐ | unpraktisch |
| attraktiv | ☐ ☐ ☐ ☐ ☐ ☐ | unattraktiv |
| sportlich | ☐ ☐ ☐ ☐ ☐ ☐ | unsportlich |
| motivierend | ☐ ☐ ☐ ☐ ☐ ☐ | frustrierend |
| einfach | ☐ ☐ ☐ ☐ ☐ ☐ | umständlich |
| elegant | ☐ ☐ ☐ ☐ ☐ ☐ | plump |
| unterfordernd | ☐ ☐ ☐ ☐ ☐ ☐ | überfordernd |



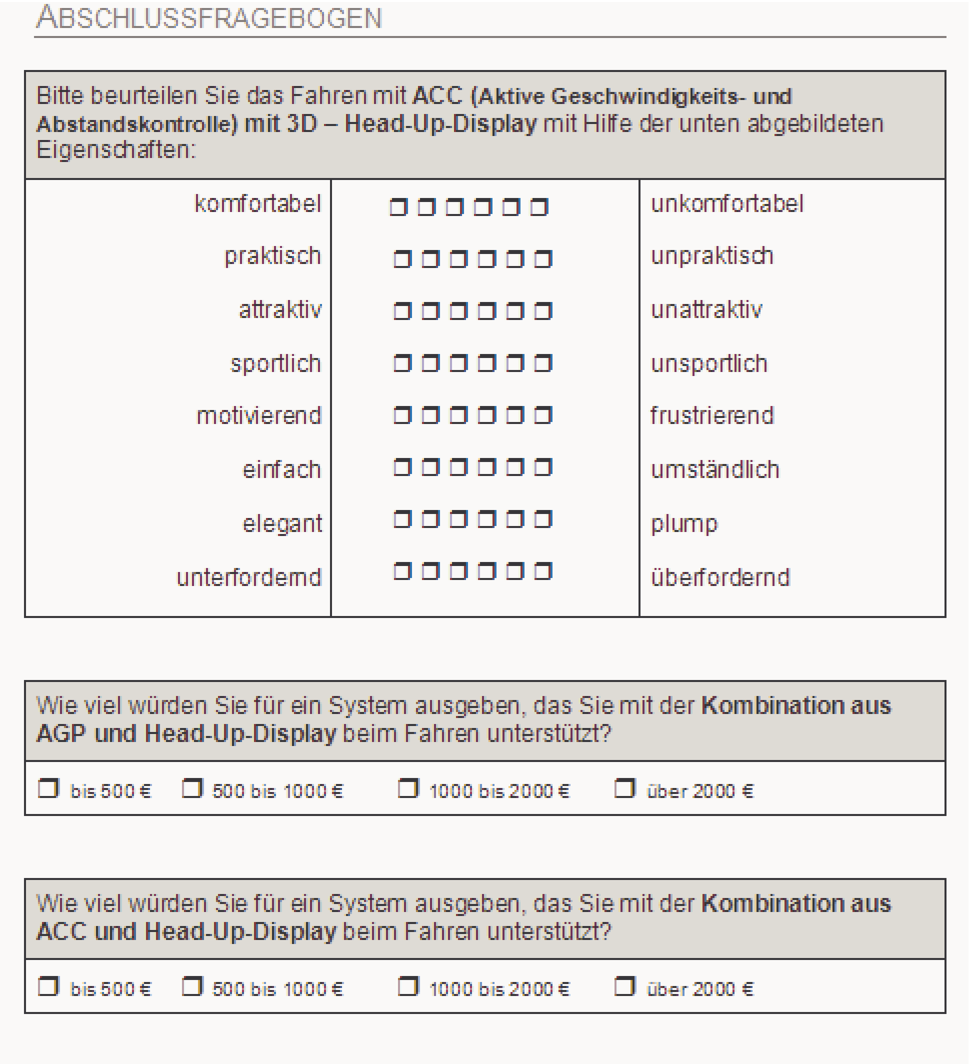
## ABSCHLUSSFRAGEBOGEN

Bitte beurteilen Sie das Fahren mit **AGP(Aktives Gaspedal) mit 3D – Head-Up-Display** mit Hilfe der unten abgebildeten Eigenschaften:

| | | |
|---|---|---|
| komfortabel | ☐ ☐ ☐ ☐ ☐ ☐ | unkomfortabel |
| praktisch | ☐ ☐ ☐ ☐ ☐ ☐ | unpraktisch |
| attraktiv | ☐ ☐ ☐ ☐ ☐ ☐ | unattraktiv |
| sportlich | ☐ ☐ ☐ ☐ ☐ ☐ | unsportlich |
| motivierend | ☐ ☐ ☐ ☐ ☐ ☐ | frustrierend |
| einfach | ☐ ☐ ☐ ☐ ☐ ☐ | umständlich |
| elegant | ☐ ☐ ☐ ☐ ☐ ☐ | plump |
| unterfordernd | ☐ ☐ ☐ ☐ ☐ ☐ | überfordernd |

Bitte beurteilen Sie das Fahren mit der **ACC (Aktive Geschwindigkeits- und Abstandskontrolle) mit 2D – Head-Up-Display** mit Hilfe der unten abgebildeten Eigenschaften:

| | | |
|---|---|---|
| komfortabel | ☐ ☐ ☐ ☐ ☐ ☐ | unkomfortabel |
| praktisch | ☐ ☐ ☐ ☐ ☐ ☐ | unpraktisch |
| attraktiv | ☐ ☐ ☐ ☐ ☐ ☐ | unattraktiv |
| sportlich | ☐ ☐ ☐ ☐ ☐ ☐ | unsportlich |
| motivierend | ☐ ☐ ☐ ☐ ☐ ☐ | frustrierend |
| einfach | ☐ ☐ ☐ ☐ ☐ ☐ | umständlich |
| elegant | ☐ ☐ ☐ ☐ ☐ ☐ | plump |
| unterfordernd | ☐ ☐ ☐ ☐ ☐ ☐ | überfordernd |



## ABSCHLUSSFRAGEBOGEN

Bitte beurteilen Sie das Fahren mit **ACC (Aktive Geschwindigkeits- und Abstandskontrolle) mit 3D – Head-Up-Display** mit Hilfe der unten abgebildeten Eigenschaften:

| | | |
|---|---|---|
| komfortabel | ☐ ☐ ☐ ☐ ☐ ☐ | unkomfortabel |
| praktisch | ☐ ☐ ☐ ☐ ☐ ☐ | unpraktisch |
| attraktiv | ☐ ☐ ☐ ☐ ☐ ☐ | unattraktiv |
| sportlich | ☐ ☐ ☐ ☐ ☐ ☐ | unsportlich |
| motivierend | ☐ ☐ ☐ ☐ ☐ ☐ | frustrierend |
| einfach | ☐ ☐ ☐ ☐ ☐ ☐ | umständlich |
| elegant | ☐ ☐ ☐ ☐ ☐ ☐ | plump |
| unterfordernd | ☐ ☐ ☐ ☐ ☐ ☐ | überfordernd |

Wie viel würden Sie für ein System ausgeben, das Sie mit der **Kombination aus AGP und Head-Up-Display** beim Fahren unterstützt?

☐ bis 500 € ☐ 500 bis 1000 € ☐ 1000 bis 2000 € ☐ über 2000 €

Wie viel würden Sie für ein System ausgeben, das Sie mit der **Kombination aus ACC und Head-Up-Display** beim Fahren unterstützt?

☐ bis 500 € ☐ 500 bis 1000 € ☐ 1000 bis 2000 € ☐ über 2000 €