Highly automated vehicles for intelligent transport

7th Framework programme
ICT-2007.6.1
ICT for intelligent vehicles and mobility services
Grant agreement no.: 212154

The future of driving.

Deliverable D61.1
Final Report

Version number	Version 1.0
Dissemination level	PU
Lead contractor	Continental Automotive GmbH
Due date	30.09.2011
Date of preparation	23.09.2011

List of beneficiaries

Beneficiary Number	Beneficiary name	Beneficiary short name	Country	Date enter project	Date exit project
1 (coord.)	Continental Automotive GmbH a) Coordination (CAGCO) b) Architecture and Copilot (CAGAC) c) Chassis and Safety Controller (fail-silent generic ECU), (CAGCS) d) Automated assistance in roadworks and congestion (CTAAA) [1] e) Front sensing system (ADC)[2]	CAG	Germany	1	42
3	Volvo Technology AB a) Automated queue assistance (VTEC) b) Active green driving (VTEC) c) Brake-by-Wire truck (V3P)[3]	VTEC	Sweden	1	42
4	Volkswagen AG	VW	Germany	1	42
5	EFKON AG	EFKON	Austria	1	42
6	Ibeo Automobile Sensor GmbH	Ibeo	Germany	1	23
7	Irion Management Consulting GmbH	IMC	Germany	1	13
10	Knowllence	KN	France	1	42
11	Continental Automotive S.A.S.	CAF	France	1	42
12	Deutsches Zentrum fuer Luft- und Raumfahrt e.V.	DLR	Germany	1	42
13	Ecole Polytechnique Fédérale de Lausanne - Laboratoire de Production Microtechnique	EPFL	Switzerland	1	42
14	Institute of Communication and Computer Systems	ICCS	Greece	1	42
15	University of Applied Sciences Amberg-Weiden (HAW AM-WEN)	UAM	Germany	1	42
16	Budapest University of Technology and Economics	BME	Hungary	1	42
17	Universität Stuttgart, Institut für Luftfahrtsysteme	USTUTT	Germany	1	42
18	Wuerzburg Institute of Traffic Sciences GmbH	WIVW	Germany	1	42

[1] Automated Assistance in Roadworks and Congestion (ADAS and electric steering) was in responsibility of Continental Teves AG & Co. oHG, Frankfurt, who was included as a third party assigned to beneficiary no. 1.
[2] The front sensing system part was in responsibility of A.D.C. GmbH, Lindau, who was included as third party assigned to beneficiary no. 1.
[3] The EMB activities were partly in responsibility of Volvo 3P. Volvo 3P is organisationally connected to Volvo Lastvagnar AB. Volvo Lastvagnar AB was included as Third Party assigned to partner no. 3, VTEC.

19	Institut National de Recherche en Informatique et en Automatique[4]	INRIA	France	1	42
20	Laboratoire Central des Ponts et Chaussees	LCPC	France	1	35
21	Allemann Technologies Sarl	ATS	Switzerland	1	1
22	Siemens VDO Automotive AG	SVD	Germany	1	5
23	Haldex Brake Products AB	HLX	Sweden	10	42
24	Explinovo GmbH	EXP	Germany	2	42
25	Sick AG	SICK	Germany	24	42
26	Institut français des sciences et technologies des transports, de l'aménagement et des réseaux	IFSTTAR	France	36	42

[4] In the context of the joint research institute "La Route Automatisee" two third parties were assigned to partner INRIA: Ecole Nationale Superieure des Mines de Paris (ENSMP) and Association pour la Recherche et le Developpement des Methodes et des Processus Industriels (ARMINES).

Authors

Reiner Hoeger, Continental Automotive
Holger Zeng, Continental Automotive
Alfred Hoess, Continental Automotive
Thomas Kranz, Continental Automotive
Serge Boverie, Continental Automotive
Matthias Strauss, Continental Teves
Erika Jakobsson, Volvo Technology
Achim Beutner, Volvo Technology
Arne Bartels, Volkswagen
Thanh-Binh To, Volkswagen
Hannes Stratil, EFKON
Kay Fürstenberg, SICK
Florian Ahlers, SICK
Estelle Frey, Knowllence
Anna Schieben, DLR
Henning Mosebach, DLR
Frank Flemisch, DLR
Alain Dufaux, EPFL
Davide Manetti, EPFL
Angelos Amditis, ICCS
Irene Mantzouranis, ICCS
Heike Lepke, UAM
Zsolt Szalay, BME
Balint Szabo, BME
Philipp Luithardt, USTUTT
Michael Gutknecht, USTUTT
Nadja Schoemig, WIVW
Armin Kaussner, WIVW
Fawzi Nashashibi, INRIA
Paulo Resende, INRIA
Benoit Vanholme, IFSTTAR
Sebastien Glaser, IFSTTAR
Philippe Allemann, Explinovo
Fredrik Seglö, HLX
Anders Nilsson, HLX

Project Managers

Alfred Hoess
Continental Automotive GmbH
Siemensstrasse 12
93055 Regensburg, Germany
Phone +49 941 790-5786
Telefax +49 941 790 99 5786
E-mail: Alfred.Hoess-EXT@continental-corporation.com

Holger Zeng
Continental Automotive GmbH
Siemensstrasse 12
93055 Regensburg, Germany
Phone +49 941 790 92330
Fax. +49 941 790 99 92330
E-mail: holger.zeng@continental-corporation.com

Project Co-ordinator

Reiner Hoeger
Continental Automotive GmbH
Siemensstrasse 12
93055 Regensburg, Germany
Phone +49 941 790 3673
Fax +49 941 790 13 3673
E-mail reiner.hoeger@continental-corporation.com

Revision and history chart

Version	Date	Reason
0.1	2011-05-13	Initial template by Alfred Hoess
0.2	2011-05-22	General part and SP2000 part included
0.3	2011-05-29	Comments from coordinator included, SP3000 part included
0.4	2011-06-02	SP4000 part included
0.5	2011-06-04	SP5000 part included, reviewer comments included; final formatting and final editing; submission to EC (note: deliverable draft version)
0.6	2011-09-18	EC reviewer comments implemented, Joint System chapter updated, minor editorial corrections
0.7	2011-09-19	Conclusions updated, document sent to partners for final review
0.8	2011-09-22	DSA chapter updated, final partner comments included, final editing and formatting, preparation for publication
	2011-09-23	Final deliverable version submittet to EC; publication via project web

Table of contents

Figures

Tables

Executive summary

HAVEit is an essential step forward to the realization of the long-term vision of highly automated driving for intelligent transport. The project developed, validated and demonstrated important intermediate steps towards highly automated driving. The results offer a high potential for exploitation within 3-7 years from project end. In the longer term they also form the ideal basis to integrate further next generation ADAS and drive train components that offer highly automated functionalities.

HAVEit significantly contributes to increased traffic safety and fuel efficiency for passenger cars, buses and trucks. The significant HAVEit safety, efficiency and comfort impact was generated by three aspects:

(i) At first a layered approach has been realized for the interplay between driver and the co-driving system, which optimizes the task repartition between driver and co-driving system in monotonous driving situations like traffic jams or long distance driving as well as in demanding situations like road works. This approach for optimum task repartition between the driver and the co-driving system takes driver alertness into account and forms the basis for all HAVEit applications addressing the fact that 95 percent of all accidents are driver related and more than 22 percent are related to missing driver alertness. Therefore, it is of utmost importance to ensure that the driver is in the loop when required. It has to be ensured that he or she is able to react properly in a potentially critical situation. Within HAVEit, a approach was be developed that is relatively new in automotive, but has been successfully implemented in automation concepts of other domains like aviation: Instead of just switching off an ADAS system in case of an impending potentially critical situation, a progressive step-by-step-approach was used to transfer the driving task back from the automated system to the driver. The interaction starts quite early in the event chain, i.e. few seconds before a potentially critical situation occurs. It brings the driver back into the loop in advance of the critical situation and provides him or her with the optimum level of automation and assistance needed in critical situations.

(ii) Secondly, a vehicle architecture scalable in terms of safety from fail silent to failure robust with advanced redundancy management was developed and successfully implemented. A further important focus of the architecture was enabling a rapid market introduction by usind technologies which are close to series development (CSC). Therefore, for less safety relevant system components a fail-safe ECU compliant with the Autosar standard and development methodology was developed and implemented (XCC). The aim of this development was to perfectly match the needs and requirements of highly automated vehicle applications and to arrive at optimal system availability and reliability. Addressing safety issues in a proper way in particular represents a key issue in steer-by-wire (e.g. HAVEit Joint System demonstrator) and brake-by-wire vehicles (e.g. HAVEit brake-by-wire truck). In case of the brake by wire truck a pre homologation was done to prove the maturity of the HAVEit architecture approach.

(iii) The third measure aimed at developing and validating a next generation advanced driver assistance systems (ADAS) directed towards a higher level of automation in comparison to the current state of the art by integration of hitherto independent ADAS functions. HAVEit implemented 7 pioneering vehicle applications for both passenger cars and trucks aiming at improved safety and comfort as well as improved fuel efficiency. The most important feature for support in terms of mental overload is represented by the automated assistance in roadworks. Key features for driver support in terms of mental under load are the automated queue assistance and the temporary autopilot. Finally, the active green driving application based on the energy optimizing co-pilot contributes to safe and ecological driving (of trucks and buses) by considering hybrid drive train and digital maps.

With these functionalities, HAVEit addressed the most important accident scenarios and ecological needs.

1 Motivation

1.1 Accident causation analyses

Accident analysis ATZ[5] show that 95% of the accidents are human related. Thus, driver support on different levels offers high potential to achieve enhanced road traffic safety. The main error prone behaviours are[5]:

- Wrong estimation of control variables, command variables and constraints are resulting from errors in parameter precision and the fact that a driver can only concentrate on just one area at a certain time.
- The driver provides just limited speed and precision in adjusting the control variables. Additionally, especially in physical driving borderline situations, the average driver does not have the requested capabilities to undertake stabilizing driving maneuvers.
- Inexperience, the failure results particularly from critical un-known situations in which the driver has no automatisms or rules available and the time-to-collision is too short to allow him or her to mentally go through different action alternatives (such as braking, steering etc.).
- Fatigue, drowsiness, complacency or other motivations/activities in parallel to the driving task detract driver attentiveness and reduce the driver's environment perception capabilities (environment perception acted out with the required accuracy or timing). Recent studies conducted in France for example, demonstrate that drivers who are going through a divorce have a quadruplicated accident risk. It is also fully accepted now that using a mobile phone while driving drastically increases the accident risk. The driver's emotional state, cognitive distraction and fatigue are generally characterized by an evolution of the driving style and the driving performance.

The 100 Car Study[6] reveals very important facts as well:

- Driver Inattention: Nearly 80 percent of all crashes and 65 percent of all near-crashes involved driver inattention just prior to (i.e. within 3 seconds) the onset of the conflict. Prior estimates have been in the range of 25 percent of all crashes. In 2/3 of all accidents with distraction involvement, the source has been identified:

  46.7% objects outside the vehicle
  22.5% operating radio / CD
  17.3% passenger
  6.8% moving interior objects
  4.1% eating, drinking, smoking
  2.4% phone calls

- Rear-End-Striking Crashes: Visual inattention was a contributing factor to 93 percent of rear-end-striking crashes. In 86 percent of rear-end-striking crashes, the headway at the onset of the event was greater than 2.0 s. Most near crashes involving a conflict with a lead vehicle occurred while the lead vehicle was moving.
- Driver Drowsiness: Contributing factor in 20 percent of all crashes and 16 percent of all near-crashes, while most current database estimates place fatigue-related crashes at a much lower percentage (i.e. under 10 percent) of all crashes. Driving dozily increases an individual's near-crash or crash risk by four to six times. Engaging in secondary tasks that

[5] Automobiltechnische Zeitschrift, ATZ, 01/2007, year 109, p. 58-62
[6] The 100-Car Naturalistic Driving Study, Phase II - Results of the 100 Car Field Experiment, Report No. DOT HS 810 593, April 2006

require multiple steps or eye glances away from the forward roadway, increases the risk by two to three times. Certain behavior has increased the risk of involvement in a near-crash or crash. Reaching for a moving object increased risk nine times, looking at an external object 3.7 times, reading 3.4 times, applying makeup 3 times and dialing a hand-held device 2.8 times. Looking away from the forward roadway for long glances at inopportune moments, increases the crash risk by two times in comparison to an alert driver.

- Table 1 summarizes the results from the United States 100 Car Study in terms of numbers of crashes, near crashes and incidents.

Conflict Type	Crash	Near-crash	Incident
Single vehicle	24	48	191
Lead vehicle	15	380	5783
Following vehicle	12	70	766
Object/obstacle	9	6	394
Parked vehicle	4	5	83
Animal	2	10	56
Vehicle turning across subject vehicle path in opposite direction	2	27	79
Adjacent vehicle	1	115	342
Other	0	2	13
Oncoming traffic	0	27	184
Vehicle turning across subject vehicle path in same direction	0	3	10
Vehicle turning into subject vehicle path in same direction	0	28	90
Vehicle turning into subject vehicle path in opposite direction	0	0	1
Vehicle moving across subject vehicle path through intersection	0	27	158
Merging vehicle	0	6	18
Pedestrian	0	6	108
Pedalcyclist	0	0	16
Unknown	0	1	3

Table 1: Number of crashes, near-crashes and incidents for each conflict type (taken from 100 Car Study, Table RO.3)

Between 1999 and 2001 deaths resulting from accidents involving trucks accounted for around 16% of all road accident fatalities in the Netherlands[7]. This means that approximately one in every six deaths on the road was caused by a collision with a truck. Similar figures were reported In 1998 from Great Britain[8]:HGVs[9] were involved in 17% of all road accident fatalities despite making up just 7% of the traffic on the roads of Great Britain. This demonstrates the fact that HGVs are more likely to cause a fatality when they become involved in an accident than other vehicle types. According to the European Road Safety Observatory, only 13 % of the fatalities in accidents involving vehicles were occupants of HGVs. Other vehicles involved in these accidents suffered a majority of the fatalities (59 %)[10]. All these facts show the necessity for measures which reduce the involvement of heavy trucks in accidents.

[7] SWOV (2003), Cognos, SWOV, Leidschendam, NL in On track!
Results of the trial with the Lane Departure Warning Assistant system (2004), Ministry of Transport, Public Work and Water Management, AVV Transport Research Centre, Rotterdam
[8] TRL Project Report Fatalities from accidents involving vehicles - trends, causes and countermeasures, Dec 1999
[9] HGV: Heavy Goods Vehicle
[10] Web site : http://www.erso.eu

Safety can be improved by these three factors: infrastructure, vehicle and driver. As 95% of all accidents are driver related, the HAVEit project aims at the third factor. Two measures for driver improvements are feasible:

- Improving the reliability of the driver, e.g. by feedback monitoring. This has been successfully demonstrated by the projects AIDE, SPARC, AWAKE etc.
- Increasing the substitution of the driver both in “middle & long-term” - corresponding to a higher degree of automation with first demonstrations like in Chauffeur and on “short-term” – meaning intervention in critical scenarios.

While SPARC focused mainly on the short-term substitution (typically less than half of a second), HAVEit aimed at extending the working range with integration of longer term substitution (up to few seconds), thus leading to a higher level of automation and furthermore to the transition to short-term substitution.

The problem of driver inattention reported by the 100 Car Study was addressed in HAVEit by the realization of a co-system, which takes the driver state (e.g. drowsiness) into account and implements a step-by-step transfer of the driving task from the highly automated system back to the driver. The problem of the high number of crashes, near crashes and incidents for different conflict types was addressed by the HAVEit highly automated vehicle applications.

1.2 Highly automated driving vision to improve overall safety

HAVEit supports the necessity of higher integration and more reliable and driver sensitive support and intervening safety functions to improve road traffic safety. HAVEit, thus, represents a significant step towards higher road safety by combining intervention and substitution models. The HAVEit consortium interprets the long-term "Highly Automated Driving" vision in following sense:

The key actor in safe driving must be the driver. Everything has to be done to optimize her or his performance; therefore we built the automation centered on the driver. Taking into account that the need for assistance strongly depends on the varying performance level of the driver, the need for a dynamic balance or flexible task partition between driver and automation becomes obvious. A higher degree of automation in this context means to support the driver in monotonous driving tasks (e.g. queuing on crowded motorways, continuous lane keeping) as well as in highly demanding tasks like driving through the narrow lanes in roadwork areas. Automation must be designed in a way that different degrees or stages of support can be flexibly produced (ranging from mere warning up to a temporary auto-pilot etc.).

- One of the key issues is represented by finding the optimum way for sharing the driving task between the driver and the highly automated system. If the driver is allowed to be out of the loop for a few seconds, he or she needs to get a few seconds time to get back into the loop to react properly (i.e. to continue driving) in case the highly automated system detects a scenario that cannot be handled automatically. Current state of the art systems and other research systems warn the driver in such cases and just switch off, thus leaving the driver alone in the critical situation, in some cases even without prior warning. In HAVEit, a novel approach was followed to overcome this problem: A step-wise strategy (starting sufficiently in advance of an up-coming situation that might not be handled automatically) was developed to give the driving task back to the driver to ensure that the driver is really capable to react accordingly to the driving situation. Due to its safety relevance, the optimum task repartition in the joint system driver / co-system (i.e. automation system) has been identified as one key objective of this project and is considered to be a horizontal activity.

- The multi-stage concept for optimum task repartition in the joint system was made available to the vertical application activities aiming at different innovative safety and comfort functions.
- Systems aiming at a higher degree of automation and at safety definitively need to include the possibility of intervention to avoid a collision or mitigate a crash in case the driver is out of the loop for a dedicated moment. To take this safety requirement into account, highly automated vehicle applications need to integrate some kind of minimum risk maneuver, e.g. emergency brake function.

2 Project objectives

2.1 Three HAVEit key objectives

To support the vision of highly automated driving, three key objectives (sorted according a top-down approach) were derived to achieve the targeted safety benefits:

2.1.1 Development and validation of next generation ADAS functions as a co-system

Safety problems in driving arise from two sources: From the driving situation itself and from the current performance level of the driver. Assuming a U-shaped function (see Figure 1) between driving difficulty and performance, a fundamental relationship discovered 1908 already by the psychologists Yerkes and Dodson[11], research has shown that the need for assistance arises at least in two different scenarios.

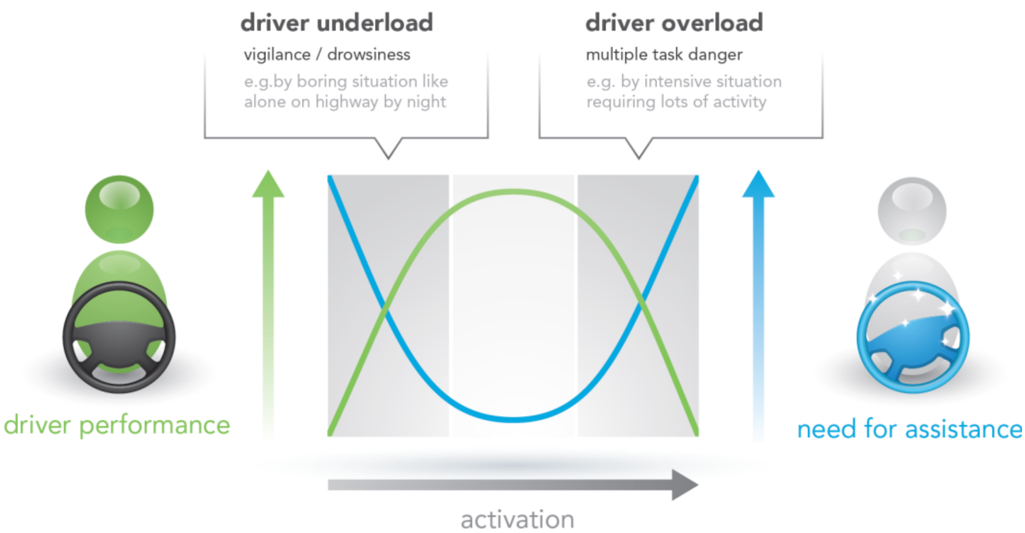


Figure 1: U-shape function between driving difficulty and driver performance

In case of monotonous driving, mental underload will lead to decrements of vigilance and in consequence to dangerous situations. In case of highly difficult situations (like driving in the narrow lanes of roadwork areas) symptoms of stress will result in a deteriorated performance. Highly automated driving is a promising way to avoid both difficult regions and is therefore directly linked to traffic safety. The third component of highly automated driving tries to keep the driver on the optimal performance level as long as possible by facilitating the normal driving task.

Therefore, one key objective of HAVEit is represented by the development and validation of next generation ADAS functions (based on a co-system) directed towards a higher degree of automation in particular to support the driver to arrive at a higher level of safety, i.e. reducing accidents.

[11] Yerkes RM, Dodson JD (1908). "The relation of strength of stimulus to rapidity of habit-formation". *Journal of Comparative Neurology and Psychology* **18**: 459–482. http://psychclassics.yorku.ca/Yerkes/Law/

2.1.2 Optimum system joining and interaction between driver and co-system

Up to now, automated systems are switched either “on” or “off”, a severe disadvantage in case of systems integrating different functions like HAVEit. If one part of the integrated system shows malfunction or reaches its system limits, the remaining parts of the integrated system may still work correctly or must only be degraded in part. The same holds true for the driver. There may be an intention for maximum automation in some situations, whereas partial assistance may fit better to other driving situations. Also, different alertness states of the driver (e.g. sleepiness) may prohibit switching in a highly automated mode. Therefore, the basic idea of HAVEit is to define different degrees of automated driving which can be selected according to the needs of the driving task.

On the one hand, offering different adaptable degrees of automation is not trivial. On the other hand, this is already working successfully in a lot of non-car vehicles. As a first example, imagine a modern aircraft. Besides automation, there is one main reason why aircrafts are usually safer than cars: There is a second entity, a second brain in the cockpit. A co-pilot, who checks the actions of the pilot, who can take over certain tasks to relieve the pilot in case flying would wear out or bore the pilot (left side of the U-shape), or overload the pilot (right side of the U-shape). In aircraft with only one pilot, the role of the co-pilot can be performed by an advanced electronic system or electronic co-pilot (e.g. Champigneux et. al. 1989, Flemisch & Onken 1999[12]). Co-pilots in cars, e.g. in rally cars, usually do not drive themselves, but the comparison can still be exploited and used for technical systems in cars and trucks that act like aviation co-pilots. Car prototypes were demonstrated e.g. in (Hassoun et al.1993[13]) or (Holzmann 2007[14]). Another important inspiration for HAVEit came from other base research project that once again started in aviation and went to automotive: In the H-Mode project, the highly automated vehicle is not compared to a co-pilot, but to intelligent animals like horses, These animals are are not so intelligent as a human, but very skilled in movement, have own preservation instincts, and can be cooperatively controlled in a spectrum between more directly, with a tight rein, and more indirectly with a loose rein (e.g. Flemisch et al. 2003[15], Goodrich et al. 2006[16], Flemisch et. al. 2011[17]).

In the HAVEit project, experience from the above mentioned examples is used carefully and blended with more pragmatic approaches: HAVEit vehicles will of course not be as intelligent as a real co-pilot or as real horses, but will contain a sophisticated co-driving system (named co-system in this project) that can, in limited driving situations, perform a higher percentage of the driving task automated (i.e. highly automated). The co-system is usually dependent on the driver to allow, supervise and/or participate in the automated behavior. If well designed, driver and co-systems form an ideal symbiosis, a “Joint System” that drives better and safer than any of the two partners would be capable alone. One essential key to such a successful combination of human (driver) and automation (co-system) lies in the proper design of the transitions between lower and higher degrees of automation.

[12] Flemisch, F.O.; Onken, R: The Search for Pilot’s Ideal Complement, Experimental Results with the Crew Assistant Military Aircraft CAMA, HCI International, Munich, 1999

[13] M.Hassoun, C.Laugier (Lifia) and D.Ramamonjisoa, N.LeFort (Heudiasyc): "towards Safe driving in traffic situation by using an electronic copilot" Proceedings of the IEEE Intelligent Vehicles Symposium, Tokyo Japan, pp 84-89, July 1993

[14] Holzmann, Frédéric: Adaptive Cooperation between Driver and Assistant System, Improving Road Safety. Springer, Berlin, 2007

[15] Flemisch, F.O.; Adams, C. A.; Conway S. R.; Goodrich K. H.; Palmer M. T. ; Schutte P. C.: The H-Metaphor as a guideline for vehicle automation and interaction; NASA/TM—2003-212672; NASA Langley Research Center; Hampton, Va, USA; 2003

[16] Goodrich, K.; Flemisch, F.; Schutte, P.; Williams, R.: A Design and Interaction Concept for Aircraft with Variable Autonomy: Application of the H-Mode; Digital Avionics Systems Conference; USA; 2006

[17] Flemisch, F.; Bengler, K.; Bubb, Heiner; Winner, H.; Bruder, R.: Towards a Cooperative Guidance and Control of Highly Automated Vehicles: H-Mode and Conduct-by-Wire; Submitted to Human Factors Special Edition Automation in Vehicles, Ed. Natasha Merrat & John Lee; 2011

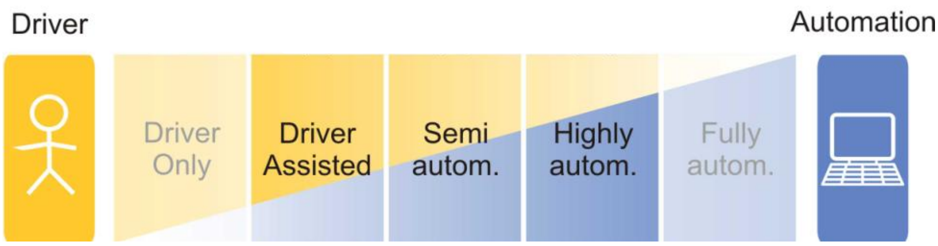


Figure 2: Automation levels considered in HAVEit - the stepwise transfer of the driving task forms the basis for optimum task repartition in the joint system

In HAVEit, over a couple of iterative design and evaluation cycles, three different levels of assistance and automation were selected: 1.) Driver-assisted, where the driver is in full control, but the co-system supports with a light assistance e.g. during lane change. 2.) Semi-Automated / ACC+, which integrates the "classical" and more sophisticated ACC-systems. 3) Highly automated driving, where the co-system does most of the driving, but the driver is still in the loop and can take over the driving task anytime (see e.g. deliverables D 33.3[18], D33.6[19]).

The rationale of the development is shown in the figure above. In addition to the definition of meaningful stages of automation, special care has to be given to the transitions between stages in both directions. These transitions may be initiated by the driver with respect to his or her intention or by the co-system with respect to the needs of the situation.

To achieve the above key objective, existent ADAS functions were integrated and expanded to a joint system driver / co-system whereby driver and co-system are considered as "observer" and complementary of each other. A mode selection and arbitration unit allows selecting the adequate automation mode (also by means of substitution and active intervention) and decides about the most suitable driving command. The main idea behind this development is illustrated in Figure 3 below.

Based on her/his mental representation of the driving situation and her/his driving intention, the driver gives a command which is processed through the mode selection and arbitration unit before being executed via the drive control. On the other hand, based on the sensor representation of the driving situation, the co-system develops its own system command strategies (all safe maneuvers, see block co-pilot in the diagram), which are also fed into the mode selection and arbitration unit.

In addition, the mode selection and arbitration unit (MSU) gets information about the current state of the driver as well as about the state of the co-system. Driver's state is mainly influenced by the degree of attentiveness, of activation state and of his or her awareness about the situation and the automation state of the joint system. But the state of the co-system in terms of sensors and system reliability, error level and system limits must also be taken into account by the MSU before deciding which command should be executed. The result of this process is a safe motion vector which then will be sent to the execution level to be performed. Both, the driver and the co-system must be informed about any change of the automation mode (internal state) and the environment (external state).

Obviously, the degree to which the driver is "in the loop", as well as his mental model of the system, are key determinants in the joint system. The state of the driver strongly modifies the decision about the action (e.g. in case of drowsiness the joint system will try to rely more on the commands of the system and start actions to bring the driver back into control). In the same way, quality and quantity of the feedback about the final action are adapted to the state of the

[18] Flemisch, F.; Schieben, A.(Ed.): Validation of preliminary design of HAVEit systems by simulation (Del. 33.3). Public deliverable to the EU-commission; Brussels; 2010
[19] Schieben, A.; Flemisch, F. (Ed.): HAVEit Del. 33.6., 2010

driver (e.g. in case of overload the feedback has to be weak, in case of underload strong). Evidently, difficulties may arise if the stage of automation is changed (transition problem). Up- or downgrading the automation must be represented to the driver as well as to the co-system. Adequate measures must be developed to guarantee that each actor in the joint system has sufficient mode awareness.

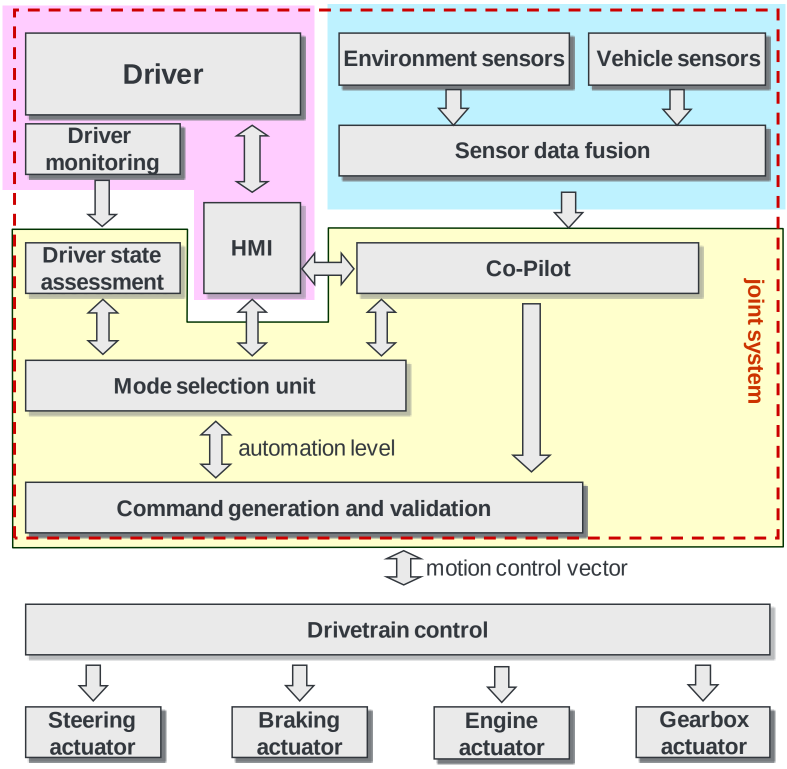


Figure 3: System design for optimizing the task repartition in the joint system driver - co-system

Therefore, the next key objective of HAVEit is represented by the design of an optimal interaction and task repartition in the joint system driver / co-system. Special emphasis has to be given to the assessment of the degree to which the driver is in the loop and possible countermeasures to bring him/her step-by-step back, e.g. in case of oncoming situations which the highly automated system may not be able to handle on its own.

2.1.3 Development and validation of failure tolerant, scalable, safe vehicle architecture

In case of the need for strong, fast and accurate actuation (e.g. in an emergency braking situation in curves), in many situations smart actuators (e.g. activated by the emergency behavior of the co-system) are superior to most humans. To further improve traffic safety, HAVEit therefore followed a third main objective: The development and validation of a safe and flexible architecture combining redundant, failure tolerant (so-called XCC[20]) and fail-safe, scalable automotive ECUs (so-called CSC[21]) communicating via suitable automotive bus systems (FlexRay, CAN) to arrive at both better economical use of resources, improved safety and higher overall vehicle reliability.

[20] X-by-wire Control Computer
[21] Chassis and Safety Controller

The redundancy management in the XCC platform forms an abstraction-layer between safety[22] and functionality. Safety thereby can be characterized via integrity[23] and reliability[24]. The software layers can be visualized as shown in Figure 4. The platform approach is illustrated in Figure 5.

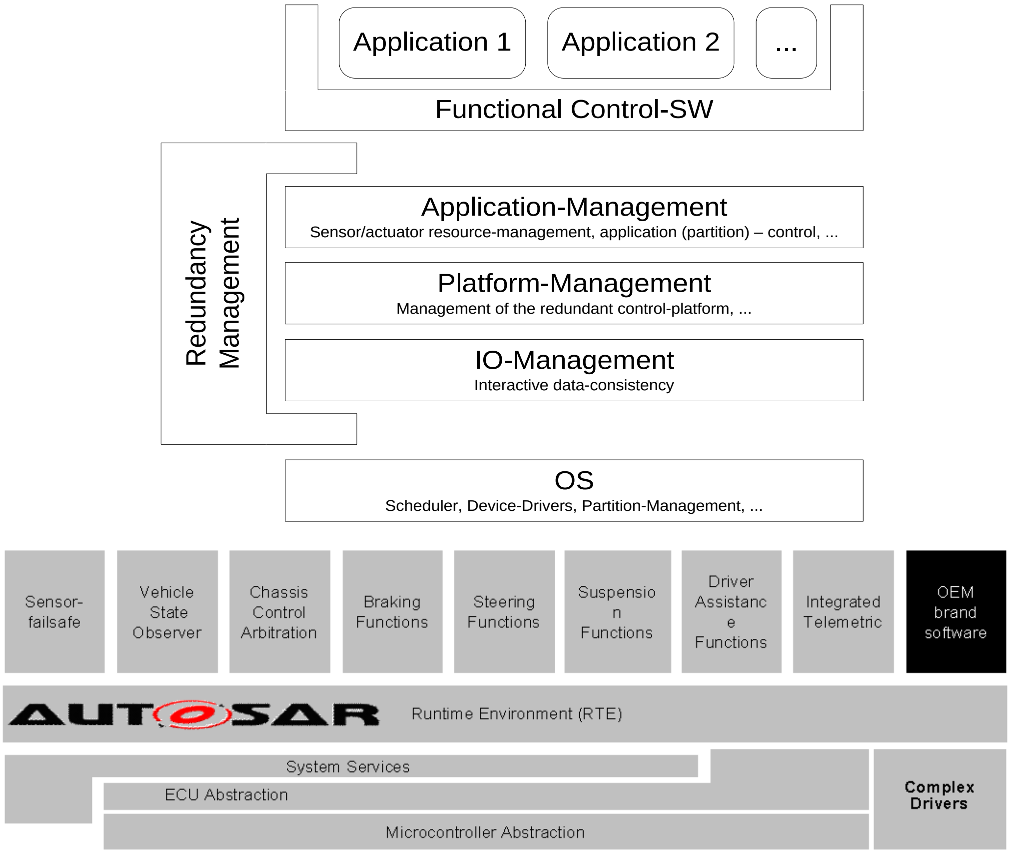


Figure 4: HAVEit safety architecture software layers
a) fail-tolerant XCC system (for safety critical actuators) and
b) fail-silent CSC system[25] (for joint system implementation, AUTOSAR compatibility)

With respect to safety[26], the key tasks of the redundancy management can be split up in failure[27] treatment (management of checking redundancy), based on interactive data consistency and failure tolerance (management of operational redundancy). Only in case that the checking redundancy is working properly, the operational redundancy can be used. The

[22] Safety: Describes a state in which risk is lower than the boundary risk. The boundary risk is the upper limit of the acceptable risk; it is specific for a technical process or state (ARP 4754).

[23] Integrity: Attribute of a system or an item indicating that it can be relied upon to work correctly on demand (ARP 4754).

[24] Reliability: The probability that an item will perform its intended function for a specified interval under stated conditions (ARD 50010).

[25] Failure detection in the CSC system is achieved by hardware redundancy means.

[26] operational functionalities as control-sequencing of the communication- resp. energy-clusters or the high-level control-functions are not take into account within this chapter.

[27] error: An omission or incorrect action by a crewmember or maintenance personnel, or mistake in requirements, design, or implementation (AMC 25.1309).

failure: An occurrence, which affects the operation of a component, part, or element such that it can no longer function as intended, (this includes both loss of function and malfunction). Note: Errors may cause Failures, but are not considered to be Failures (AMC 25.1309).

fault: A physical condition that causes a device, component, or element to fail to perform in a required manner; for example, a short circuit or a broken wire (ARD50010).

quantification of the corresponding required detection and monitoring rates in order to reach the reliability per function can be done using a Markov model[28]. This can also be used to derive the degree of operational redundancy based on a given board reliability of a single control computer.

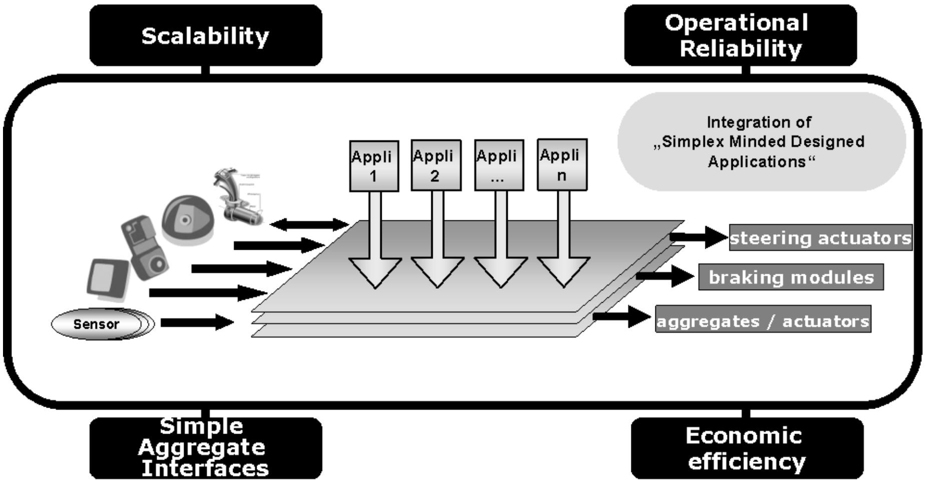


Figure 5: HAVEit safety platform approach (built on the SPARC project)

The degree of checking and operational redundancy is thereby not only dependent on the criticality of a function but also on the basic concept of the vehicle:

- advanced driver assistance with mechanical backup (e.g. electric steering actuator, joint system modules): use of fail-safe CSC ECUs and Autosar methodology
- advanced driver assistance without mechanical backup (e.g. steer-by-wire and brake-by-wire actuators): use of failure-tolerant duo-duplex XCC ECUs

With these three key objectives HAVEit paves the way for higher levels of automation by integrated ADAS functions and significantly contributes to improved road traffic safety. Moreover, the unique and flexible architecture with open interfaces is ideally suited to integrate additional functionalities by adding software modules (e.g. based on of a standardized runtime environment, such as AUTOSAR).

2.2 Scientific and technological objectives: Challenges and functional clusters derived from the three key objectives

The three key objectives presented above were broken down to challenges which were clustered according to their nature: technological challenges were grouped in two so-called horizontal clusters. Results achieved from them were made available for all highly automated vehicle applications, which were organized in two so-called vertical clusters, see Figure 6.

[28] R. Reichel: Flight Control Systems – Fly-By-Wire, at – Automatisierungstechnik, Volume: 52, Issue: 12-2004, pp: 588 – 595, Oldenbourg Wissenschaftsverlag GmbH

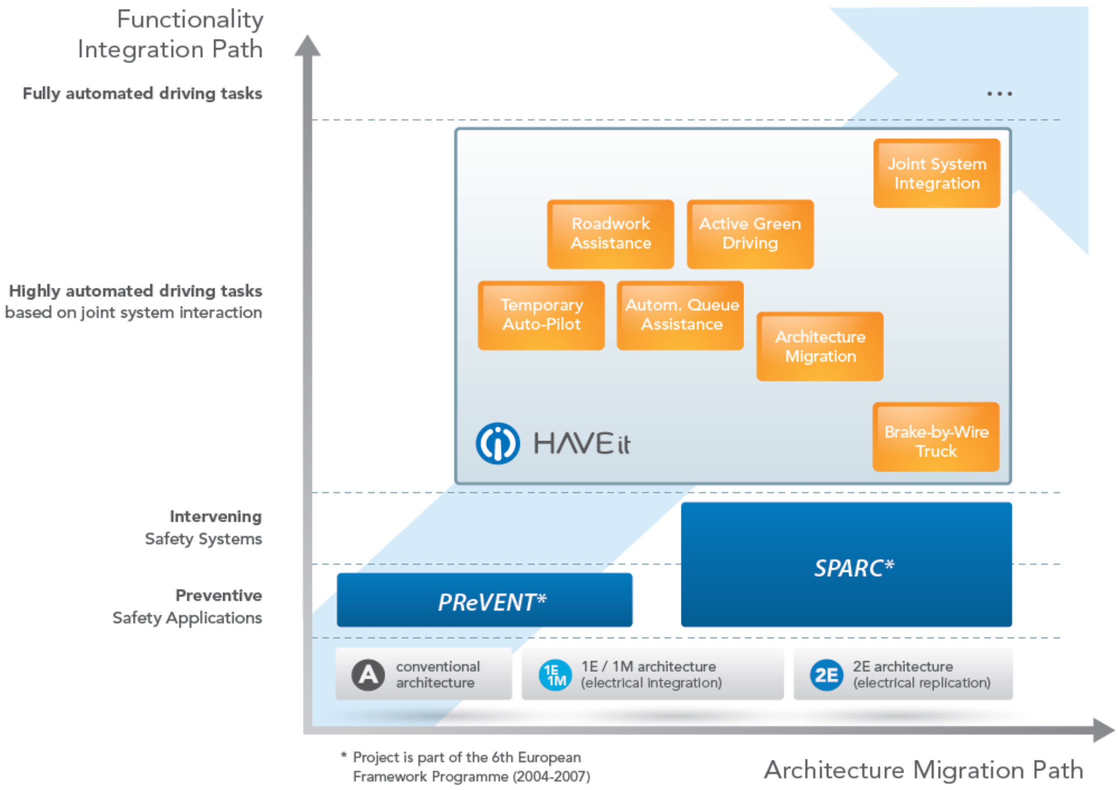


Figure 6: Introduction of the highly automated driving migration path

<u>"Horizontal" clusters leading to results to be included into vertical applications:</u>

- Cluster 1: Integration challenges (for reasons of clarity not outlined in Figure 6).
- Cluster 2: Implementation of the failure tolerant, redundant, flexible and scalable safety architecture aiming at improved traffic safety and at suiting the needs of future safety systems, i.e. highly automated functions.
- Cluster 3: Situation adaptive, optimized task repartition and interaction in the joint system driver - co-system aiming at improved safety.

<u>"Vertical" application clusters:</u>

- Cluster 4: Joint system and safety architecture validation: Validation of the innovative Joint System interaction in a generic car ("rapid prototyping" approach as a first step). Improved ergonomics and safety for vehicles on the basis of the scalable, failure redundant XCC 2E architecture and on the basis of the scalable, fail-silent CSC concept to be extended in HAVEit towards automation capabilities. Further intention was to show the migration path to the new HAVEit architecture.
- Cluster 5: Highly automated driving applications suitable in public traffic for continuous driver support and improved road traffic safety and efficiency.

Obviously, these five technical clusters mark migration paths in two directions (see Figure 6): Migration in the safety vehicle architecture direction (horizontal axis) and migration towards higher levels of functionality (vertical axis). Both migration directions are linked: Considering e.g. the cluster 5, in case of the first actuator failure, today's co-system would stop working. In contrast, a vehicle equipped with a fault tolerant HAVEit architecture will use the actuator's redundancy management to overcome at least some failures, thus being able to continue driving. Using the HAVEit mode selection and arbitration unit in case of an actuator failure the

ADAS function would be decreased gradually but not stopped (e.g. not allowing the highest automation mode as a potential further actuator failure would require faster reaction of the driver). Different steps in both architecture direction (horizontal axis) and automation level direction (vertical axis) are expected in the direction of "highly automated driving".

HAVEit also defined which levels of architecture integration are required depending on the levels of automation (corresponding to the different vertical clusters) to overcome correctly the risk of failures. Consideration of relevant scenarios lead to the simple rule: The higher the level of automation, the higher the safety and failure tolerance requirements. Accordingly, the arrow in the background of Figure 6 is pointing to the right top of the illustration is indicating the overall migration path.

The challenges assigned to each functional cluster are briefly described next.

- ***Horizontal cluster 1: Integration challenges***

  This cluster comprises integrating activities common for all technical tasks:

  Challenge 1.1: Requirements and specifications

  One of the main issues consisted in the definition of the functional specifications and of an agreed and validated set of specifications. HAVEit cared for the systematic collection and tracking of any requirements throughout the project. Work in this challenge included the design of general systems structure, e.g. of the joint system “driver - co-system” (to give just one example). The different levels of automation were defined as well as the relevant factors (both from the driver and the co-system) which are used to select the automation level. Feedback strategies were developed both to maintain and to get back mode awareness of the driver.

  Challenge 1.2: Derivation of the overall safety architecture:

  Based on and partly in parallel with the definition of requirements at the different system levels partners developed the common overall system architecture. In the meaning of a top-down-approach, partners further agreed on a common architecture including common interfaces and protocols from the top system level down to as much as possible subsystem and component levels (e.g. vehicle architecture, co-system architecture, sensor system architecture, processing architecture including communication matrices), thereby forming the basis for all further technical developments in this project.

  Challenge 1.3: Application optimization and validation

  Having achieved first highly automated functionalities, partners entered the common optimization and validation phase of the project. Close cooperation helped to foster the common understanding and the sharing of key knowledge achieved in terms of the various horizontal and vertical activities and second to prepare for the common, final demonstration of the validation vehicles (during the final event in June 2011).

- ***Horizontal cluster 2: Safe vehicle architecture implementation***

  This cluster covered the development of the safe and scalable vehicle architecture. The results were made available to all vehicle applications:

Challenge 2.1: Development, integration and verification of a safe control platform

a) Failure-tolerant duo-duplex platform for safety relevant actuators (XCC with integrated redundancy management)

On the basis of the SPARC results, duo-duplex ECUs (so-called XCCs) were developed and provided for the x-by-wire actuators (steer-by-wire in WP4100 and brake-by-wire in WP4200).

Out of the platform-requirements the XCC platform functional description can be generated and broken down to the HW/SW co-design. As in HW and SW just functionalities can be realized, it is the task to transfer all functional and safety requirements to pure elementary functions and to allocate those to HW and SW. A challenge within this work package resulted out of the task to provide an optimum platform for different validation vehicles with a high degree of configurability and scalability. Therefore, an intelligent configuration mechanism was developed to derive out of a database the full configuration data for the platform consisting out of the central platform computer, subsystems, the overall communication network(s) and the test-bench.

Having integrated this platform, the high-level control functions (for steer-by-wire in the WP4100 passenger car and for brake-by-wire in the WP4200 truck) could be linked to the redundancy management. The integration verification then took place first in the lab and afterwards the platform was integrated in the vehicles and the functional tests could start.

b) Fail-silent platform for safety relevant modules, e.g. the joint system (CSC, AUTOSAR compatible runtime environment)

Based on the partners' strong product experience in the field of chassis and safety systems, a powerful ECU was developed and offered to all partners. In particular the CSC platform was used for the joint system integration, as - based on the AUTOSAR runtime environment - it represents the ideal platform for the easy integration of software developed by different parties on the same ECU.

Challenge 2.2: Communication

This challenge has been planned to precisely define the overall communication common for all HAVEit highly automated systems. As explained above, this step is required for the full configurability of the individual systems and the possibility of an easy exchange of software modules between partners. Due to the safety relevance of the HAVEit applications and the large amount of data to be handled, FlexRay communication was required in addition to CAN communication.

HAVEit functionalities and in particular most future HAVEit+ applications can benefit from the incorporation of communication channels to exchange relevant information with other vehicles and infrastructure. Suitable communication channels (hardware, software and protocols) have already been developed in other projects, e.g. CVIS. HAVEit adopted these existing modules to the HAVEit architecture to be compatible to widely accepted communication standards. Both, directional IR V2V and non-directional RF V2I communication channels were integrated. In the HAVEit architecture V2V and V2I communication channels are considered as additional sensors.

Challenge 2.3: Fast, smart and reliable actuators

The development of smart, intelligent high speed actuators for the application challenges 4.1 and 4.2 and 5.1 described below was considered a further horizontal challenge. Innovative steering actuators including control strategies are required for improved ergonomics for drive-by-wire applications (e.g. challenge 4.1 with 2E actuator providing no mechanical fallback during normal operation).

Novel brake-by-wire actuators, electro-mechanical wheel brake (EMB) actuators utilizing the principle of self-enforcement for generating brake force, are a development to a higher level of maturity. The brake system was integrated in a truck and be pre-homologated for the first time for use on public roads (challenge 4.2). Compared with state of the art pneumatic brake systems, stopping distance was significantly reduced. Further, EMB reduced the energy consumption during the braking sequence.

- ***Horizontal cluster 3: Situation adaptive, optimized task repartition in the joint system driver - co-system***

  To achieve this key objective, for optimum task repartition between driver and co-system, an innovative situation adaptive interaction concept was developed and implemented which takes driver intention, driver attention, vehicle status as well as the co-system command vector into account. Using probability measures to assess confidence of both driver and co-system command vectors, a combined vehicle command vector is calculated ensuring safe and stable reaction of the vehicle on the one hand and allowing the stepwise transfer of the driving task from the highly automated system back to the driver or vice-versa in case of an imminent situation. As mentioned above, such adaptive cooperation in the joint system is considered as a key issue for future safety systems.

  To achieve this key objective, three challenges were identified:

Challenge 3.1: Co-system

One input channel of the mode selection and arbitration unit is represented by the co-system command. Parallel to the driver, a co-system integrating the ADAS functionalities provides this safe command. Furthermore, this co-system has a cognition model similar to the real driver. A perception layer by means of sensors and data fusion module provides an environment model to the knowledge layer. Similar to a real driver this layer performes like a multi-agent system which integrates different control tactics for different situations. The co-system needs to monitor itself to define a confidence value for the mode selection and arbitration unit like for the driver. However, a self estimation was developed here directly by combining information from the data fusion (a-priori knowledge) and matching rate of the functionality patterns (in situ knowledge).

Challenge 3.2: Driver in the loop assessment

The second input channel is assigned to the driver state information. Both direct (e.g. monitoring the driver vigilance with a camera system) and indirect measures (evaluating driver activities, such as history of pedal, steering activities, vehicle behaviors) were combined to achieve the required high confidence level for the driver vigilance information.

Relevant dimensions are on one hand attentiveness, drowsiness and stress level of the driver, on the other hand the driver’s mental model of the system (e.g. considering distraction caused by daydreaming etc.). Methods and parameters to assess the degree the driver is in the loop were developed.

The development also needed to include dangerous test scenarios (falling asleep, critical traffic situations). Therefore, driving simulators needed to be used as a development tool, e.g. simulation software SILAB (from WIVW, driving simulators of DLR, VTEC and VW).

Challenge 3.3: Joint system design and validation

It is further essential to understand the driver command and maneuver: Why does he react in a specific way in a certain scenario? An environment model needed to be developed and implemented for this purpose. Understanding the driver intention allows the generation of important expert knowledge which in turn can be used for the further optimization of the co-system.

From the previous challenges it was concluded that the interaction between the elements of the joint system hads to be designed according to the degree the driver is in the loop. This includes information about (1) the relevant dimensions of system and situation awareness, (2) the methods to assess this awareness, and (3) measures to keep the driver in the loop and accordingly countermeasures to bring him/her back step-by-step into the loop in case he is not (sufficiently early before a situation escalates that might not be handled by the highly automated system).

Offering different degrees of automation also means that transitions between these levels will occur. Therefore, special emphasis was given to the mode awareness, i.e. the driver is always aware about the current level of automation.

After having developed and validated the algorithms for optimum task repartition in the joint system by simulation means, these were made available to the vertical applications (semi-automated driving tasks in public traffic).

- ***Vertical cluster 4: Flexible and scalable safe vehicle architecture***

  The vehicle architecture migration path is outlined in the horizontal axis of Figure 6. The use of a "conventional system architecture" is not well suited for the development of cluster 5 applications. For the long-term vision of automated driving additional architectural features are required.

  The second step in terms of a failure tolerant architecture is represented by the "1E/1M architecture" (i.e. having full electronic control over actuators but a mechanical fallback solution) This steps give the necessary degrees of freedom to realize the HAVEit functionality and is reliable enough due to the fact that HAVEit requires to have the driver in the loop.

  For full by-wire application a so-called 2E architecture is needed. Because there is no mechanical fall back the control electronics and the power supply to the actuators must be designed failure tolerant. In SPARC a first proof of concept for the 2E integration has been achieved (by using a duo-duplex ECU architecture).

  To validate the architecture migration path including scalability, within HAVEit different architecture integration steps were demonstrated for different highly automated vehicle applications. To achieve HAVEit's third key objective, challenges 4.1, 4.2 and 4.3 have been defined to validate the scalable safe vehicle architecture and its safety and economical benefits:

Challenge 4.1: FASCAR - Extended joint system demonstration, JSD

One challenge in developing the HAVEit approach is represented by the parallel development of two crucial points: The development of joining of and interaction between the driver and the co-system; and the development of the hardware architecture for implementation. To avoid the development risks that failures will be first detected in a real vehicle very late in the process we introduced a generic experimental car (FASCar) where initial implementations can be pre-integrated and tested right from the beginning of the development process and closely tied to the evaluation in the simulators.

Another challenge in developing the HAVEit approach was that in the horizontal activity of joint system design and validation, a generic concept was developed and then tailored to a couple of realistic applications in the vertical activities of the vehicle demonstrators. This activity had only a chance if the generic joint system could also be integrated, tested and demonstrated in a test vehicle before it was tailored down to the realistic applications.

The FASCar from DLR was optimized for this iterative development cycle. Its software and hardware architecture were compatible with the simulator environments, and ready to host preliminary versions of the safe HAVEit vehicle architecture as early as possible, so that integrations and demonstrations could be shifted with minimal distance between the initial, generic architecture and the final redundant architecture.

As mentioned earlier, this generic vehicle was also used for the validation of the 2E steering activities for drive-by-wire applications to show extended possibilities for improved ergonomics by a proper design of the steering feedback.

Challenge 4.2: EMB-Truck – Brake-by-Wire for public roads, BbW

This vehicle is the logical extension of the development done during the SPARC project. After having shown the feasibility of EMB for every kind of road vehicles and the reliability of a drive-by-wire platform, the next step was to prepare x-by-wire components for pre-homologation.

Based on the Haldex brake actuators developed in challenge 2.3, the EMB system replaced the current pneumatic actuators to brake the vehicle. Very important in terms of safety is the fact that a heavy vehicle using an EMB system achieved a shortened stopping distance of 15% in average compared to existing systems on the market. To further develop the system towards public road operation was an essential step to commercialize this new technology, which will make automized stop and go more comfortable and save enegy, because there is no need for permanently provide compressed air for the pneumatic brakes.

The resulting vehicle (installation and integration of the truck was done jointly by Volvo 3P and Haldex) was used to build a safety case and to start the pre-homologation procedure to operate the system on public roads. Pre-homologation was done in close cooperation with subcontractor TUEV Nord, Germany.

Challenge 4.3: Architecture migration demonstrator, AMD

While challenge 1.2 covered the derivation of a safe vehicle architecture and related paper work, challenge 4.3 was dedicated to the prototypic realisation of the architecture aiming at first at the demonstration of migration steps to bring the new highly automated systems into products and at second to show the integration of basic HAVEit functionalities into the vehicle in a safe and economic way by making use of the flexible and safe architecture derived in challenge 1.2.

The control flow of HAVEit (see Figure 3) forms the basis of this demonstrator and is common in all HAVEit vehicles. The architecture was developed using a top-down methodology, intending to achieve a structure being basically independent of the exact functional content. A common motion control vector was defined as the interface between the command layer and the execution layer which are generally independent of the underlying vehicle platform. While for most other HAVEit demonstrators a couple of PCs were used to realize the joint system, the architecture migration demonstrator is based on control units that are closer to the market, i.e. wherever possible the generic fail-silent ECU developed within HAVEit challenge 2.1 was used.

- ***Vertical cluster 5: Highly automated driving (in public traffic) aiming at continuous driver support and improved road traffic safety***

  This functional cluster includes the development and validation of 4 innovative safety, comfort and active green driving applications:

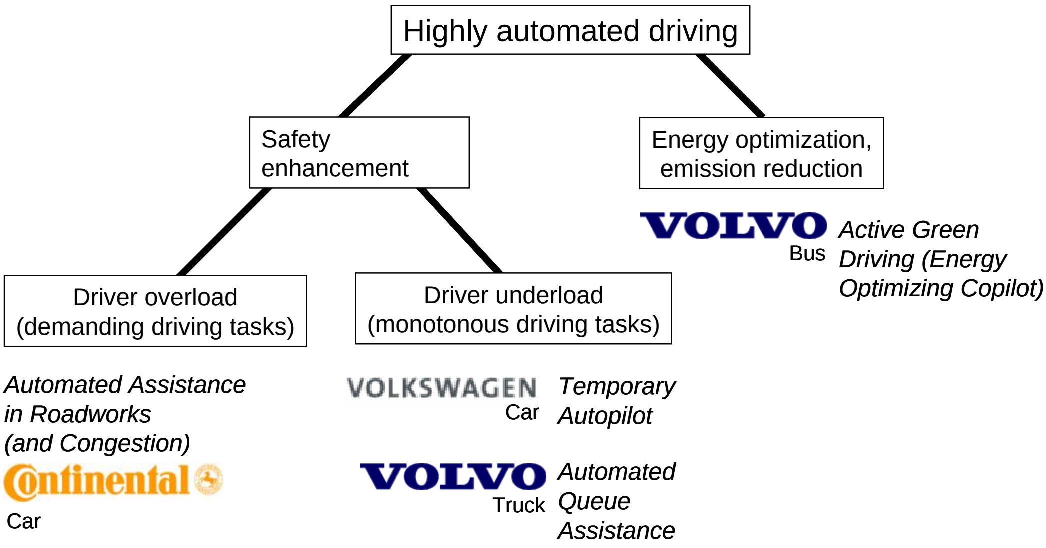


Figure 7: Structure and objectives of the different highly automated driving applications (challenges cluster 5)

Challenge 5.1: Automated assistance in roadworks and congestion, ARC (passenger car application)

This application is directed towards highly automated driver support in particular when driving through a roadwork area. Special challenges for automated driving through a road-work area are narrow lanes and curves, ambiguous lane marks, changing speed limits and other vehicles driving closely next to the own vehicle. In case the vehicle aside moves laterally just some 20cm, the lane may become too narrow so the own vehicle needs to intervene quite hard. To cope with this potentially very critical situation, the experimental car needed to feature emergency braking functionality.

Automated assistance in roadworks and congestion integrates different automation functionalities and levels. In particular when driving through the narrow lanes of roadworks special care needs to be taken on the transitions between the different automation levels and on the evaluation of the system limits. Obviously, the complex roadwork scenarios bring today's state of the art sensing technologies toward their limits.

Challenge 5.2: Automated queue assistance, AQuA (HGV application)

Automated queue assistance supports the driver in motorways and rural roads by integrated longitudinal and lateral control (thus representing a new generation of automated cruise control). In reality this means that the system continuously supports the steering, accelerating and braking of the vehicle.

The level of automated control is continuously adapted based on the state of driver, vehicle and environment. At the highest level of automation the system autonomously handles steering, accelerating and braking to keep the vehicle in the correct lateral position in the lane, at desired speed and at a safe distance to other vehicles. In particular, this is possible in low speeds and dense traffic down to a complete vehicle stop. The system functions and supports the driver also in non-congested situations and higher speeds but the level of automation is determined by the performance of the perception system and user acceptance.

Challenge 5.3: Temporary auto-pilot, TAP (passenger car application)

The temporary autopilot system developed in HAVEit integrates three different levels of functionalities: pilot functionality (hands-off driving), e.g. driving in a traffic jam; assisted driving (hands-on driving, driver in the loop), i.e. driving in normal traffic mode, e.g. driving when traffic jam terminates; and intervening safety functions, e.g. driver initiated emergency braking.

To achieve these high and complex functionalities, a dedicated sensor platform needed to be developed. It comprises a set of ultrasonic sensors which detect other vehicles and pedestrians in front of the ego vehicle, a front view and a rear view mono camera for the detection of lane markings, a 77 GHz radar sensor for the detection of vehicles, a laser scanner for the detection of vehicles and pedestrians and an electronic horizon, e.g. for the detection of speed limits.

Challenge 5.4: Active green driving, AGD (energy optimizing co-pilot, bus application)

The intention of the active green driving application was to use a "forecast" on the vehicle movement over the next period of time (e.g. 30 seconds) to optimize the powertrain control of the hybrid bus. Hybrid buses are of particular importance in urban areas. It is anticipated that there will be "zero-emission" zones in city centers. In that case, even buses and trucks need to operate with zero emission in these areas.

It is a big challenge to optimally control the complete system in order to achieve the best balance between saving energy and providing optimal power. Regarding overall fuel consumption of the HAVEit active green driving concept, according to the evaluation results achieved in HAVEit, fuel and CO_2 reductions between 6-8% beyond the reduction by a normal average hybrid citybus can be achieved. This number is depending on the urban driving cycle and on the driver’s driving style.

According to the clusters of challenges derived to achieve the HAVEit key objectives, each cluster has been assigned an own sub-project, while each challenge in a certain cluster has been assigned a work package. Consequently, 5 technical sub-projects result from the 5 HAVEit clusters. These were accompanied by a 6th sub-project covering management tasks. The links between the different horizontal and vertical work packages of the HAVEit Integrated Project are summarized in the matrix shown in the figure below.

SP 6000: Project management (CAG) WP 6100: Overall management WP 6200: Financial management WP 6300: Steering Committee WP 6400: Project management support WP 6500: Dissemination and exploitation WP 6600: Link to other projects (liaison) WP 6700: Legal issues, homologation	**SP 4000: Safety architecture applications** (DLR)	WP 4100: FASCAR – Joint system demonstrator	WP4200: EMB truck – BBW for open roads	WP4300: Architecture migration demonstrator	**SP 5000: Highly automated driving (in public traffic)** (VTEC)	WP 5100: Assistance in roadworks and congestion	WP 5200: Automated queue assistance	WP 5300: Temporary autopilot	WP 5400: Active green driving
SP 1000: Integration tasks (CAG)									
WP 1100: Requirements and specification		☒	☒	☒		☒	☒	☒	☒
WP 1200: Architecture definition		☒	☒	☒		☒	☒	☒	☒
WP 1300: Optimization and validation		☒	☒	☒		☒	☒	☒	☒
SP 2000: Safety architecture implementation (CAG)									
WP 2100: Generic ECU and redundancy management		☒	☒	☒		☒	☒	☒	☒
WP 2200: Communication (FlexRay, C2C, C2I)		☒	☒				☒		☒
WP 2300: Actuators		☒	☒	☒		☒			
SP 3000: Joint system driver - copilot (DLR)									
WP 3100: Co-driving system		☒		☒		☒	☒	☒	☒
WP 3200: Driver in the loop assessment		☒		☒		☒	☒	☒	☒
WP 3300: Joint system design and validation		☒		☒		☒	☒	☒	☒

Figure 8: HAVEit work package structure and interaction between horizontal and vertical activities

3 HAVEit requirements and specification

3.1 Requirements Engineering

To fulfill the needs of all challenges relevant for highly automated vehicles, HAVEit defined and run through a very structured requirements and specification phase. Due to the size of the integrated project HAVEit and its organisation in horizontal and vertical subprojects leading to a manifold of links between partners and results to be exchanged, the consortium has agreed on a tool based methodology to derive requirements and specifications. The requirements and specification phase was split into a sequence of steps illustrated in in Figure 9.

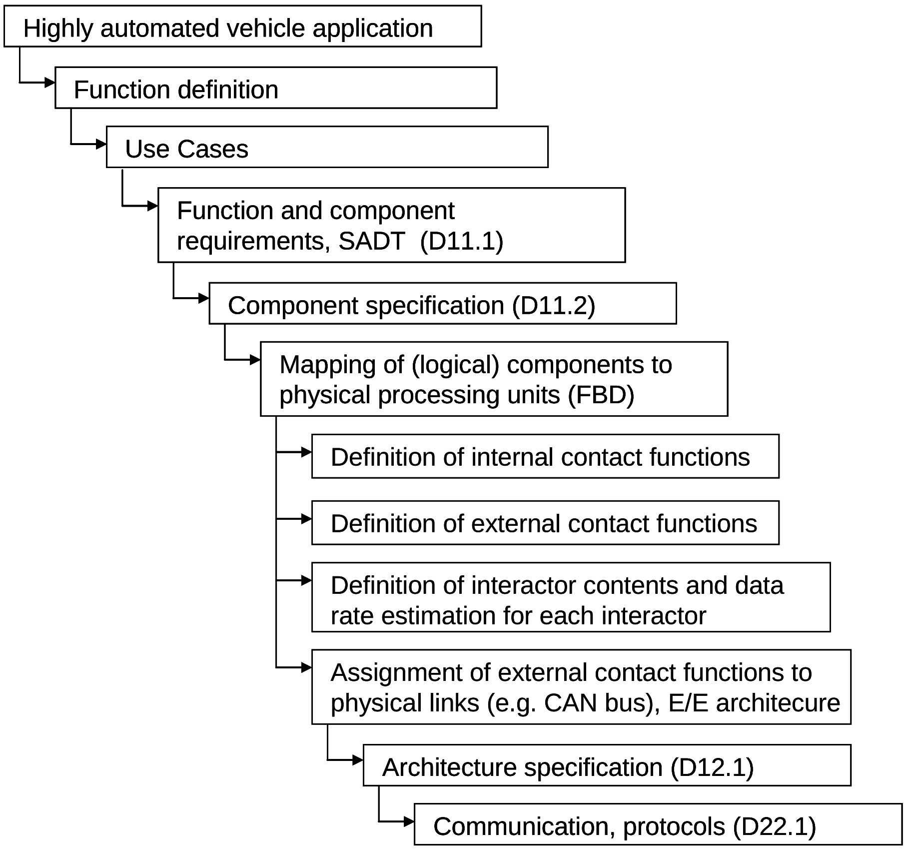


Figure 9: Different steps of the requirements and specification phase

a) According to this top-down methodology, at first the HAVEit highly automated functions were defined in detail: Joint System Demonstrator (JSD), Brake-by-Wire Truck (BbW), Architecture Migration Demonstrator (AMD), Automated Assistance in Roadworks and Congestion (ARC), Automated Queue Assistance (AQuA), Temporary Auto-Pilot (TAP) and Active Green Driving (AGD). Second, a set of common and application specific use cases was derived, which form the basis for the definition of test cases and test scenarios for system validation at project end.

   On this basis the partners defined and agreed on a common functional structure for all highly automated systems. Functional requirements for all HAVEit applications have been structured using the so-called SADT (Structured Analysis and Design Techniques) technique that presents the information flow between the different functions and sub-functions. Interactors are introduced to handle the information exchange between the different functions. Figure 10 presents as an example for the SADT view of the HAVEit top-level function: “Bring driver and vehicle from A to B in the traffic flow with different levels of automation”.

Results achieved at function level are described in detail together with all use cases in HAVEit deliverable report D11.1 “Function description and requirements”. This public deliverable is available for download from the project web www.haveit-eu.org.

b) Dedicated components needed to realize the functions were defined. According to the employed methodology, these are organized in a so-called component tree. A generic block diagram has been derived for all the required components. The component tree for example includes sensors, HMI elements and actors, but also more intelligent components like processing blocks for the driver state assessment (DSA), or the mode selection and arbitration unit (MSU) required for switching between different automation levels. Resulting from this step, a detailed specification document was generated. A public version of this deliverable (D11.2 “Specification document”) is available for download from the project web.

c) An assignment of the processing blocks (components) to physical processing units (ECUs) was made. For instance, a generic ECU (CSC[29] platform) includes the DSA. This physical processing unit including the DSA software is used in the same way for several HAVEit demonstrators, e.g. JSD, AMD, ARC, AQuA and TAP.

d) According to the chosen methodology, the physical processing units were arranged in the merely hardware oriented functional block diagram (FBD). Due to the fact that each of the seven HAVEit demonstrators realizes different functionalities, there are differences in the sensor systems and also in the physical processing units. To consider such differences, it was decided to start with a generic FDB and to generate the vehicle specific FBDs from that. An example is given in Figure 11 which presents the functional block diagram of the architecture migration demonstrator The FBDs are described in detail in HAVEit deliverable D12.1 “Architecture document”, also here, a public version is available.

e) Function blocks included in the SADT view the exchange of information utilizing so-called interactors. An interactor includes one ore more parameters, e.g. vehicle speed, acceleration, yaw rate etc. The list of interactors used in the HAVEit systems is included in D11.1. Within this step, the interactors which include clear specifications for all information required or delivered from each function block need to be mapped to links between the physical processing units. In the tool methodology used in HAVEit these links are called contact functions. In HAVEit we distinguish between internal contact functions (this means contacts inside a certain processing unit, e.g. the CSC ECU) and external contact functions (these are contacts outside the processsing unit).

f) In this stage all data to be transferred by means of the so-called external contact functions are known from the interactor specification. On basis of the data rate required for each external contact function, a mapping of several contact functions to a certain bus system (physical link, such as CAN or FlexRay) can be made, thus defining the bus system for each demonstrator vehicle.

Again, the architecture migration demonstrator network topology shown in Figure 12 serves as an example for this methodological step. Internal contact functions will not cause traffic on the bus system, thus can be neglected for the bus load estimation. Internal data rates can be very high, thus will not cause any transfer problems. Detailed results from these steps are presented in the architecture deliverable D12.1.

[29] CSC: chassis and safety controller (fail-silent ECU)

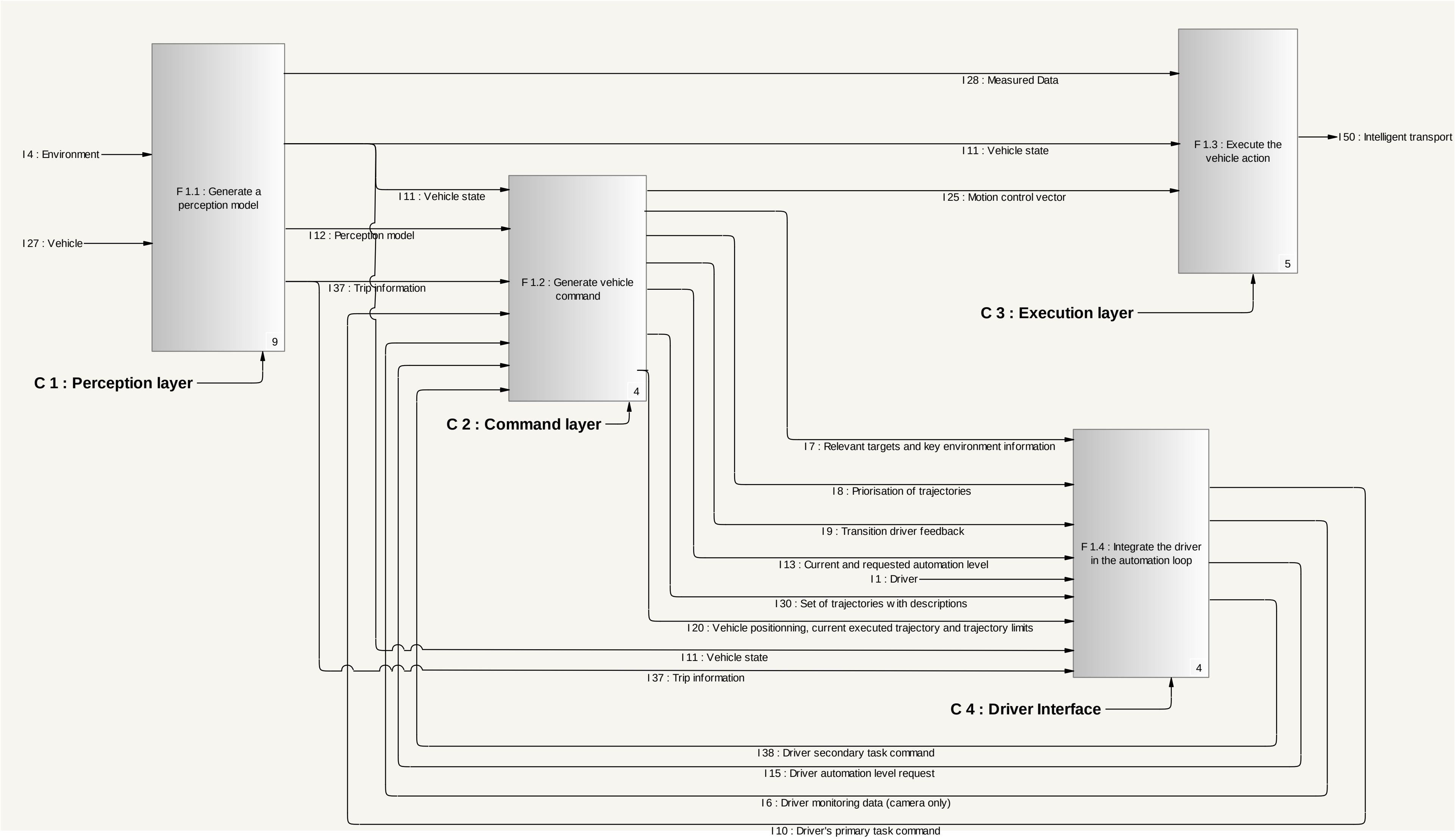


Figure 10: SADT view: Top-level function: "Bring driver and vehicle from A to B in the traffic flow with different levels of automation"

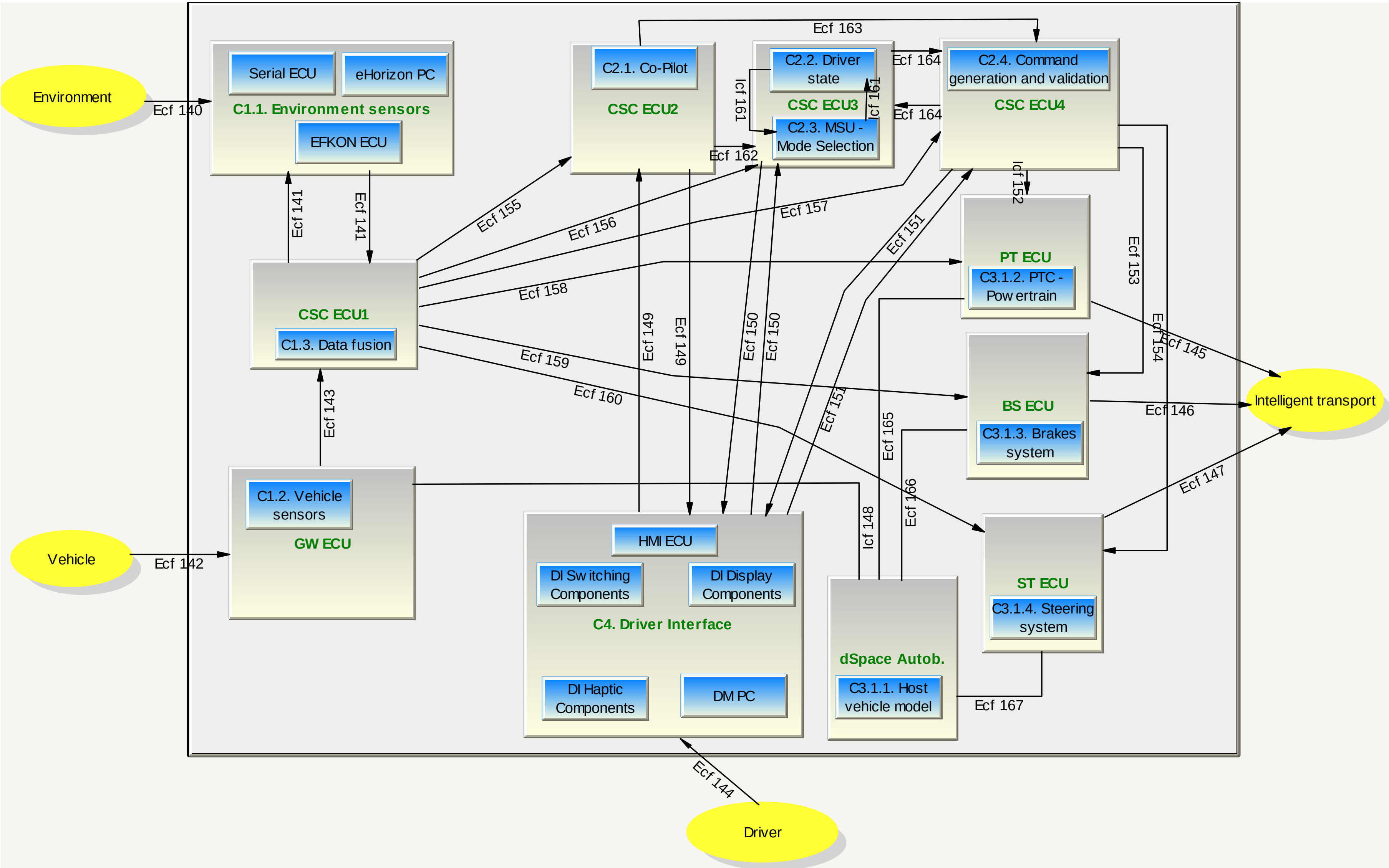


Figure 11: HAVEit generic functional block diagram (basis WP4300)

g) Finally, the communication protocols (e.g. CAN matrices, FlexRay communication etc. can be fixed. Results are presented in HAVEit deliverable D22.1 “FlexRay and CAN communication available”.

As public version of requirements, specification and architecture deliverables are available, in this document we intend to set the frame just by including a brief functional description of the different HAVEit demonstrators. For more detailed information, the reader is kindly referred to HAVEit deliverables D11.1, D11.2 and D12.1. Consequently, the next sections provide a description of the HAVEit highly automated vehicle applications introduced above by the two application clusters. All applications were realized and demonstrated in relevant use cases and scenarios during the HAVEit final event on June 21-22, 2011 at Volvo Proving Ground Hällered, Sweden.

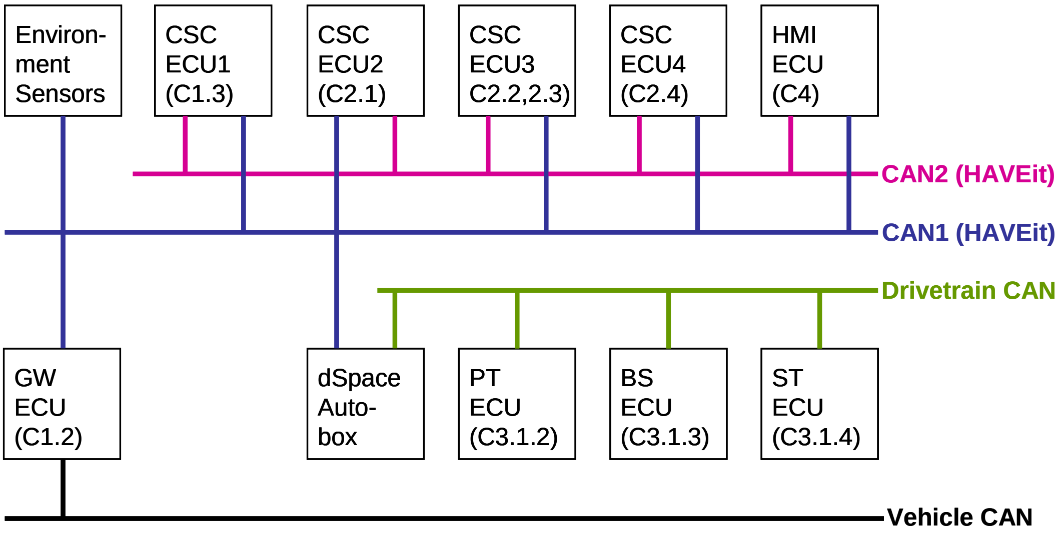


Figure 12: Architecture migration demonstrator network topology

3.2 The HAVEit Architecture: 4 Layers

During the function definition and requirements phases, the global HAVEit structure was defined and agreed for all HAVEit demonstrators. It consists of four layers:

- Perception layer[30]
- Command layer
- Execution layer
- Integration of the driver in the automation loop (HMI layer)

According to these layers, the SADT view of the HAVEit systems was structured (see section 4.2.1 in D11.1). This structure is also reflected in Figure 13 below, which illustrates the highest level information flow common to all HAVEit applications. One key issue of research in the HAVEit project is the optimum task repartition between driver and co-system. Both parts need to understand and supervise each other to arrive at acceptable highly automated vehicle appli-cations. Components deeply involved in this interaction are combined in the so-called "Joint System" framed with a red dashed line in Figure 13.

[30] Note: In HAVEit V2V and V2I is considered as sensor. Thus, “environment sensors” include V2V and V2I communication channels (both IR and RF based) as well as on-board environment sensors (e.g. radars, laser scanners, cameras).

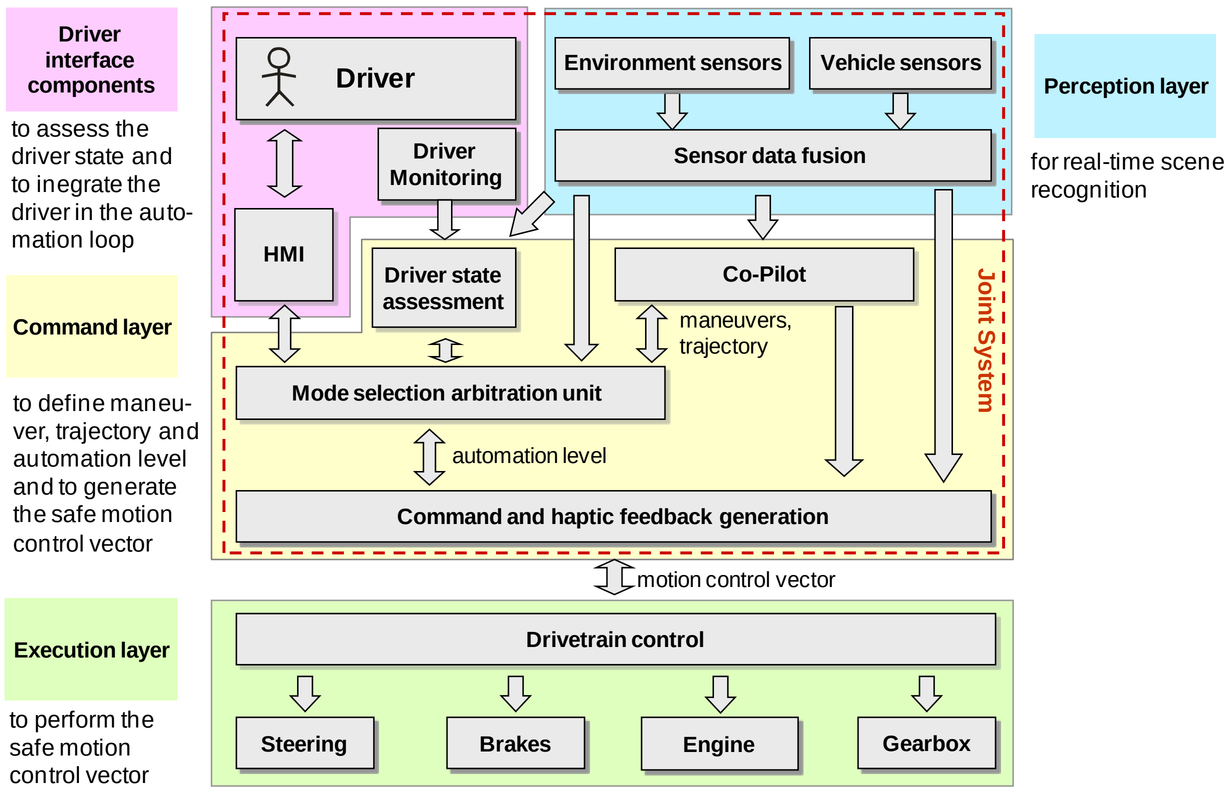


Figure 13: HAVEit overall block diagram

- Perception layer (C1)

  The intention of the perception layer is to collect information about the external environment and the vehicle. Different kinds of sensors are used in the HAVEit demonstrators for the detection of the environment, e.g. radar sensors (front, rear and side), lidar sensors, camera sensors and communication (vehicle to vehicle, vehicle to infrastructure). Required information concerning the vehicle state is achieved by various vehicle sensors, e.g. wheel speed and acceleration sensors, yaw rate sensors as well as further signals provided by the ESP, for instance. The information of all environment and vehicle sensors is checked for plausibility and merged into the data fusion module.

- Integration of the driver in the automation loop (C4)

  One key element for the command layer is represented by the perception system. The other key element is given by the driver who needs to be integrated in the automation loop. His/her commands are collected by different HMI elements (e.g. buttons). The information concerning the automation level and the overall system status is presented to the driver, e.g. by audible, visual or haptic means.

- Command layer (C2)

  The HAVEit command layer essentially consists of following parts:

  - Copilot component: First, the driving strategy needs to be defined (e.g. a decision, whether to slow down behind a preceding vehicle or to overtake, is required). Second, the most suitable trajectory to continue driving is to be calculated (considering vehicle stability issues and information about the environment from the perception layer).
  - Driver activities were recorded and assessed in order to derive information about the driver's alertness (indirect driver monitoring). Furthermore, a camera based

driver monitoring system was installed that evaluates the driver state in a direct way. Both channels, direct and indirect driver monitoring, were merged to attain more reliable information about the driver's alertness (e.g. vigilance, distraction) and his/her possibilities in the automation loop.

- Mode selection and arbitration unit, MSU: On the basis of the copilot system on the one hand and the driver commands including the results of the driver in the loop assessment on the other hand, the mode selection and arbitration unit decides about the most suitable automation mode. In case a change of the automation level is the consequence, suitable transition strategies (including the interaction with the driver) were established.
- Command generation and validation: Based on the copilot trajectory to be followed and the information about the automation level to be used as well as by considering direct driver control (e.g. emergency brake), the vehicle command is calculated. This command leads to the so-called secure motion vector (longitudinal and lateral vehicle control commands). Final plausibility checks (and comparison with the physical limits, e.g. in terms of vehicle stability) are applied before the motion vector is transferred to the execution layer.

- Execution layer (C3)

The main task of the execution layer is to execute the longitudinal and lateral control commands given by the secure motion vector. To realize these commands and with that to control the vehicle drive train, the execution layer get access to the engine management, brake and steering systems (so-called actuators).

Concerning safety requirements, it is necessary to notice that in comparison to all layers mentioned before, the cycle periods are much shorter (typ. 10 ms). Furthermore, as the driver is not able to intervene on this level, the safety requirements are much higher. This especially applies to demonstrators that were equipped with by-wire actuators (i.e. WP4100 with steer-by-wire and WP4200 with brake-by-wire).

- In HAVEit, these stronger safety requirements for the execution layer in WP4100 and WP4200 were considered with the use of so-called XCCs – control units providing a duo duplex architecture, suitable redundancy management as well as dual channel FlexRay communication between the safety critical components. Overall, this concept results in a fail-operational execution layer.
- The execution layer in all other demonstrators was realized using fail-silent components, i.e. the Continental CSC – Chassis and Safety Controller.

3.3 Highly automated vehicles for safety architecture validation

As described before, the HAVEit idea of smart intelligent vehicles does not only rely on the development of new functions, the interaction with the environment and with the behavior of the human driver but also significantly on the availability of new actuatorsand the appropriate architecture For this reason, the entire system consisting on the vehicle itself, the human driver and new technical components becomes more complex and unforeseen effects might occur. Due to this condition, it is beneficial to make use of rapid prototyping methods and tools in the project.

Therefore, two vehicle platforms are available in HAVEit to prove the capabilities of the developed by-wire technologies and the appropriate architecture solutions. The first one is the DLR steer-by-wire vehicle FASCar which performs easy integration methods of new technologies and allows replacing the driver's inputs on gas, brake and steering by software commands. The second one is a truck provided by Haldex and Volvo 3P that was equipped with a brake-by-wire system.

With both vehicles the idea of developing and demonstrating the safety architecture components was fulfilled in a convincing manner. These components are new sensors and actuators, the joint system (driver / co-system) as additional driving authority and of course safe data concepts like the FlexRay bus system or the redundant control architectures. In order to show the strong migration potential of these technologies a third demonstration vehicle "architecture migration demonstrator" was defined, which realizes the HAVEit architecture using state-of-the-art automotive control units.

3.3.1 Joint system driver / co-system

The joint system development and validation includes two related aspects:

- The formulation, discussion and validation of the joint system ontology (common language) and principles, that can be commonly understood and applied throughout the HAVEit project The joint system ontology and principles include:
  - The automation spectrum with defined levels of automation.
  - The transitions between the levels of automation.
  - The dynamic task reallocation including automation initiated transition based e.g. on the driver state.
  - The multimodal driver interface strategy.
- The design, implementation and validation of a generic joint system demonstrator. It was implemented and tested in simulators and in a couple of test vehicles. The joint system is built of:
  - A perception layer including the vehicle sensors to the outside environment and to the driver.
  - A command layer including an intelligent co-pilot system and a mode selection and arbitration unit (MSU). The co-pilot is based on command generation utilizing trajectory and maneuver based automation.
  - An execution layer including the controllers for longitudinal and lateral control. This layer includes experimental controllers with special regard to the stability in automation transitions.
  - A driver interface including the primary driving task interface like steering wheel and gas pedal, and specific HAVEit interfaces like automation level switches.

The automation spectrum

The automation levels that are important for the HAVEit project were mapped on a spectrum of automation (Figure 14). The spectrum is on the left border defined by the level "Driver only", in which the driver has full manual control over the vehicle, and on the right border by the level "Fully automated", in which the automation controls the vehicle completely. The three other levels of assistance and automation and their sublevels lie in-between. The spectrum can be seen as a mapping of different degrees of driver involvement in the driving task. In the following, the automation levels and their sublevels as well as the three additional states "Off", "Minimum risk state" and "Failure" are described.

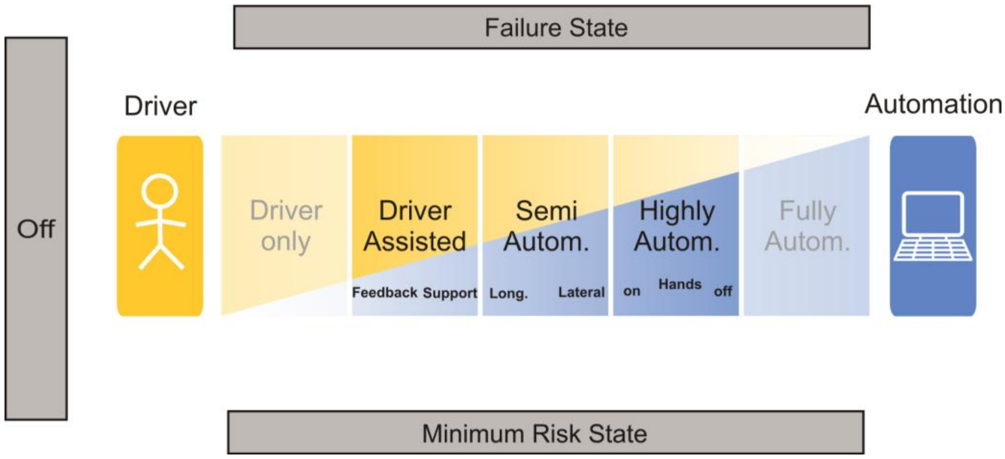


Figure 14: Automation spectrum with different automation levels, sublevels and additional states

Description of the automation levels

- *Driver only:* This is the left border of the spectrum in which the driver has 100% manual control over the vehicle (only the classical driver assistant systems like ESP, ABS etc. are operational). *Driver-Assisted:* A level in the automation spectrum between “Driver only” and “Semi-automated”.

  Feedback: A sublevel in the level “assisted”. This includes informing feedback from the system like acoustic, visual or vibratory lane departure warnings and parking assistance, but not yet direct intervention of the automation with the primary driving task.

  Support: A sublevel in the level “assisted". The provided assistance is the same as the one provided by the "Feedback" sublevel. Additionally, the system intervenes directly during the primary driving tasks, if the current vehicle situation is considered to be too dangerous by the system. Functions are lane keeping support, emergency braking and avoiding of objects, for example.

  For reasons of easier understanding and avoidance of mode confucion, automation levels *Driver only* and *Assisted* were combined to the automation level *Driver Assisted*.

- *Semi-automated:* Automation of about half of the driving task. One example is the Adaptive Cruise Control (ACC), where the complete longitudinal control is done by the automation (sublevel: “Longitudinal”). Other examples are some Lane Keeping Systems that can almost do the complete lateral control of the vehicle (sublevel: “Lateral”). In HAVEit, the longitudinal support was subsummized into

- *Highly-automated:* The highest automated level of the HAVEit system where automation for longitudinal and lateral vehicle guidance is combined, but where the driver is still involved in the driving task by monitoring the system. Automation level Highly Automated is devided into two sub-levels:

  - Hands on: During highly automated driving the driver is not allowed to take away his hands from the steering wheel (e.g. at higher speeds).
  - Hands off: During automated driving the driver is allowed to take away his hands from the steering wheel (e.g. while driving in congestion).

  It becomes clear that the border between Hands on and Hands off is quite complex, considering e.g. the duration for how long the driver is allowed to take off the hands and the reactions of the automation if the driver takes the hands off too long.

- *Additional States*

  - *Off:* the automation is completely switched off (Joint System is by-passed).

- *Minimum risk state:* a state of the vehicle that the automation might try to reach in case the automation is no longer capable of performing a certain level of automation. Therefore, the driver needs to take over. In case the driver does not take over in an appropriate time span, the automation might try to reach the safe state with the minimum risk, e.g. by performing a safe stop of the vehicle.
- *Failure:* a state which is reached if the automation works incorrectly or does not work at all, e.g. due to a hardware or software failure. It is clear that the occurrence of this state has to be minimized. To minimize this technically and scientifically, the automation spectrum has to include this state explicitly.

Transitions between automation levels

The transition of control is defined as the changing of an automation level to another whereby the driver or the automation gets more control over the vehicle depending on the automation levels.

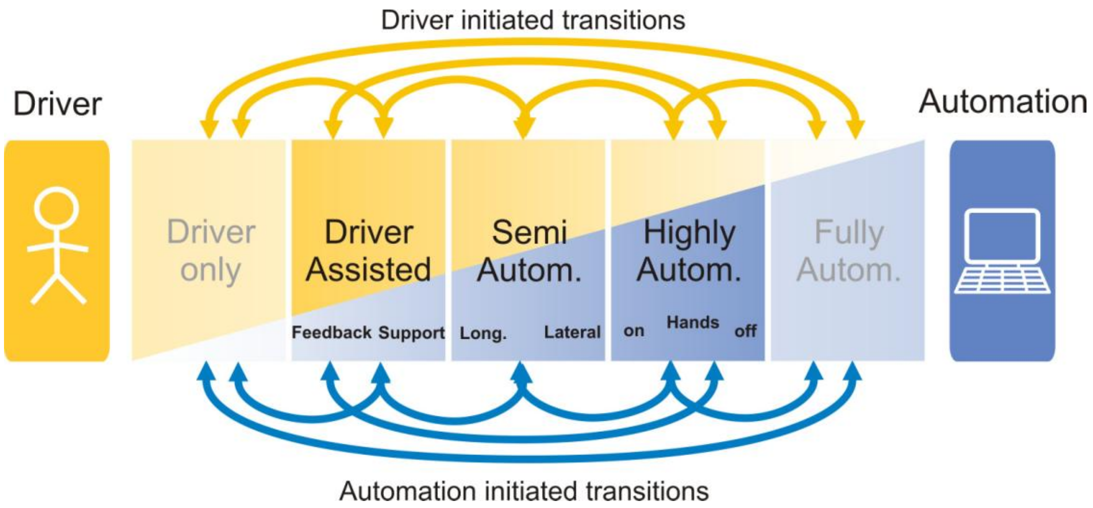


Figure 15: Possible transitions within the automation spectrum

Transitions can be classified by their direction toward a higher automation level and toward lower levels of automation. These different types of transitions are included schematically in Figure 15. Arrows above the automation spectrum stand for transitions initiated by the driver. Arrows below the spectrum are used for transitions initiated by the automation. The directions of the arrows indicate if the transition is classified as transition towards automation or transition towards the driver. Both aspects can be integrated in a short notation (Table 2):

Short Notation	$D_i \rightarrow A$	$D \rightarrow A_i$	$D_i \leftarrow A$	$D \leftarrow A_i$
Direction of transition	Transition towards automation	Transition towards the driver		
Initiation	by driver	by automation	by driver	by automation

Table 2: Classification and short notation of different transitions

For the transitions between different automation levels, HMI concepts had to be defined. To keep an overview over the complex human-automation interaction, we used a UML-based diagram to bring out the main interactions in a time line. Figure 16 shows this timeline for one specific interaction between the driver and the automation at the beginning of a traffic jam. The grey vertical lines describe who has control over the vehicle. The light grey line stands for longitudinal control, the dark grey line for lateral control. The arrows indicate the interaction and communication flow and show which interaction elements are planned (e.g. acoustic warning).

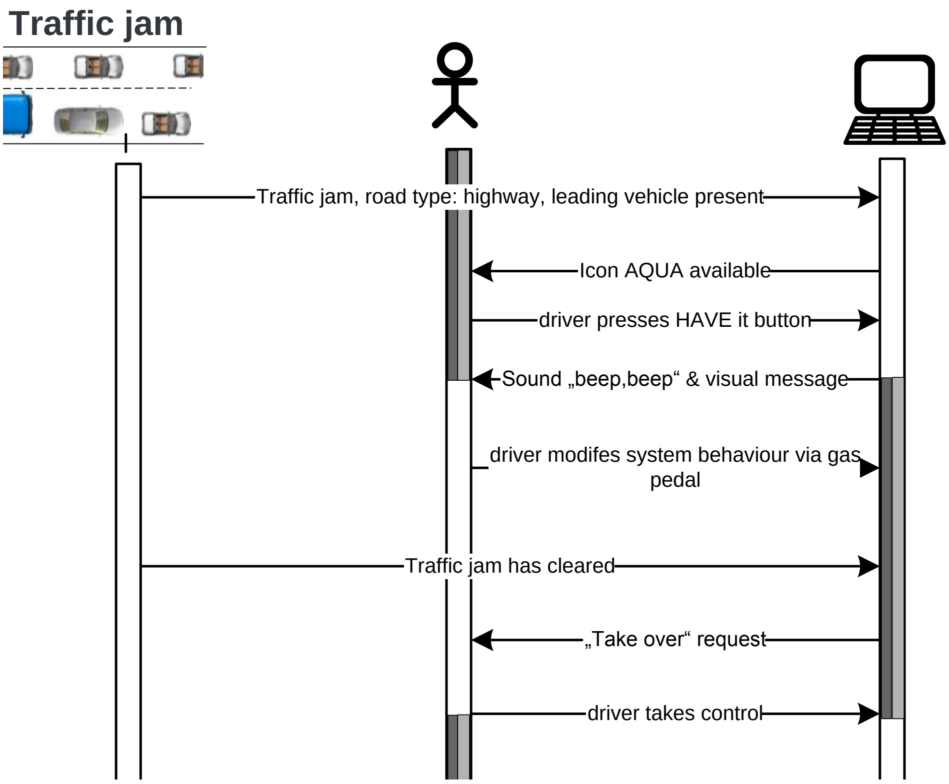


Figure 16: UML-based diagram for one specific transition of the AQUA system

Application description

The application Joint System can be regarded as a second driving instance in parallel to the human driver. It generates a driving motion vector that will be then converted in demand values and sent to the engine, the brake system and the steering actuator in the execution layer of the DLR demonstrator.

3.3.2 Brake-by-Wire Truck

To improve the effectiveness of the braking system and to open new degrees of freedom for automation, a new brake design with an appropriate architecture was developed. The key targets for this vehicle are:

- Shorten the braking distance by up to 15% compared to state-of-the art pneumatic brakes. New actuators are required to achieve this goal (electronic brake actuators EBA for service and parking brake).
- Because the structure and the system have to fulfill the requirements of the ECE 13 regulation, the application must be carried out in a way that provides opportunity to perform the pre-homologation of the structure and system independently. Clarification of questions on pre-homologation of the brake-by-wire system for use on public roads was an important topic addressed in HAVEit (together with TÜV Nord, Germany).

To demonstrate the BbW system, and the communication between the system components (even in case of failure), a Volvo FH12 4.2 truck was equipped. The Haldex EMB system is an electrically controlled brake system for trucks, intended for actuation of electromechanical disc brakes that control the clamp forces by an electric motor. The system has a redundancy in that it has four brake pedal sensors, four sensors on the parking brake handle and there are two communication circuits and two power circuits; see the overall architecture in Figure 17.

The system provides the basic brake system functions such as service braking and parking brake, and automatic blending of brake force from retarders. Further, it has functions for preventing wheel lock during service braking or engine braking or excessive wheel spin during acceleration. There are also directionally stabilizing functions implemented preventing under- and oversteering.

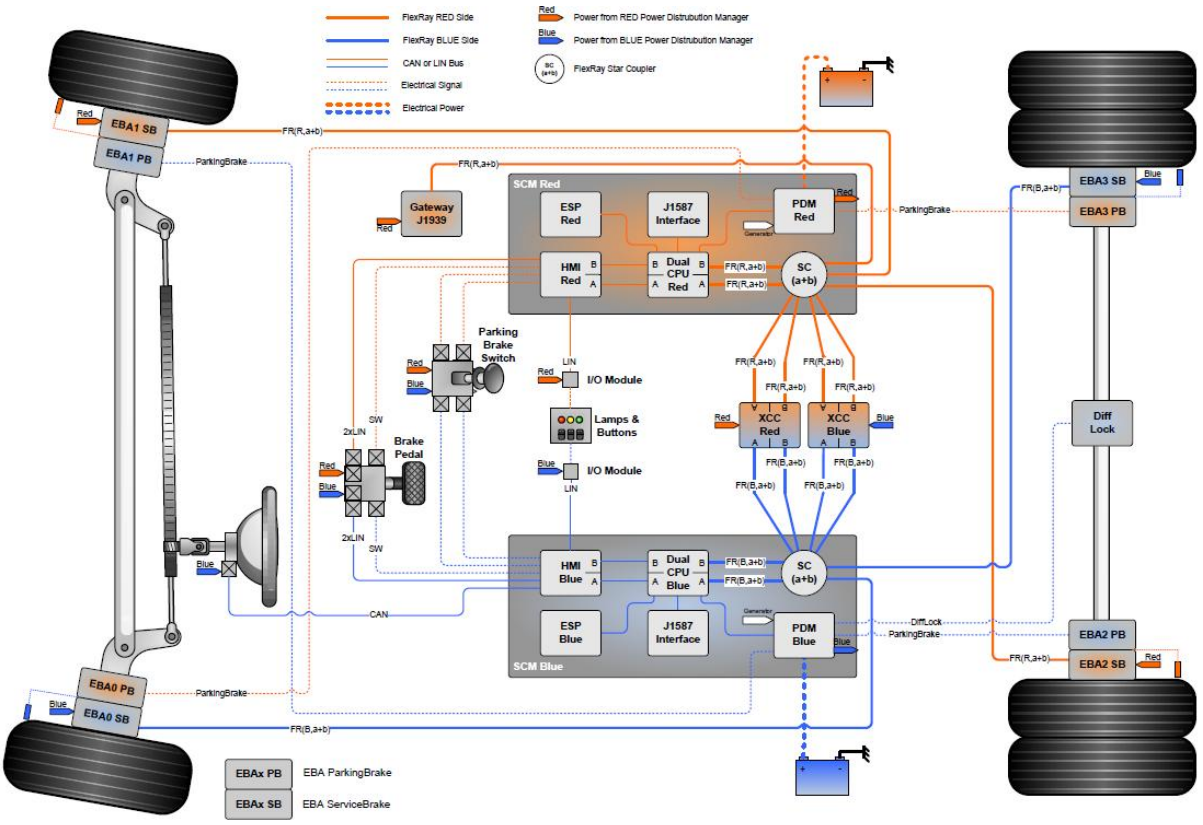


Figure 17: Brake-by-Wire truck architecture overview

3.3.3 Architecture Migration Demonstrator

A vehicle automation system as developed within the HAVEit project can contribute a lot to higher traffic safety, improved traffic efficiency and less environmental pollution. The more HAVEit vehicles are on the road, the more overall benefits will result. The purpose of the Architecture Migration Demonstrator in HAVEit was to show a possible migration path of HAVEit technologies towards mass production. For this purpose the basic idea was to start with an architecture common in series vehicles and to realize as much functions as possible using embedded automotive electronics and to connect them using standard automotive bus technology.

Instead of realizing the HAVEit architecture using prototyping equipment, the goal of this demonstrator was to build the HAVEit Joint System with automotive standard electronics, in this case the Chassis and Safety Controller (CSC) from Continental. There were three major challenges in the development of the Architecture Migration Demonstrator (AMD):

1. Bring the mostly PC developed HAVEit applications to run on the embedded CSC platform
2. Adapt the functionality, bus communication and network structure to cope with the relatively small bandwidth of a CAN bus
3. Set up the communication network compliant with the AUTOSAR methodology

In addition, the common HAVEit architecture had to remain largely unchanged. The further development of the architecture was a time consuming process, since thousands of signals had to be defined, checked and managed. Special software tools were developed to enable at least a partly automated processing of the extensive architectural data.

The base vehicle, a Volkswagen Passat, was equipped with actuators controlled through CAN bus interfaces. Longitudinal and lateral controllers were developed to accept the HAVEit Motion Control Vector to actually drive the car. Two Continental ADAS sensors - a radar sensor and a mono camera - allow the lateral and longitudinal automation within one lane. Accordingly, the set of use cases demonstrated with the AMD during the final event was a subset of the Joint System Demonstrator use cases described in deliverable D11.1.

3.4 Highly automated vehicles for public roads

In addition to the previously described vehicles to validate the HAVEit system architecture, four different highly automated vehicle applications for demonstration of HAVEit functions on public roads were developed. The highly automated driving applications give the driver continuous support to improve road traffic safety and fuel efficiency.

Depending on the situation and the driver load class, different assistance systems are implemented to support and partly substitute the driver or even more to intervene in time when the driver is not fully capable to handle the respective traffic situation before the situation becomes dangerous. The mental driver load is split into classes; overload situations (e.g. intensive situations with increased driver workload like lane changes or roadworks) and underload situations (e.g. monotonous driving situations like in congested traffic).

The vehicles were equipped with numerous sensors to monitor the complete surrounding of the vehicle and the driver. Sensor technologies used are laser, radar, electronic horizon, cameras and also other state of the art sensors. The environment perception information from the co-system is also used to further improve the reduction of fuel consumption and CO_2 in the active green driving application.

The following sub-sections describe the different assistance systems included in this HAVEit application cluster.

3.4.1 Automated assistance in roadworks and congestion

Functions developed for the ambitious goal of automated driving have to have a high level of robustness and must be easy to use for the driver. The goal of the automated assistance in roadworks and congestion (ARC) was to expand highly automated driving in roadwork areas on highways. Entering the road works and driving for longer distance in the road works area is very challenging for the driver. Some drivers even feel fear while driving close to vehicles in the adjacent lane.

The vehicle speed is automatically adapted to the statutory speed limits and the vehicle is kept in safe distance to front and side vehicles and stationary objects. In case the system can not handle the situation, e.g. missing or destroyed lane markings the driver is advised to take over at least part of the vehicle control again. According to the HAVEit approach, a stepwise transfer of the driving task was applied. If the driver does not react properly, the system will decelerate the vehicle according to the situation. In case a very critical situation inside the roadwork occurs (e.g. a vehicle in the right lane coming too close to the own vehicle) due to the low distances to other vehicles and the narrow lanes, the system may have to intervene, e.g. by emergency braking.

An integrated approach of the lateral and longitudinal control is used to adapt the speed and the lateral position of the vehicle. The environmental perception is enhanced in robustness, which is necessary due to the higher transition times from automated to normal driving mode. The driver awareness is included into the control strategy to adapt the automation level to the driver.

Available lateral control algorithms which are based on lane mark detection are very robust already. Also longitudinal control algorithms like ACC work very well for moving obstacles or those who have moved before. Problems occur, if lane markings, moving and standing obstacles interfere with each other. To react properly, detailed information about the vehicles' environment and the driver awareness has to be gathered and the control strategies have to be adapted to the situation.

In situations out of the system boundaries, the control will be handed to the driver. If the driver does not react in time, the system will decelerate autonomously. Main differences to available lateral and longitudinal control systems are:

- System adaptation to driver awareness
- Several automation levels
- More Use-Cases are regarded
- Detection of generic obstacles
- Detection of speed limits
- Blind spot detection
- Integration of longitudinal and lateral control algorithms
- Integrated warning and intervention concept for longitudinal and lateral automation

3.4.2 Automated queue assistance

Automated queue assistance supports the driver on motorways by integrated longitudinal and lateral control (thus representing a new generation of automated cruise control). In reality this means that the system continuously supports the steering, accelerating and braking of the vehicle. Increased safety is achieved by combining the task of the driver and this support system. The purpose of the automated queue assistance is to relieve the driver of the monotonous tasks associated with driving a truck in low speeds and in congested traffic situations, hence driver-underload situations.

A high level of automation requires a high level of perception around the vehicle. To meet these requirements, the vehicle was equipped with numerous sensors using sensor technologies such as laser, radar and camera to monitor the complete surroundings of the vehicle and the driver of the vehicle.

The level of automated control is continuously adapted based on the state of driver, vehicle and environment. At the highest level of automation, the system autonomously handles steering, accelerating and braking to keep the vehicle in the correct lateral position in the lane, at desired speed or at a safe distance to other vehicles. In particular, this is possible at low speeds and in dense traffic down to a complete vehicle stop. The system supports the driver also in non-congested situations and at higher speeds and here the level of automation will be determined by the performance of the perception system and user acceptance. The system continuously assesses its own ability to handle the situation and to transfer the control back to the driver if needed. Special attention is paid to the development of safe transitions between semi-autonomous control and driver control of the vehicle. Strategies were developed to always keep the driver in the loop.

3.4.3 Temporary autopilot

The Temporary Auto Pilot (TAP) is a passenger car application which supports the driver on motorways and motorway similar roads with different levels of automation in longitudinal and lateral control of the vehicle at speeds between 0 and 130 km/h. The Temporary Auto Pilot is fundamentally intended to support the driver in monotonous traffic situations like traffic jams or monotonous long distance driving from A to B where she/he can experience work underload which can lead to a lack of focus and increased accident risk.

In the highest level of automation steering, accelerating and braking is completely managed by the Temporary Auto Pilot. This highest level of automation is offered to the driver only if certain boundary conditions are fulfilled, e.g. the vehicle is driving on a motorway with less than 130 km/h. If these boundary conditions are not fulfilled, a lower level of automation will be offered to the driver.

“Lower level of automation” means that the driver can activate other assistance systems like Adaptive Cruise Control and/or Heading Control. This concept guarantees the upward compatibility of existing driver assistance systems with the Temporary Auto Pilot. On the other hand the concept of a stepwise change on the automation level is a promising approach for a transparent system, where the driver is always aware of what he is responsible for.

For every level of automation the driver is always capable to override the Temporary Auto Pilot. The driver is still responsible for her/his car and therefore has to monitor the system. The Temporary Auto Pilot is not capable of managing every possible critical situation. Therefore, the driver has to be attentive and ready for an intervention.

To increase the safety and reliability of the system at the highest level of automation special measures are taken. These measures are validation of sensor information, driver state monitoring and emergency braking function.

The validation of sensor information reduces the risk of false information about the environment which might lead to false decisions. The driver state monitoring reduces the risk in situations where the driver does not comply with his duty to be attentive and to be ready for an intervention. The emergency braking function reduces the risk of situations where an intervention of the driver is necessary but the driver does not react appropriately.

The TAP consists of five major functions:

- Pilot Function: automated driving on the freeway (automated longitudinal and lateral control), while driving hands-off
- ACC (Adaptive Cruise Control): assistance in longitudinal control only
- HC (Heading Control): assistance in lateral control only
- Emergency brake
- Minimum Risk Maneuver

3.4.4 Active green driving

The active green driving application differs from the other applications in HAVEit because the driver always controls the vehicle. Nevertheless, a suitably designed *Human-Machine Interface* (HMI) is used to coach the driver with the aim to reduce the fuel consumption and pollutions. In the long-term vision of highly automated driving, fuel reduction strategies will be important inputs to the autonomous driving system in hybrid vehicles.

The intention of the active green driving application is to foresee the vehicle movement over the next period of time with the aim to reduce the fuel consumption. In general, it is possible to save fuel also in vehicles with conventional powertrains by predicting the future vehicle movement; a skilled vehicle driver selects proper gears based on visual traffic information intercepted by the eyes. In hybrid vehicles, there is an even larger potential in saving fuel since there is more flexibility in the control task of the hybrid powertrain. Besides selecting proper gears, the internal combustion engine (ICE) can be switched either on or off. Furthermore, there must be the choice of how to construct the partition between the available power sources, namely the ICE and the electrical machine.

The green vehicle application was demonstrated in a Volvo bus consisting of a parallel hybrid. It has been demonstrated that this powertrain concept can reduce the fuel consumption considerably. The demonstrator bus in the green vehicle application was equipped with sensors covering the area in front of the vehicle to foresee the vehicle movement in the longitudinal direction. The Volvo bus in HAVEit also was equipped with several perception sensors to foresee the vehicle movement. The preview information from the sensors was used in the powertrain controller where decisions such as powersplit between Diesel and electric machine and Diesel on/off are determined. With the preview information it is possible to plan ahead and to better make the decisions mentioned above.

There was a great challenge in the green vehicle application on how to use the new perception sensor information to improve the fuel consumption. The research performed within the green vehicle application pushed the current state-of-the art hybrid powertrain solutions to a further level, paving the way for autonomous driving including fuel consumption aspects.

For evaluation pruposes only, the green driving application can be turned on or off. When it is off, the new sensor information will not affect the powertrain controller and the selection of gears, the on/off behavior of the ICE and the split between the powertrain power actuators will be based on the usual hybrid powertrain sensor information. However, when it is on, the new sensors and the prediction capability will affect the control and reduce the fuel consumption in different scenarios, as described section 3.2.4.2 of deliverable D11.1. An HMI supports the driver with information, which - if carried out - lead to lower fuel consumption and discharge of CO_2.

4 Horizontal Challenges: Safety Architecture Implementation

In this section, results achieved within the frame of HAVEit's horizontal challenges in cluster 2 are described in detail. To summarize:

- Challenge 2.1: Safety relevant ECUs
- Challenge 2.2: FlexRay and CAN communication
- Challenge 2.3: Actuators for by-wire-systems

4.1 Challenge 2.1: Safety relevant ECUs: XCCs and CSCs

Within the vehicle architectures of HAVEit (see Figure 6) we distinguish 1E/1M architectures (means one electronic system and one mechanical system) and 2E architectures (two electronic systems; no mechanical system). The first one constitutes the current solution of the automotive industry as an answer for the question how to provide computer based vehicle control. 1E/1M architectures provide the benefit that the E-system is only an add-on to the particular socially accepted M-system which usually enables the driver to control the vehicle even in cases where the E-system is lost. Nevertheless, especially for advanced driver assistant systems (ADAS) which allow the driver to get out of the loop at a certain level, the malfunction of the E-system needs to be safely detected to prevent the vehicle from getting out of control on one side and to request the driver via the particular HMI to take over control on the other side. For the HAVEit demonstrator vehicles this particular need is satisfied by the fail-safe CSC ECU.

With growing complexity of the automation level, fail-safe behaviour might not be enough. This is the case for specific vehicle systems which compulsory need to remain computer controllable, even after a first fault (e.g. to satisfy the requested safety) or vehicles being autonomously controlled (fully automated, driverless) or with "by-wire"actuators without any mechanical fall back solution. This is only possible, if each required component is available at least twice (including the E-system). Within HAVEit, the XCCs only constitute the hardware fundament for provision of an efficient 2E architecture. This needs to be completed by the redundancy management software which provides the required "intelligence" to manage the redundant units.

All in all the CSC ECU and the XCC ECU are performing quite similar from the hardware point-of-view. In general, both ECUs apply redundancy – mainly a self-checking pair of CPUs – for safely detecting failures. The main difference between them is the strictness with which the particular redundancy is applied. The redundancy of the XCC is applied in a stricter manner than the redundancy of the CSC. In consequence, the XCC constitutes the safer but more expensive solution, while the CSC provides a suitable solution for near future serial production: providing necessary safety functionality for functions where the driver can take over in case of failure and therefore fail silent is an adequate strategy. So for each single system, which is intended to be built within HAVEit an analysis has been performed.

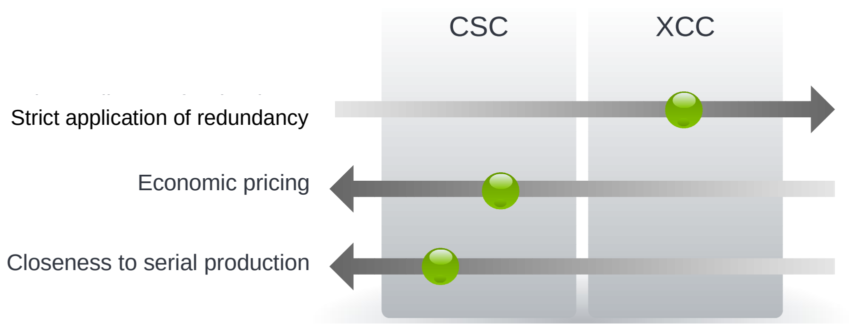


Figure 18: Advantages of the control-computers

4.1.1 Chassis and Safety Controller

System overview

The CSC electronic control unit provides scalable microcontroller performance in order to implement different vehicle functions concerning chassis control, driver assistance systems as well as safety functions. Communication is provided optionally using FlexRay or/and CAN. Additionally, the CSC ECU acts as a scalable integration platform for inertial sensors for vehicle dynamics. Figure 19 shows the principal assembly of the CSC electronic control unit. It mainly consists of following components.

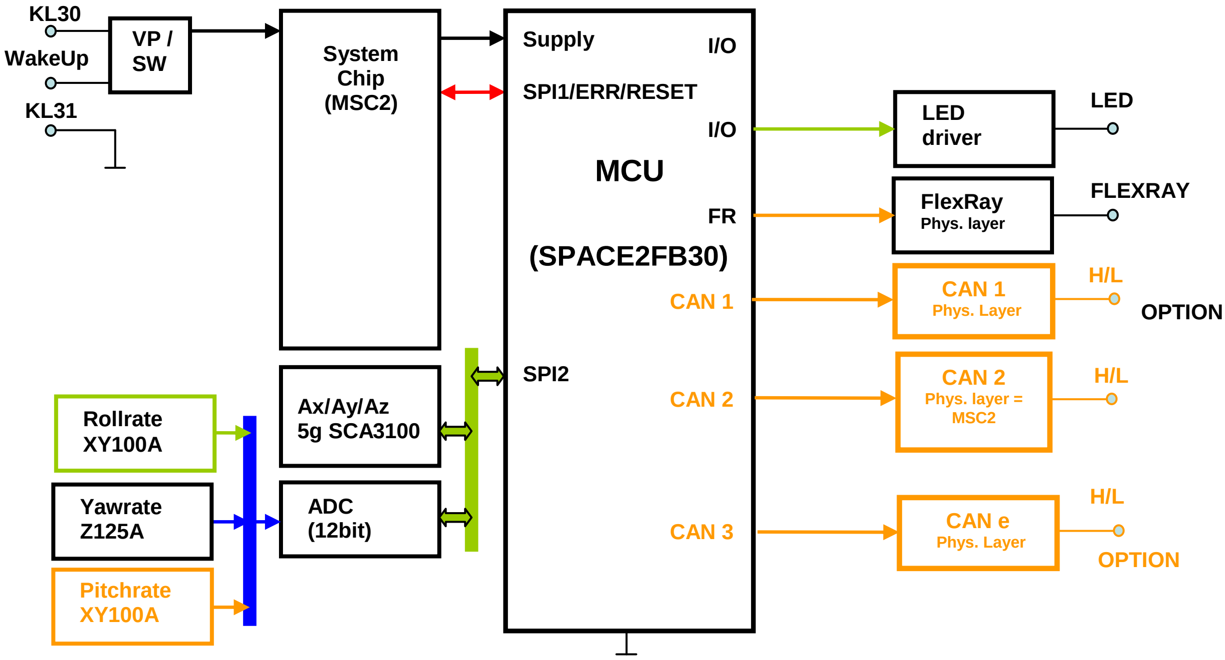


Figure 19: Hardware architecture of the CSC

- The microcontroller MCU with redundant cores and on chip memory executes the application software.
- The system chip MSC2 provides power supply and safety features.
- Three CAN communication interfaces and the FlexRay interface connect the CSC electronic control unit with the other systems in the vehicle.
- The following sensors are part of the hardware, but are not used in the HAVEit project.
    - The sensors for yawrate, rollrate and pitchrate allow for sensing the angular rate on all three axes, x- y- and z-axis.
    - The acceleration sensor allows for sensing the acceleration on all three axes, x- y- and z-axis.

They main components together with their placement on the PCB board are illustrated in Figure 20. The CSC is based on Motorola microcontroller SPACE 2FB30-M. Main features of this ECU are 3 MB flash memory, 160 kB RAM. The ECU allows maximum clock frequency of 150MHz @T_a = 105°C.

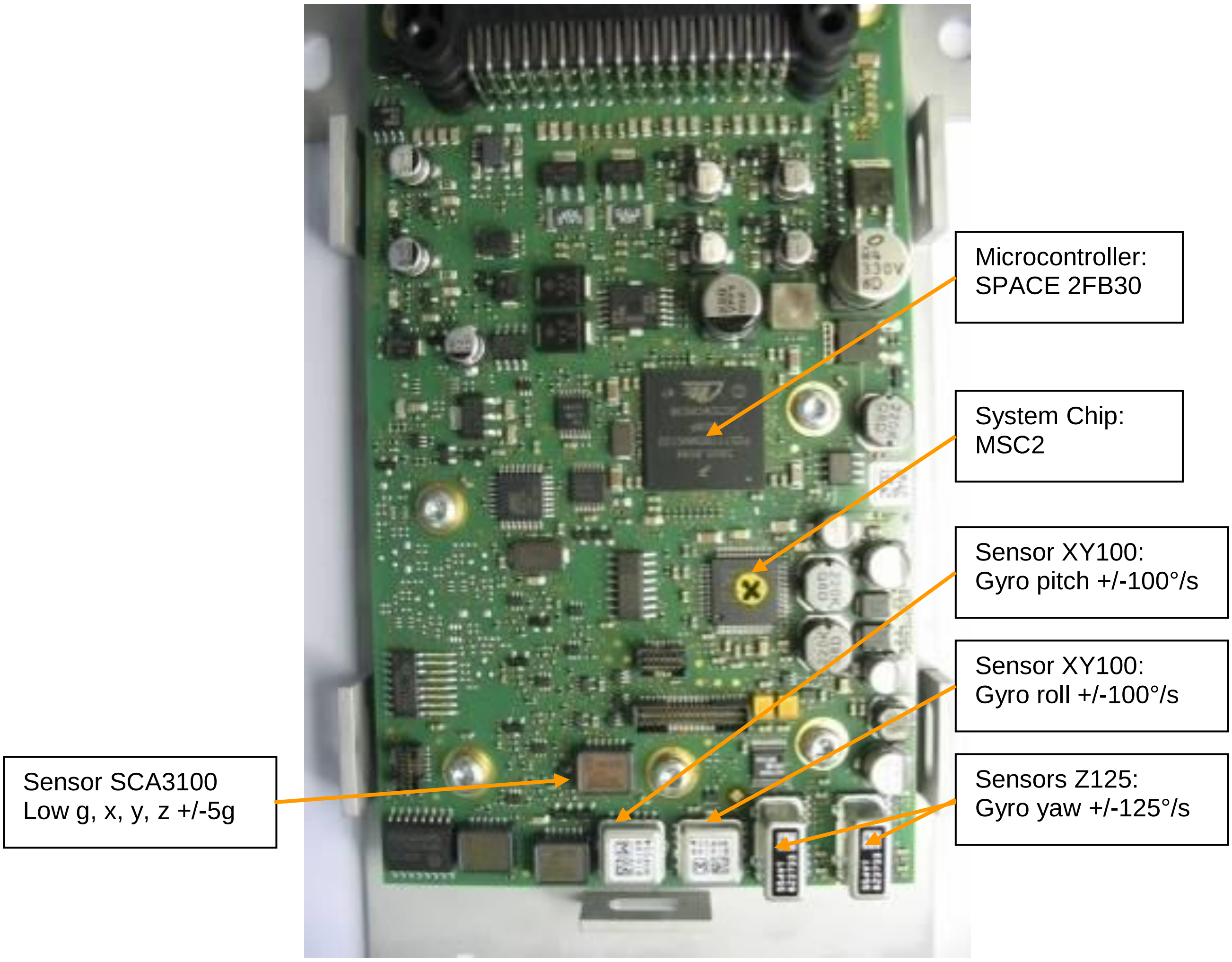


Figure 20: Main components of the CSC

The system chip MSC2 B.0 is an application specific integrated circuit that provides several features for the function of the electronic control unit. The main purposes are providing power supply for the microcontroller and sensors, and monitoring of common cause failures of the microcontroller.

Three CAN communication interfaces are available on the printed circuit board and accessible via the interface connector. Initially, the system architecture was specified with two high speed CAN interfaces. Discussions about the interactors and first estimations of bus load, however, showed that there is a necessity for a third CAN interface. This additional requirement has been taken into account during the development of the final version of CSC hardware for the HAVEit project.

The housing consists of an aluminium ground plate which provides mechanical fixation to the vehicle body and is also used for thermal power dissipation. The printed circuit board is fixed with screws to this metal plate. The cover is made of plastic material and is snapped in to the fixations of the ground plate. The cover is not intended to withstand mechanical stress. The CSC fulfills protection degree IP4/0, covers an ambient temperature range from -40°C - + 85°C (assuming mounting recommendations are adhered) and humidity up to 70% rel.

The CSC ECU acts as a scalable integration platform for safety improving functions. It provides scalable microcontroller performance in order to implement different vehicle functions concerning chassis control, highly automated vehicle applications as well as safety functions. Furthermore, the ECU measures the vehicle dynamics like turn rates and acceleration around all axes. CAN or FlexRay interfaces are available for communication with other systems in the vehicle.

The electronic design of the CSC follows ASIL D specification in order to fulfil safety requirements.

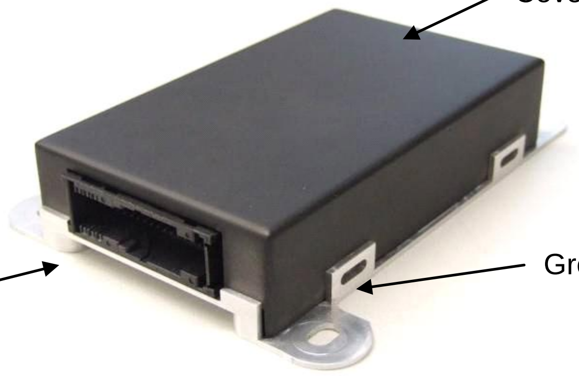


Figure 21: Housing angular view

AUTOSAR concept and methodology

AUTOSAR is used as a common software infrastructure to achieve the technical goals modularity, scalability, transferability and re-usability of functions. The following diagram (see Figure 22) shows a reduced overview of the AUTOSAR approach:

- AUTOSAR SW-C

  The AUTOSAR Software Components (AUTOSAR SW-C) encapsulate an application which runs on the AUTOSAR infrastructure. The AUTOSAR Software Components have well-defined interfaces, which are described and standardized within AUTOSAR.

- SW-C Description

  For the interfaces as well as other aspects needed for the integration of the AUTOSAR Software Components, AUTOSAR provides a standard description format, i.e. the Software Component Description (SW-C Description).

- Virtual Functional Bus (VFB)

  The VFB is the sum of all communication mechanisms (plus some interfaces to the basic software) provided by AUTOSAR on an abstract, i.e. technology independent, level. When the connections between AUTOSAR Software Components for a concrete system are defined, the VFB will allow a virtual integration of them in a quite early development phase.

- System Constraint and ECU Descriptions

  In order to integrate AUTOSAR Software Components into a network of ECUs, AUTOSAR provides description formats for the complete system as well as for the resources and configuration of the single ECUs. These descriptions are kept independent of the Software Component Descriptions.

- Mapping on ECUs

  AUTOSAR defines the methodology and tool support needed to bring the information of the various description elements together in order to build a concrete system of ECUs. This includes especially the configuration and generation of the Runtime Environment and the Basic Software on each ECU.

- Runtime Environment (RTE)

  From the viewpoint of the AUTOSAR Software Component, the RTE implements the VFB functionality on a specific ECU. The RTE can however delegate this task to the Basic Software as far as possible.

- Basic Software: The Basic Software provides the infrastructural functionality on an ECU.

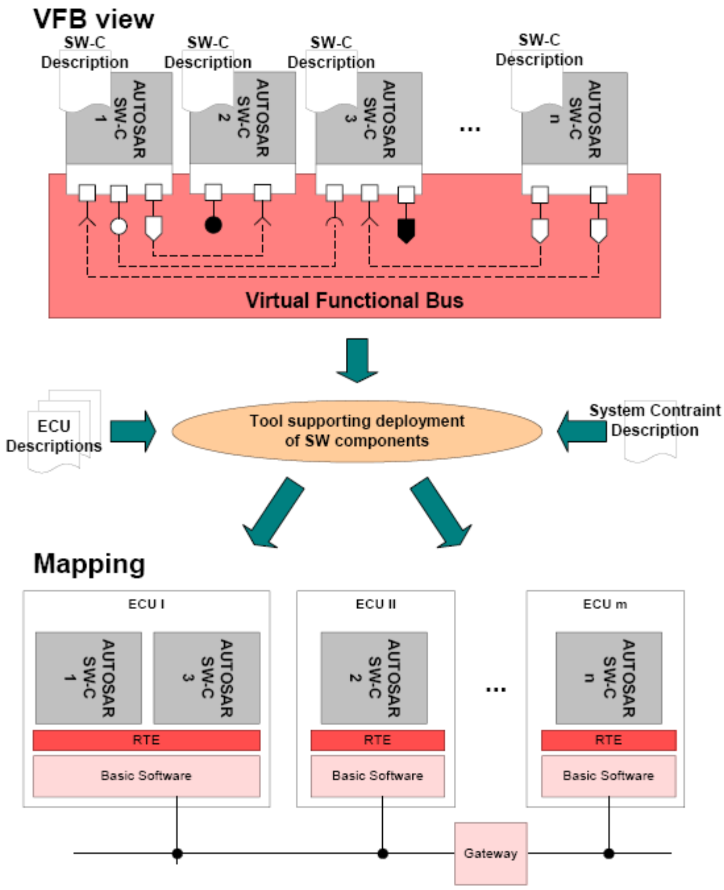
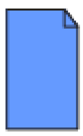


Figure 22: Basic AUTOSAR approach

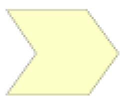
AUTOSAR requires a common technical approach for some steps of system development. This approach is called the “AUTOSAR Methodology”. AUTOSAR describes the methodology using the Software Process Engineering meta-model, or SPEM for short. In the context of the AUTOSAR methodology only a very small subset of SPEM is actually used. The following subsections describe the appropriate graphical notation.

Work Product



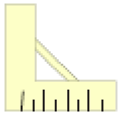
A work product is a piece of information or physical entity produced by or used by an activity. Several specific kinds of Work Products are defined for the AUTOSAR methodology, e.g. XML-Document, c-Document (for source files in the language C), obj-Document (for object files), or h-Document (for header files).

Activity



An activity describes a piece of work performed by one or a group of persons.

Guidance



Guidance elements are associated with activities and represent additional information or tools that are to be used to perform the activity.

AUTOSAR layer model

In AUTOSAR the ECU software is abstracted and sub-classified as basic software (BSW) layer, runtime environment (RTE) and application layer (see Figure 23).

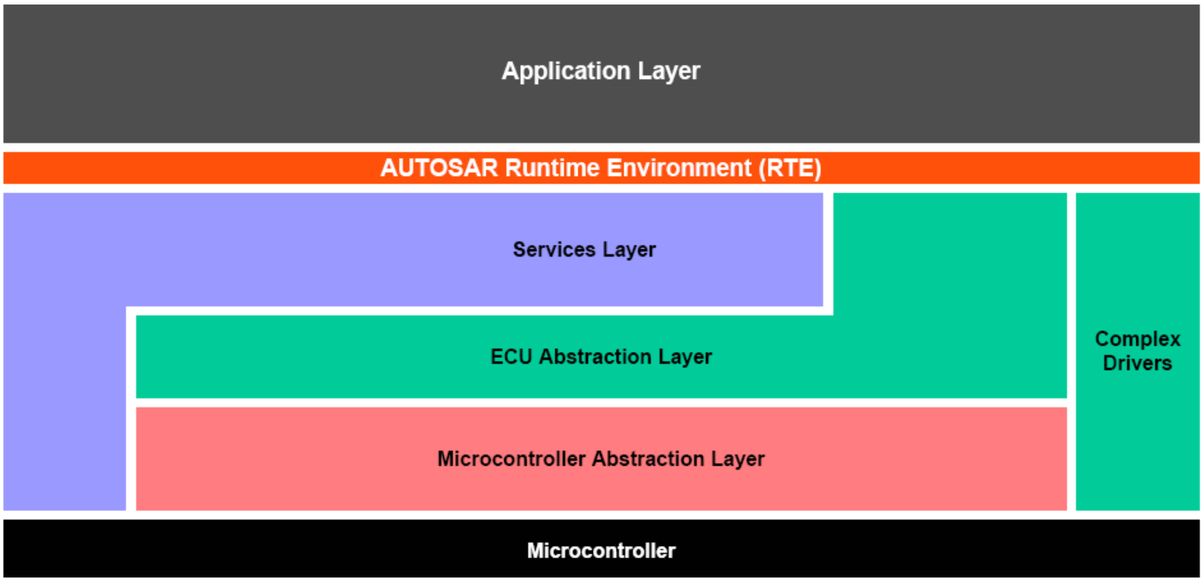


Figure 23: AUTOSAR layered software architecture

The Microcontroller Abstraction Layer (MCAL) is the lowest software layer of the Basic Software. It contains internal drivers, which are software modules with direct access to the µC internal peripherals and memory mapped µC external devices.

The ECU Abstraction Layer interfaces the drivers of the Microcontroller Abstraction Layer. It also contains drivers for external devices. It offers an API for access to peripherals and devices regardless of their location (µC internal/external) and their connection to the µC (port pins, type of interface).

The Services Layer is the highest layer of the Basic Software which also applies for its relevance for the application software: while access to I/O signals is covered by the ECU Abstraction Layer, the Services Layer offers

- Operating system functionality
- Vehicle network communication and management services
- Memory services (NVRAM management)
- Diagnostic Services (including UDS communication, error memory and fault treatment)
- ECU state management

The RTE is a layer providing communication services to the application software (AUTOSAR Software Components and/or AUTOSAR Sensor/Actuator components). The software architecture style changes from “layered“ to “component style“ above the RTE. The AUTOSAR Software Components communicate with other components (inter and/or intra ECU) and/or services via the RTE.

Model based development

A couple of steps are required in terms of the AUTOSAR methodology to start the development of an application software component and to integrate it later into the system.

The goal for the workflow summarized below is a model based application development on an embedded ECU. The following picture (see Figure 24) shows the steps from a software component design to an executable which can be flashed to the ECU.

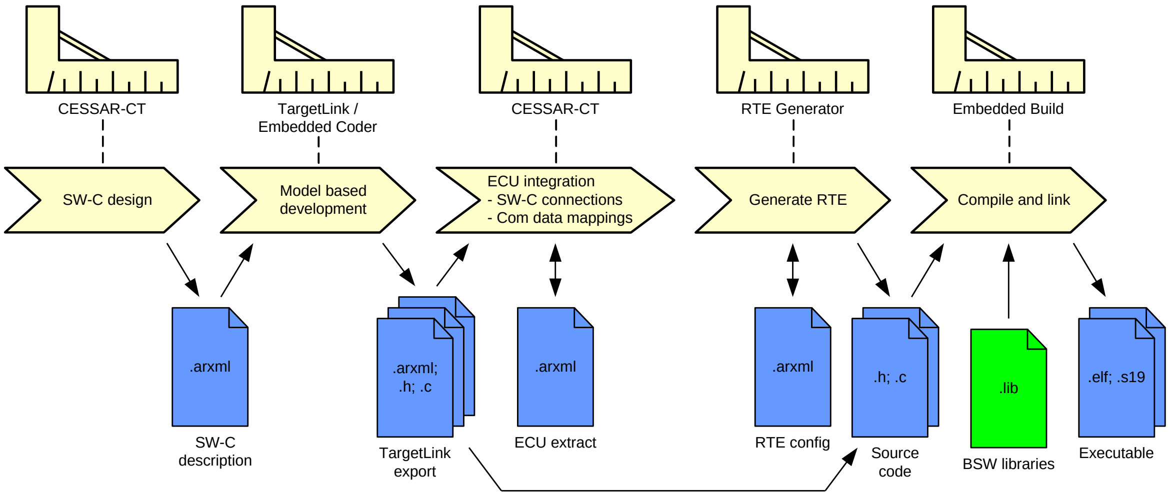


Figure 24: Development workflow

- Workflow step 1: SW-C design

  In AUTOSAR the complexity of the application software is encapsulated in Software Components (SW-C). Accordingly, the first activity is the software component design with CESSAR-CT. An AtomicSoftwareComponent (SW-C) must be defined with all interfaces, ports and data types for an application software. For the model based development the SW-C description must be exported. For this exchange AUTOSAR has defined a meta-model based data exchange format. This data exchange format is commonly accepted in the automotive industry.

- Workflow step 2: model based development

  The model based development starts with an import of the software component description into Matlab/Simulink. The comprehensive AUTOSAR XML format is used for exchanging work products between CESSAR-CT and Matlab/Simulink. Now the development of the software component internals can be done within the Simulink model. At the end the ARXML description of the software component must be exported and the C code for the model must be generated (for example with the Embedded Coder).

- Workflow step 3: ECU integration

  In this step the ARXML description of the software component must be imported to CESSAR-CT and the system (VFB) integration must be done. For the import the AUTOSAR model files (*.arxml) must be copied to the CESSAR-CT project and renamed to *.ecuextract. Afterwards a refresh of the CESSAR-CT navigator is necessary. Now the different SW-Codes have to be connected with each other. Also the communication mapping must be done. All runnables within the ECU integration must be mapped to the OS tasks. For the mapping of the runnables or for the communication it could be necessary to update the OS or COM configuration.

- Workflow step 4: RTE generation

  This activity generates the run time environment (RTE) of an ECU. Before the RTE is generated, a validation with RTE validator is strongly recommended.

- Workflow step 5: embedded build

  The last step is the embedded build. Within this step the new SW-C and the generated RTE must be added to the embedded build environment. This step involves compiling the code and linking everything together into an executable. At the very end the output of the embedded build has to be flashed with the download tool to the ECU.

In HAVEit AUTOSAR methodology has been used to implement the Joint System to several CSCs provided by Continental. The driver state assessment was developed, realised and tested by WIVW, before complete units were delivered to the partners. The other components of the Joint System were implemented and tested by Continental, IFSTTAR and DLR. The Architecture Migration Demonstrator features the whole Joint System implemented on CSCs using AUTOSAR.

4.1.2 XCC

System overview

The XCC is a fully redundant ECU (see Figure 25). It is built of two similar lanes, each constituting an independent computer by its own. The two lanes of an XCC are mainly applied to realise redundancy. Each operation is performed twice in an XCC: first in lane A and in parallel in lane B. The results of the cross-checking of the operation results are used to detect failures. Having detected a failure, USTUTT’s redundancy management will initiate reaction to this failure accordingly, which might be for example the passivation of the faulty component and the reallocation of its functionality within the processors.

Figure 25 shows the principle assembly of the XCC. The XCC is realised by two independent circuit boards, the so-called »CPU-board« and the »IO-board«. Beside the reduction of the XCC dimensions this design allows a systematic building set for future developments since for example the IO-board can be easily adapted to upcoming needs concerning communication interfaces without a redesign of the overall XCC. Figure 26 shows the XCC boards including their major components.

The particular components being applied for the depicted assembly are the CPU (MPC 5567 running at 132MHz with internal FlexRay communication controller), FR CC (Motorola MFR 4310), 32 kB EEPROM and 2 MB SRAM.

Thie figure below indicates the sandwich construction of the XCCs. Housing’s dimensions are 178 mm x 195 mm x 40 mm (length x depth x height). Since the housing does not provide protection against water or dust, the XCCs need to be installed inside the passenger compartment at a position with lowest possible vibration load.

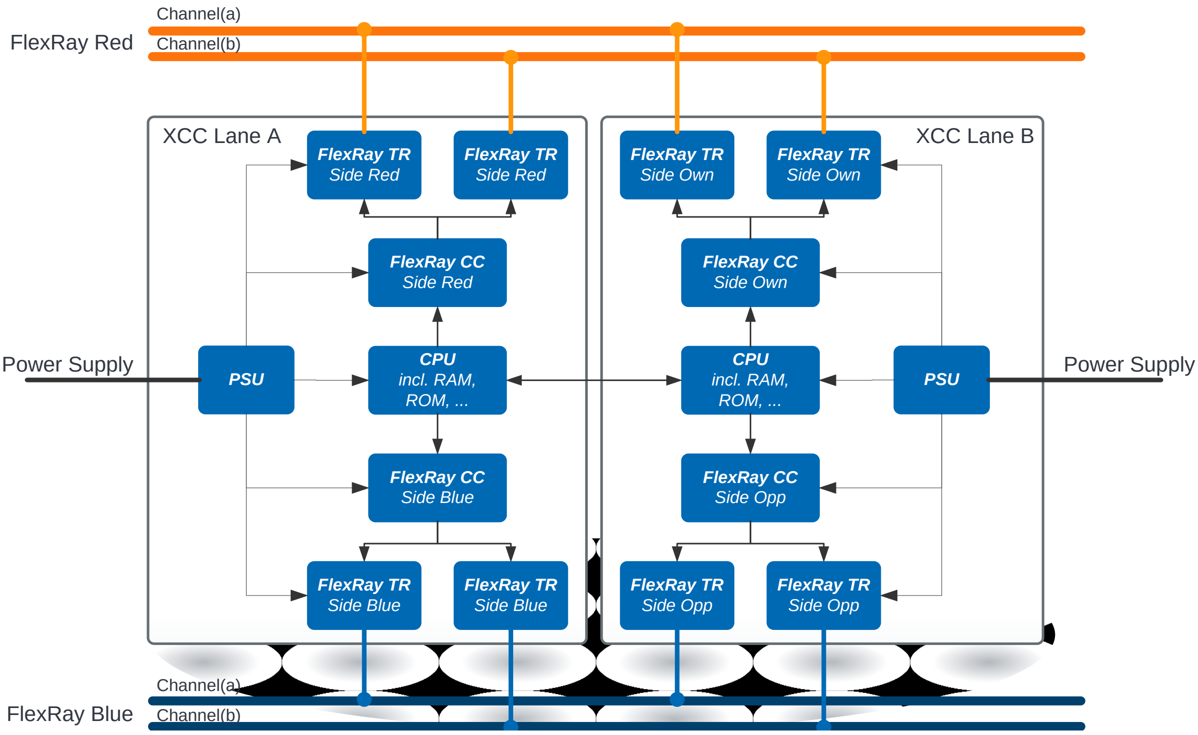


Figure 25: Full-redundant XCC

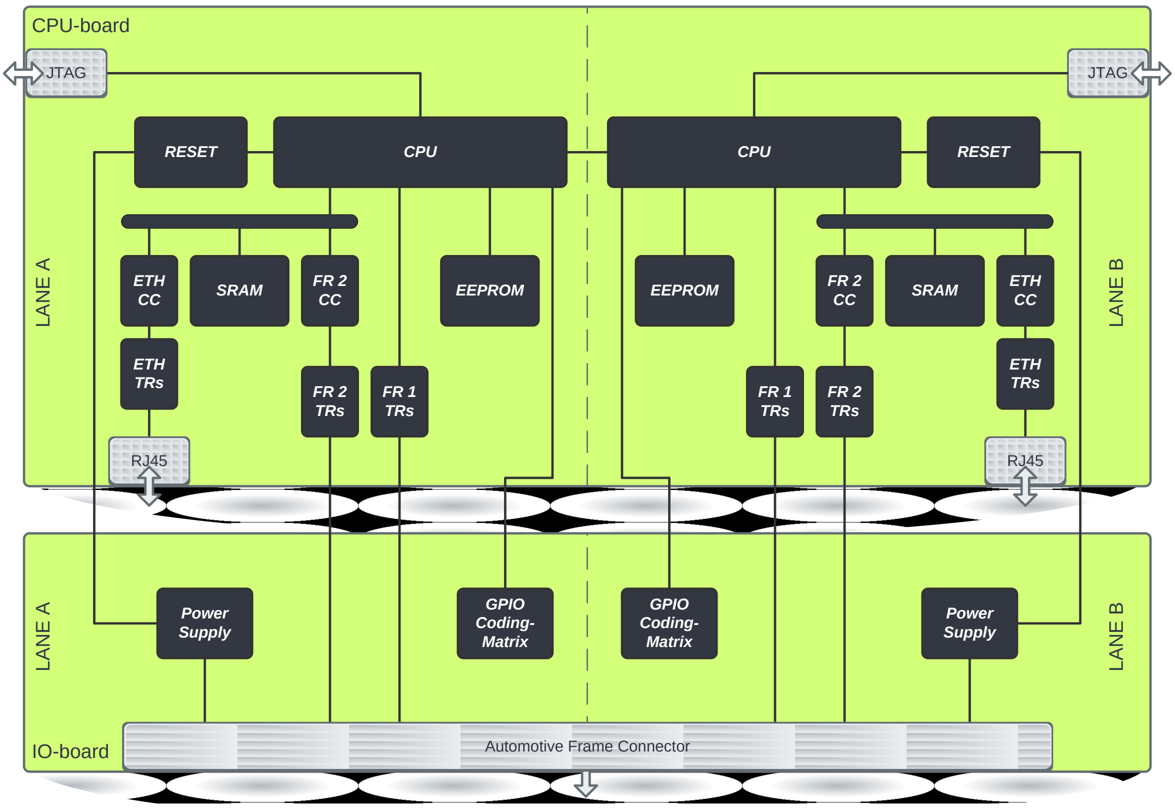


Figure 26: XCC assembly

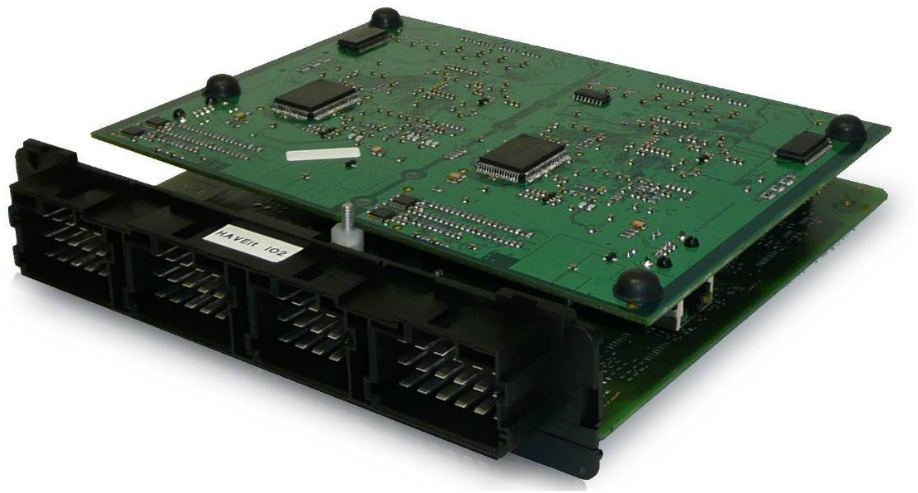


Figure 27: XCC – sandwich configuration (IO-board and CPU board)

Software architecture and configuration process

To establish the 2E architecture into the vehicle market it must obtain at least the same safety standards as the conventional mechanical vehicles. This leads to the consequence, that single electronic components are mostly not sufficient for adequate replacement of the according mechanical components. Therefore, redundancy needs to be applied. By introducing redundancy, new challenging tasks appear by the need of providing consistency and by managing the interaction between the redundancy components.

Within the SPARC project a working 2E platform concept was already shown. The implementation of the SPARC software was almost completely done by hand. To overcome this enormous task, a lot of effort had to be made to provide not only the required functionality but also to avoid errors as best as possible. A lot of time had been spent to find errors and rectify them. Therefore, already within the SPARC project, first approaches of introducing configurability were made, to simplify and accelerate the development of dedicated software modules and – in addition – to increase its quality. The basic idea was the provision of generic software modules, which needed to be verified only once. These software modules could be configured by a predefined set of parameters to fulfil the particular functionality.

Within HAVEit this approach was driven much further. Not only the number of configurable software modules was increased but also a process / tool-chain was developed to systematically collect and manage the required configuration parameters. The procedure for HAVEit is shown in Figure 28.

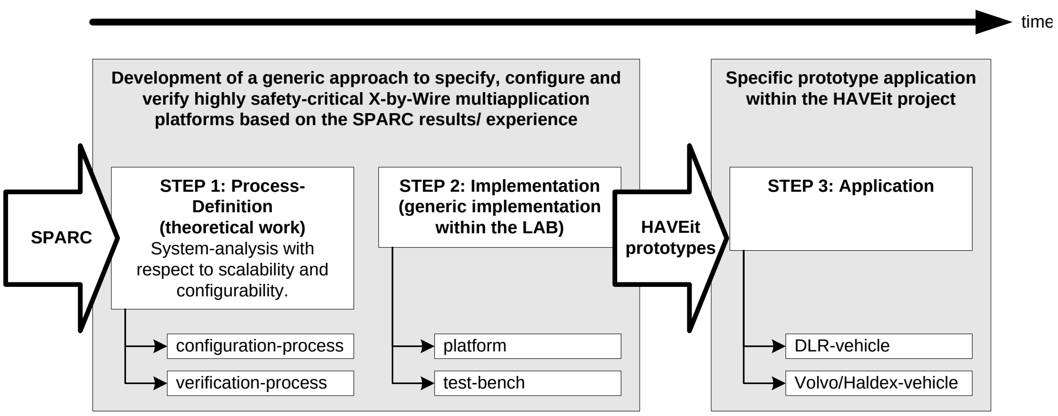


Figure 28: HAVEit development process for 2E systems

By analysis of the software modules being used in SPARC, functionality and configuration parameters for the modules was identified. In addition, effort was made to identify configuration parameters for the USTUTT test-bench software. After that a generic process was defined to manage the systematic collection and usage of these parameters (STEP 1). Subsequently, the mentioned process was implemented by the development of a tool-chain (STEP 2). Finally, the platform software for the vehicles “Joint System Demonstrator, JSD” (WP4100) and “Brake-by-Wire Truck, BbW” (WP4200) was created by applying the tool-chain (STEP 3).

Software Concept

a) *Generic Platform*

The “intelligent” centre of the generic platform (see Figure 29) is the so called central platform core (CPC). Within the CPC the applications (e.g. control laws) are executed and in parallel the redundant components as well as the overall platform behaviour is managed. This CPC is scalable depending on the need for computing power and the request for a specific safety level the whole platform should reach. For the HAVEit project the CPC is assembled out of two XCCs. These XCCs are connected to two asynchronous FlexRay buses, respectively. Each XCC will be synchronised with only one of the FlexRay buses - so called FlexRay (Own).

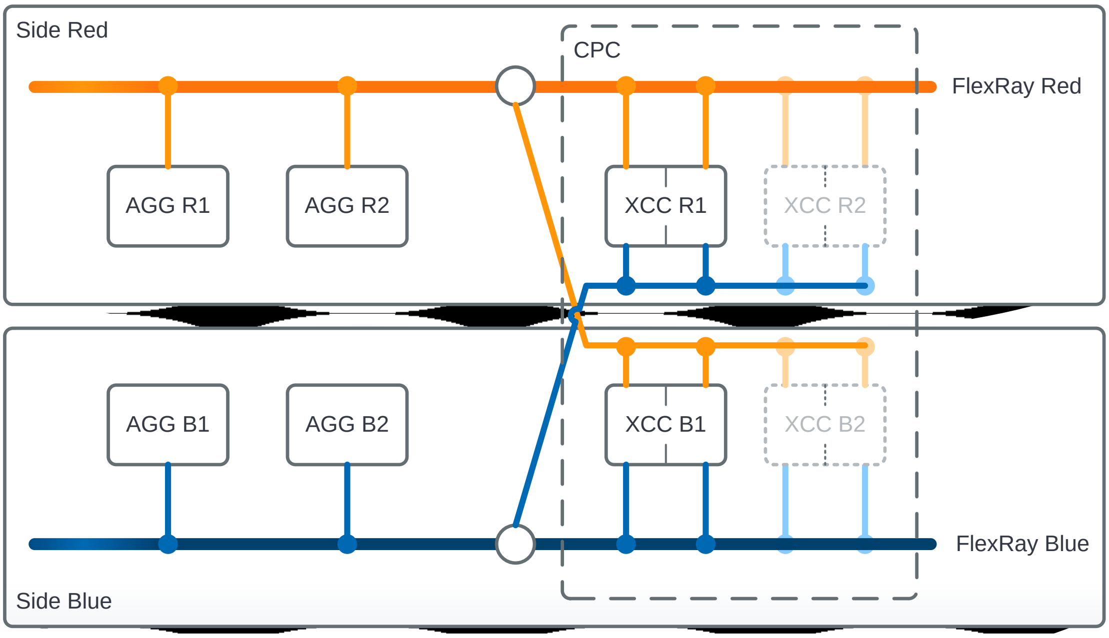


Figure 29: USTUTT’s generic platform

In addition to the CPC the generic platform contains aggregates (AGGs) like sensors, actuators, interfaces, providing the inputs, actuation and outputs required for implementation of the particular system functionalities. Aggregates are connected to only one of the FlexRay buses. Based on the FlexRay bus an aggregate is connected to, or an XCC is synchronously running, respectively, the whole platform can be separated into two sides: side red (associated with the FlexRay bus »Red«) and side blue (associated with the FlexRay bus »Blue«).

b) Software Layers

The software being executed within the XCCs can be illustrated by the different layers as shown below (see Figure 30). The applications (Appli) representing the control laws constitute the topmost layer. Every application is complemented by a related application management (AppMa), which covers the application and system specific management issues. The latter takes influence to the control of the application cycle as well as the administration of related aggregates. So, an individual adaptation to the actual platform conditions is ensured. Each application, together with its related application management is capsulated in an individual partition, separated from all other applications (incl. related AppMa) and the redundancy management. Altogether, the collection of applications and application managements represents the system functionality.

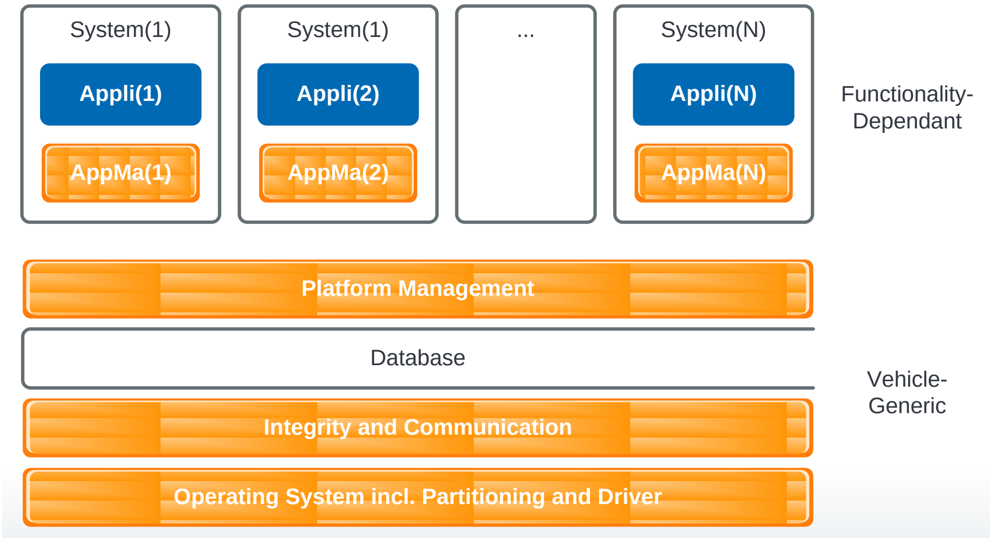


Figure 30: Overview of the software-layers of the redundancy management

The platform management (PlaMa) constitutes the next layer within the XCC software concept. Based on information about the platform and the ability of the XCCs to perform the system functions, the PlaMa determines the overall platform operation mode and the reconfiguration of the “Master”-XCC. Therefore, it is attended by the failure management and consolidation.

The software layers mentioned above are separated from the integrity and communication providing layers by a database. This database embodies a shared memory concept. While the integrity- and communication layer takes care of a consistent and trustable database, the PlaMa and Application layers access this integer data as the basis for their execution. Thus, the database on the one hand represents the detachment of the “simplex world” from the “redundant world”. On the other hand it provides an important step towards the separation of functionality and safety.

Within the integrity and communication layer, the generation of the consistent and reliable database is realized. For this purpose, the integrity and communication layer covers monitoring, voting and communication. The OS-layer provides very basic software functionality of the XCC (operating system) as well as the access to the XCC hardware components (driver / driver handler). Further, the OS-layer provides the partition management which prevents time and memory violation of the mentioned partitions.

c) Management of the Software Layers

Within the HAVEit project, applications (control laws ...) are delivered by the project partners. Therefore, configurability of the applications doesn't make sense and is not pursued. Similar statement holds for the according application management (AppMa). Although these software modules are created by USTUTT, the content is application specific. Altogether, the function dependent layer is excluded from the configuration process. Nevertheless, the AppMa constitutes the interface between the individual "application world" and the generic "platform world". In consequence, it needs to be ensured, that the AppMa provides at least the minimum compatibility with the generic interfaces of the lower layers.

The tasks of the PlaMa - as well as the integrity- and communication layer can be split into the following software modules (see Figure 31): Platform management (PlaMa), Failure management (FailMa), Consolidation module (CoMo), Module input / output (MIO) and Message-Router (MsgRouter).

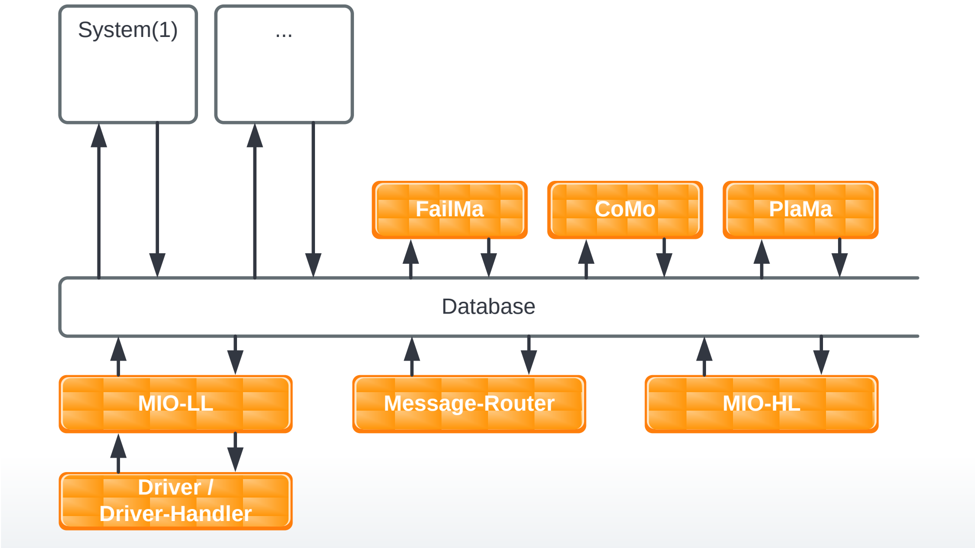


Figure 31: Detailed software-layers of the redundancy management

The voting and monitoring mechanism within the integrity- and communication layer is localized within the MIO. Especially for the FlexRay communication, MIO is used to be separated into a low-level MIO (MIO-LL) and a high-level MIO (MIO-HL). MIO-LL examines the redundant incoming FlexRay messages in whole to provide one single, trustable message for further processing. In addition, MIO-LL delivers failure requests concerning the FlexRay messages and the FlexRay buses themselves.

The message router unpacks the dedicated signals out of the voted incoming FlexRay message and checks, whether the signals are within their specific binary range or not. The other way round the message router packs an amount of signals into a particular FlexRay message which is intended to be sent. Again, this software module also delivers failure request in case of detected problems.

After unpacking the single signals by the message router, MIO-HL creates one single, trustable signal value including information about the signals' availability out of the amount of redundant signals. In addition, failure requests are created in case of detected problems.

Such MIO-LL and MIO-HL together provide the integrity of the incoming data. Therefore, it is indispensable for the MIO to have detailed information about the platform architecture, the particular redundancies and the signal types. Additionally, information about the integrity of the connected modules and the type of incoming message frames are required to configure the MIO respectively.

The failure management collects and interprets the different failure requests raised by the particular software modules. Its output is a clearly reduced collection of failure classes. These failure classes are consolidated between the two lanes of each XCC and timely confirmed within the consolidation module. Outcome is a collection of trustable, inter-XCC-consistent failure classes.

The platform management reconfigures the whole platform as well as the behaviour of the own XCC based on the consolidated failure classes and the outcome of the AppMas.

d) *Configuration Process Concept*

To simplify the configuration of the above mentioned software modules and to enhance the quality of the software development for platforms, a generic configuration process was developed and implemented as meta-tool framework. The basis for any software development is the development of a platform concept (architecture), providing not only the expected functionality, but also the required safety. Based on this platform design, the configuration process tool-chain was applied to implement the according redundancy management software.

To develop a generic configuration process it is necessary to generate a special automatism for the software configuration. Therefore, a standardized framework is necessary to provide a high amount of reusable software parts in case of changed platform architecture. For the different software modules described above there are different developing tools which fulfil the special needs of each software module in the best way.

The platform architecture and the configuration process employed are described in detail in the HAVEit deliverables D21.1 “CSC and XCC hardware development complete” and D21.2 “Software and configuration process concept available”, respectively.

4.2 Challenge 2.2: Communication: FlexRay, CAN, V2V, V2I

4.2.1 FlexRay communication

This section provides a short overview of the principles of FlexRay communication. Within HAVEit, FlexRay was applied in the vehicles of WP4100 and WP4200 for the safety relevant steer-by-wire and brake-by-wire functionalities.

FlexRay is a time deterministic communication bus system with a predefined cycle time. Within the bus system, for each node a specific time slot is defined during the design process, in which the node gets exclusive write access to the bus. These time slots recur cyclical. By this means, the point in time in which a specific message shall be transmitted via the bus can be predicted exactly. Otherwise, therewith the faulty absence of a message can be surely detected. By defining specific time slots for each node, the potential bandwidth of the FlexRay isn’t necessarily fully occupied. Therefore, FlexRay divides its cycle into a static segment (fixed time slots) and a dynamic segment (dynamic time slots), see Figure 32. For the safety relevant functionalities of WP4100 and WP4200 we focus on a safe and deterministic communication. Thus, the dynamic segments were not utilised in HAVEit.

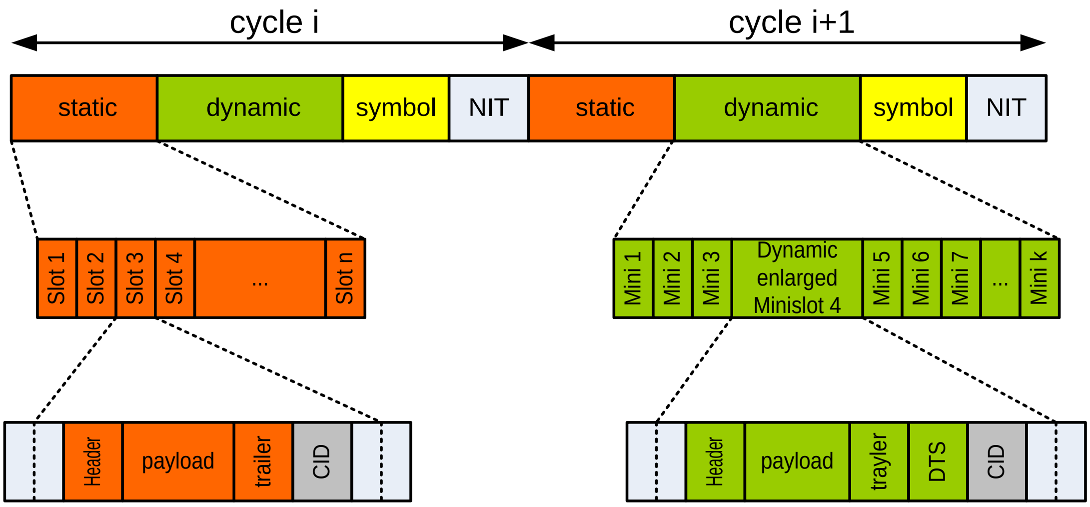


Figure 32: Segments of a FlexRay cycle

The use of time slots for the bus communication presupposes that all attached nodes operate with the same (global) time bases, i.e. synchronous. In order to maintain the basic principle of safe communication, the synchronisation takes place locally in a fault-tolerant manner. Each node adapts its local clock using cyclic time messages, provided via the static segment of a FlexRay cycle. In general, FlexRay communicates via two physically separated channels (channel (a) and channel (b)). In principle, these channels may be used for the simultaneous transmission of two different messages. However, in HAVEit they are used for redundant transmission of identical messages to increase the availability of the communication.

In general, the frames described above are not only sent once in a cycle. Since the FlexRay buses as well as the XCCs are built up redundantly, certain frames are sent by different lanes, via different buses and channels. To enhance readability of the lower defined frames, the principles for redundant communications are examined separately within this section. For more information on the XCCs, please see the HAVEit deliverables D11.2, chapter 5.2.3, and D12.1, chapter 5.2.

<u>Communication principles</u>

XCCs → Aggregate

The FlexRay bus is built by two FlexRay channels. Each aggregate receives its data from each lane of each XCC via both channels of the particular FlexRay bus, the aggregate is connected to. Therewith, each aggregate receives the following eight FlexRay frames:

- XCC1 / Lane(A) → Channel a → Aggregate
- XCC1 / Lane(A) → Channel b → Aggregate
- XCC1 / Lane(B) → Channel a → Aggregate
- XCC1 / Lane(B) → Channel b → Aggregate
- XCC2 / Lane(A) → Channel a → Aggregate
- XCC2 / Lane(A) → Channel b → Aggregate
- XCC2 / Lane(B) → Channel a → Aggregate
- XCC2 / Lane(B) → Channel b → Aggregate

The redundant information is used to enhance the availability of the aggregate's functionality.

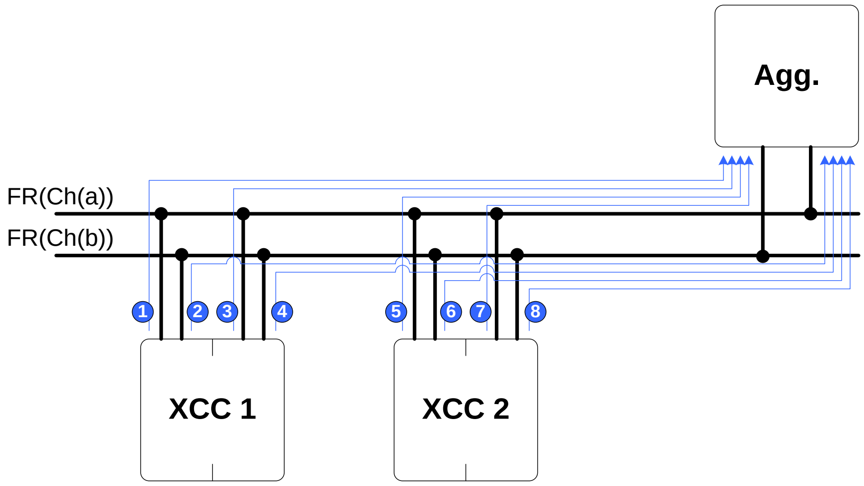


Figure 33: Frames: XCCs → Aggregate

Aggregate → XCCs

Each aggregate sends its data to the XCCs via both channels of the attached FlexRay bus. In case of simplex aggregates, this leads in total to the following two frames:

- Aggregate → Channel a → XCCs
- Aggregate → Channel b → XCCs

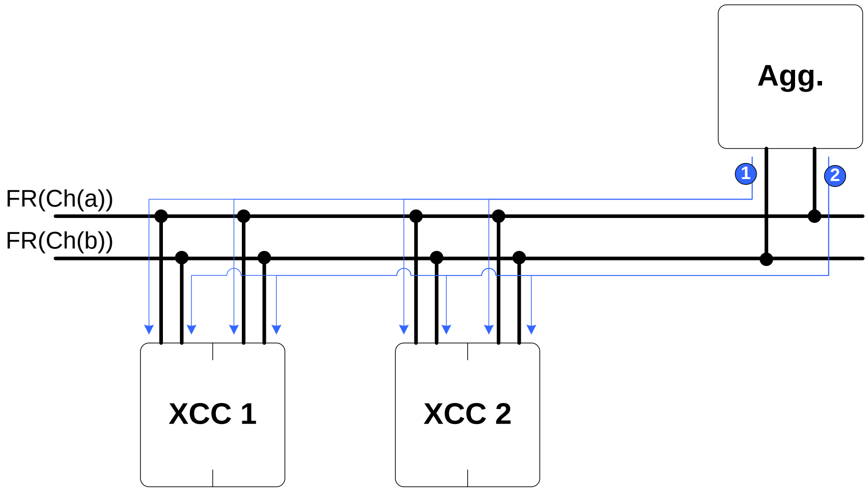


Figure 34: Frames: Simplex-Aggregate → XCCs

In case of duplex-aggregates, each aggregate sends the following four frames:

- Aggregate / Lane(A) → Channel a → XCCs
- Aggregate / Lane(A) → Channel b → XCCs
- Aggregate / Lane(B) → Channel a → XCCs
- Aggregate / Lane(B) → Channel b → XCCs

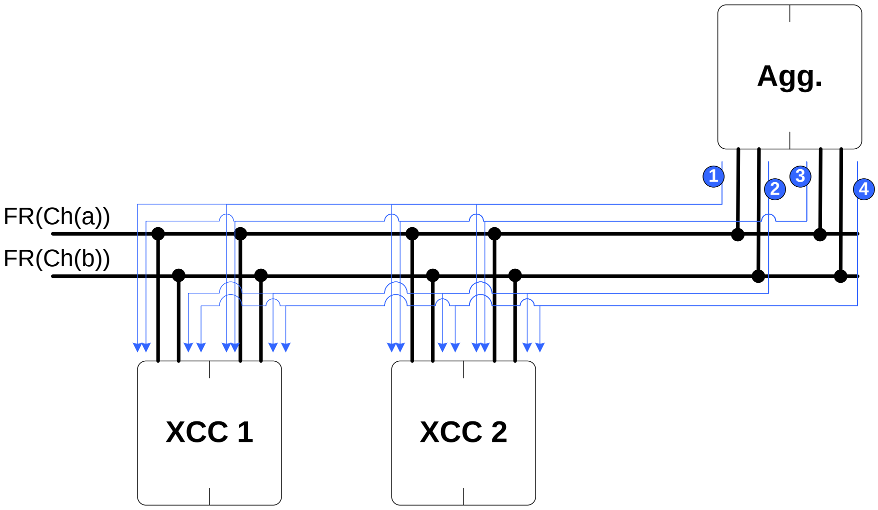


Figure 35: Frames: Duplex-Aggregate → XCCs

XCC → XCCs

Since XCCs are connected with both FlexRay buses, they transpose their data via both channels of both buses (see illustration in Figure 36). Regarding the duplex configuration of each XCC, this leads to eight FlexRay frames:

- XCC1 / Lane (A) → FlexRay(Red, Channel a) → XCC2
- XCC1 / Lane (A) → FlexRay(Red, Channel b) → XCC2
- XCC1 / Lane (A) → FlexRay(Blue, Channel a) → XCC2
- XCC1 / Lane (A) → FlexRay(Blue, Channel b) → XCC2
- XCC1 / Lane (B) → FlexRay(Red, Channel a) → XCC2
- XCC1 / Lane (B) → FlexRay(Red, Channel b) → XCC2
- XCC1 / Lane (B) → FlexRay(Blue, Channel a) → XCC2
- XCC1 / Lane (B) → FlexRay(Blue, Channel b) → XCC2

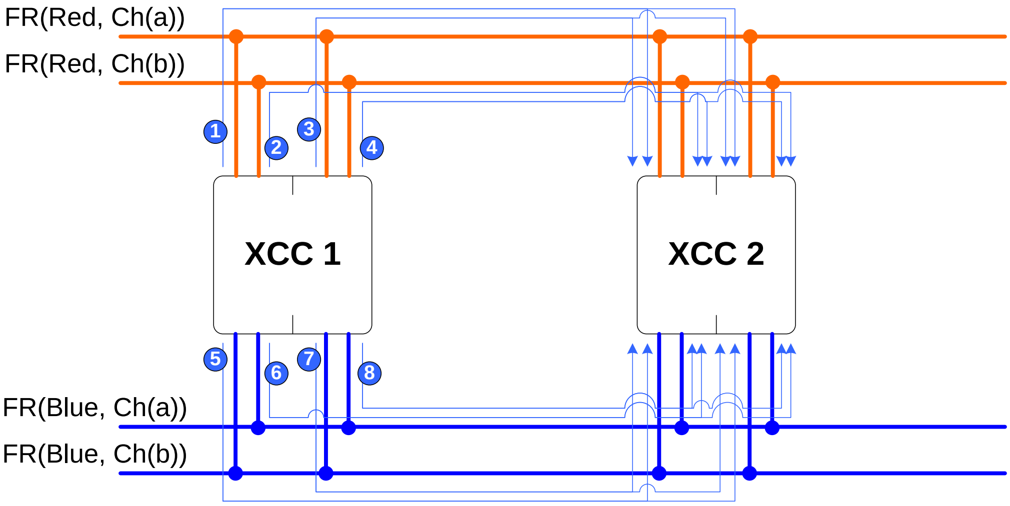


Figure 36: Frames: XCC → XCC

4.2.2 CAN communication

In order to motivate CAN communication issues and the required CAN-FlexRay gateway, again the by-wire actuators in HAVEit (steer-by-wire for the Joint System Demonstrator, WP4100, and brake-by-wire for the BbW Truck, WP4200) are used as an example. Figure 37 presents the communication architecture of the steer-by-wire system. The communication architecture of the WP4100 demonstration vehicle is based on FlexRay; however, since some components are not available with native FlexRay interfaces, connections to these units are realized using CAN. In order to achieve a more deterministic behaviour, the CAN buses are implemented as private point-to-point connections numbered CAN1 to CAN5 in Figure 37.

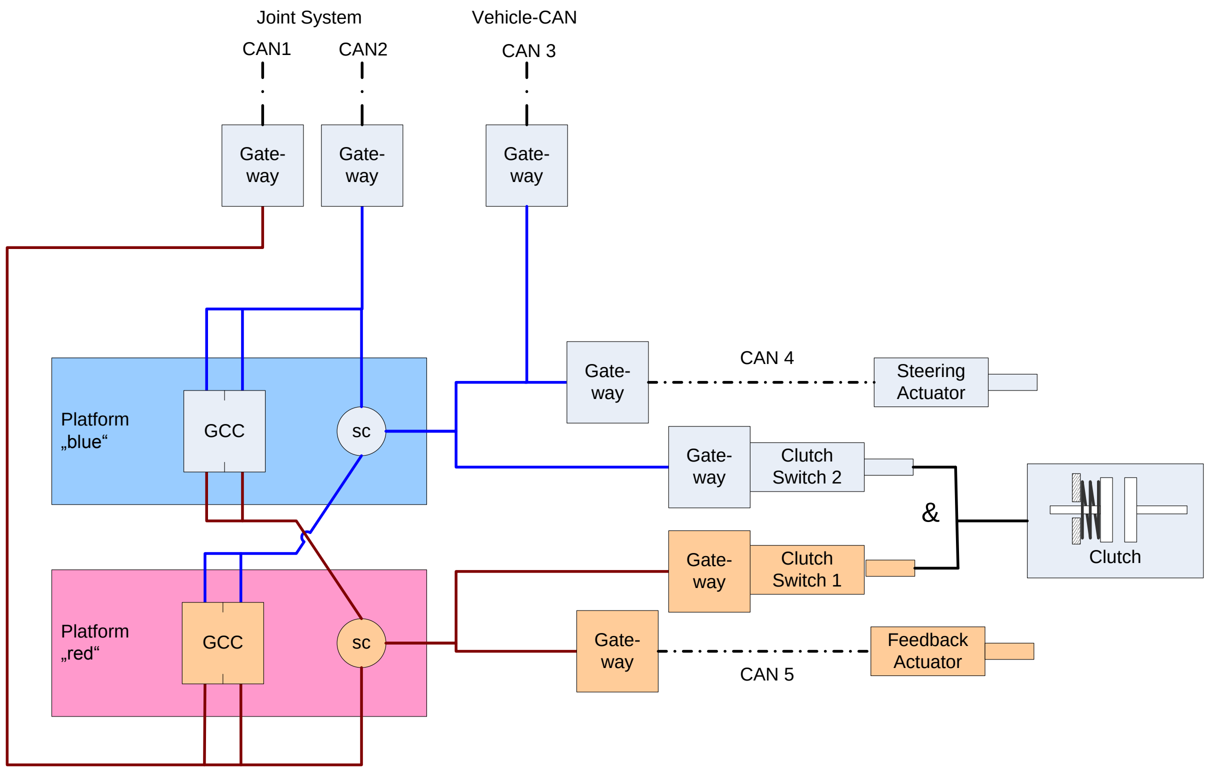


Figure 37: Communication architecture of WP4100 steer-by-wire system

4.2.3 V2V Communication

In HAVEit, vehicle-to-vehicle (V2V) communication was to be applied to the demonstrators of WP5200 (Automated Queue Assistance, AQuA) and WP5400 (Active Green Driving, AGD). The V2V communication module of a HAVEit demonstrator communicates with one dedicated communication module of another demonstrator vehicle.

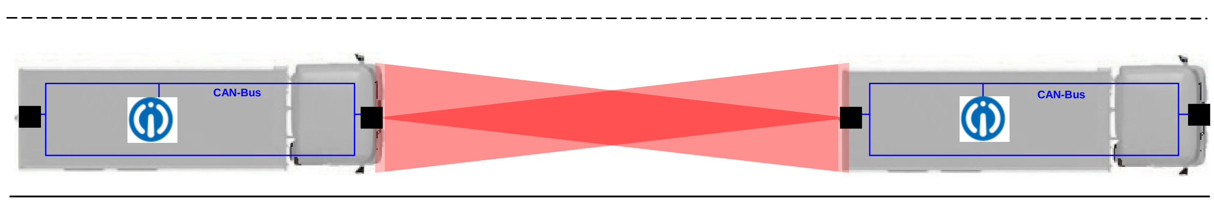


Figure 38: V2V communication

The V2V system consists of two hardware components: a front and rear transceiver. These stand-alone units include the optics, processing and CAN hardware all in one package. Therefore, only power and CAN lines must be routed to these units. This simplifies the installation on the vehicle. When selecting mounting locations, it is important to consider several factors, e.g. approximate optical alignment with any vehicles it will communicate with, remaining relatively free from dirt and water spray and not be exposed to mechanical damage. The selected mounting positions of the V2V units for WP5200 (AQuA truck) and WP5400 (Active Green Driving bus) are shown below. During mounting of V2V units, it is important to ensure approximate optical alignment of the sensors.

Vehicle integration

Integration of the V2V units on the AQuA truck is shown in Figure 39.

Figure 39: Integration of the V2V units on the AQuA truck.

Figure 40: Mounting locations for the V2V units on the WP5400 bus.

The units were mounted high on the vehicle, as this location is most appropriate to allow optical alignment on both the AQuA truck and AGD bus. The front unit was properly integrateded into the upper headlight mounting. The rear unit was mounted at the top of the rear cargo door, using the door locking mechanism as a mounting point.

Integration of the V2V units on the WP5400 bus is shown in Figure 40. The rear unit was installed on the mesh above the engine compartment and the front unit was mounted inside the upper part of the windscreen.

Hardware architecture description

The V2V communication hardware is placed within a square metal housing. This metal housing fulfils all requirements for temperature range and outdoor mounting. The metal housing provides gadgets for the mounting of the components. The front plate of the V2V communication module contains the lenses for the wireless Infrared communication.

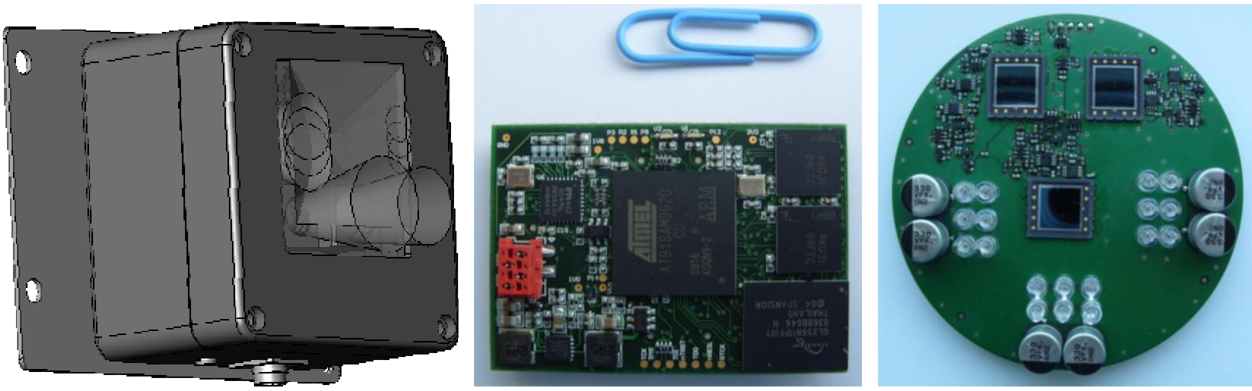


Figure 41: V2V Communication Module (left: housing, center: base board, right: IR board)

The electronic hardware of the V2V communication module has three main components: Controller board, base board and infrared board. The controller board is the brain of the V2V communication components. This PCB is equipped with an ARM-9 µController and the required memory chips. The µController hosts the software modules of the V2V communication board. The base board forms the backbone of the component. Controller board and Infrared board are connected to this PCB. The base board contains the chips for the CAN communication interface and the CALM-IR ASIC. The CALM-IR ASIC was developed by EFKON and contains the CALM Infrared communication protocol as specified in the ISO 21214 Standard.

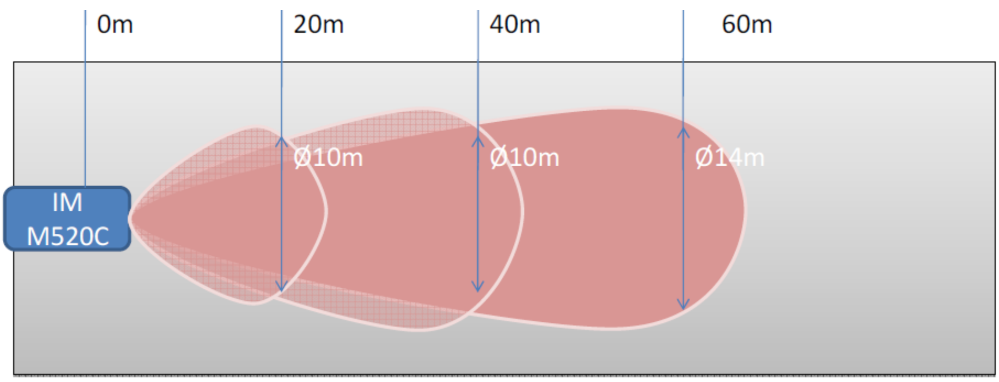


Figure 42: Shape of CALM-IR Beam

The Infrared PCB hosts the Infrared LEDs for the transmission of Infrared light and PIN-Diodes for the reception of Infrared light. The placement of the diodes is important for the shape of the Infrared beam. Figure 42 shows the shape of the CALM-IR beam.

Software architecture description

The software of the V2V communication component runs on an ARM-9 µController. Linux is used as Operating System for the controller. The software architecture of the V2V communication component consists of three main components: HAVEit application, CAN communication stack and CALM IR communication stack. An overview of the software modules is given in Figure 43.

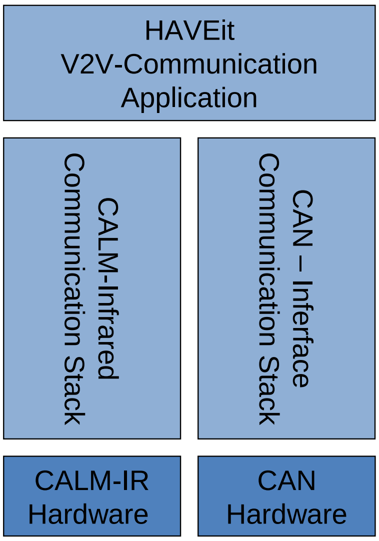


Figure 43: V2V Communication Module - SW Architecture

4.2.4 V2I Communication

Vehicle-to-infrastructure (V2I) communications can provide useful information to improve the world perception that on-board sensors cannot provide due to their limited range. To demonstrate the concept of V2I communications, where data is exchanged between the infrastructure and the vehicle to improve traffic management and safety, a wireless communication system is developed by INRIA within the HAVEit project.

Figure 44: 4G Cube – compact wireless router

This communications system is a Vehicle Web Service Communication Framework (VWSCF) also called SCOPE that has automatic service discovery. The proof-of-concept consisted of setting a dynamic speed limit sent by the infrastructure to the HAVEit Joint System Demonstrator (JSD) vehicle. This received speed was then used by the Co-Pilot component to adjust the current vehicle speed set point. This communication system was integrated into the HAVEit Joint System Framework that is running in the JSD demonstrator vehicle.

The hardware added to the FASCar consists of a 4G Cube (Figure 45) – a compact wireless router running SCOPE and OPENWRT router operating system – connected by Ethernet to the already existing onboard computer where the HAVEit Joint System Framework is running (see Figure 46).

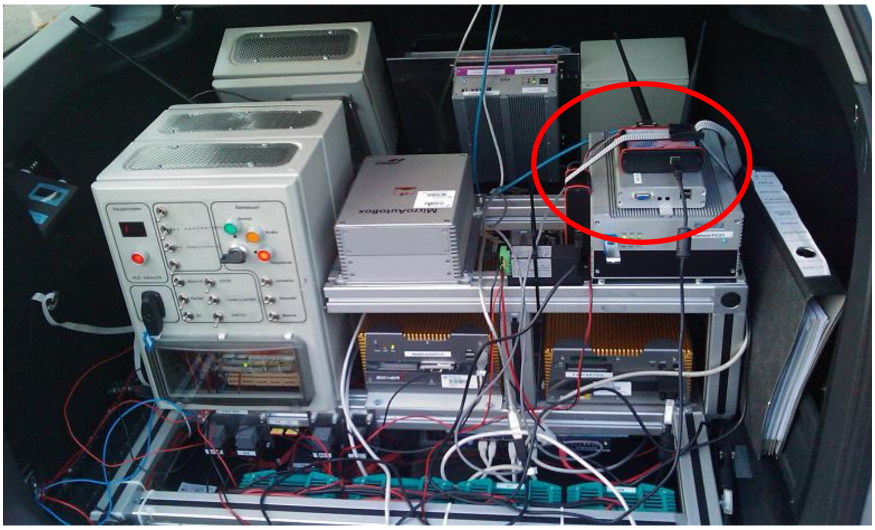


Figure 45: 4G Cube integrated into the FASCar

Architecture description

The communication architecture is modular and is based on embedded Linux boxes (4G Cubes) that are used to provide automatic connectivity for V2I applications.

The 4G cube key features are:

- CPU: x86 500 MHz AMD Geode LX800
- IPV4 / IPV6
- OLSR Mesh
- Service discovery using OLSR
- 1 mini-PCI a/b/g
- HTTP Web services Scope Server

These boxes can be used as simple routers, or like advanced communication units. The embedded communication software uses web services. These web services are scalable and any new web services can be easily developed and fast deployed. In the current version it consists of three web services: Discover, Expose and Fetch that can deal with different kind of data (structures, xml etc.).

An OLSR "Optimized Link State Routing" protocol was also developed. Its proactive behavior is appropriate for vehicle communications. OLSR allows vehicle mesh networks to be quickly created and dynamically reconfigured, since it is designed for multihop networks with a strong and efficient mechanism for data flow delivery. This protocol can run over WIFI and use UDP or Signal Noise Ratio to measure the quality of the link to establish the best route to transfer data. We use regular WIFI 802.11b/g that can have up to 54Mbit/s bandwidth (802.11a is also possible at 54Mbit/s). But 802.11p should be also used for CALM M5.

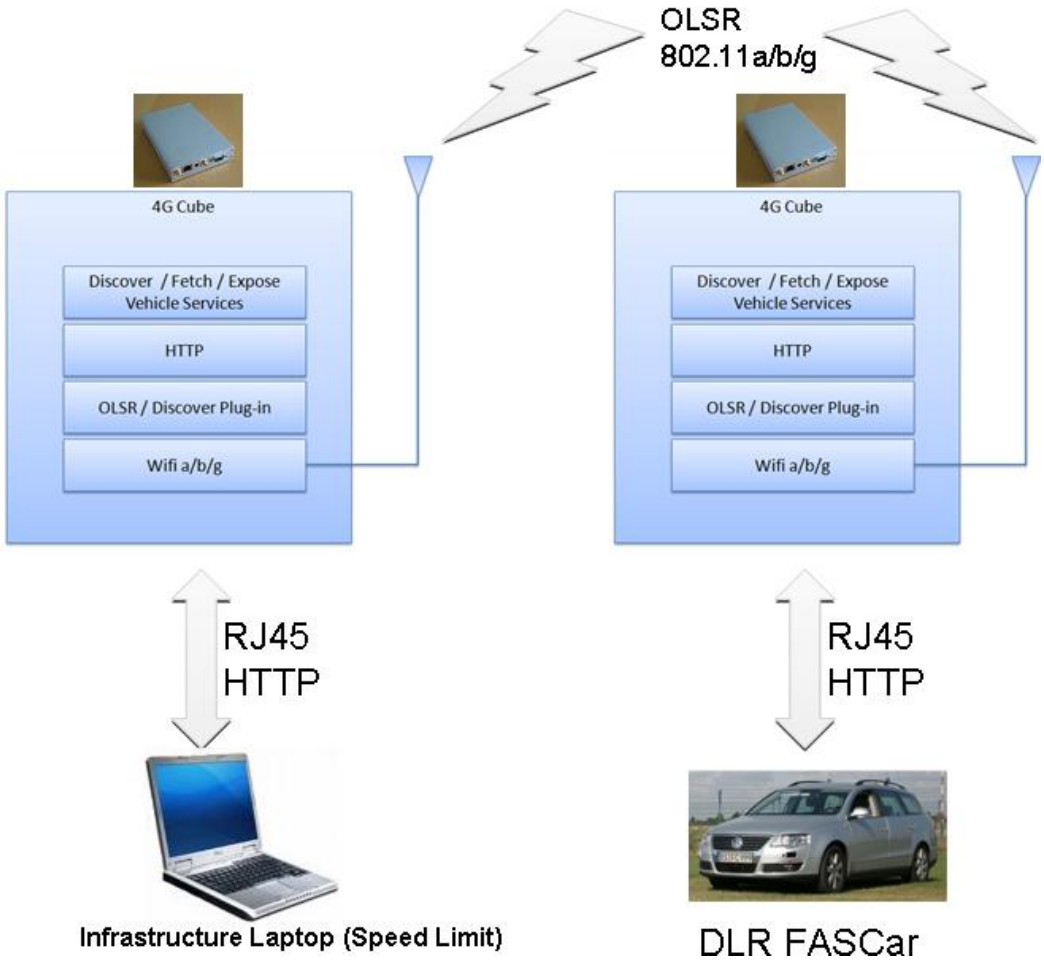


Figure 46: Communication architecture for V2I

On the system presented below, each user is able to request information to the communication hardware using a simple HTTP 1.1 protocol. It is not necessary to deal directly with the C++ communication API. Communications between all the cubes are completely transparent and they can use IPV4, IPV6 (actual work) or both protocols. The computers or microcontrollers plugged into the cubes get their IP by DHCP and they only need to deal with HTTP protocol to send, get or discover data on the network, no fixed IP.

OLSR is responsible of routing correctly and managing multihop data transfer. To transfer data, the user can use for example LIBCURL, a HTTP dedicated library to discover, expose or receive data. This library is multi operating system and can be embedded on C or C++ software. If the infrastructure communication box has access to the internet, it can give the connectivity to the connected vehicles.

Communication interface specification

The communications component is connected to the data fusion component. There is no direct link between the on board computer and infrastructure. The communication is managed by the 4G Cubes and vehicle/infrastructure data are propagated to the dynamic mesh network. The communications systems can send and receive any type of binary data: custom data structures, video stream, laser scans data, etc. For the following proof-of-concept, a simple data structure containing a speed limit variable in *short int format* is used.

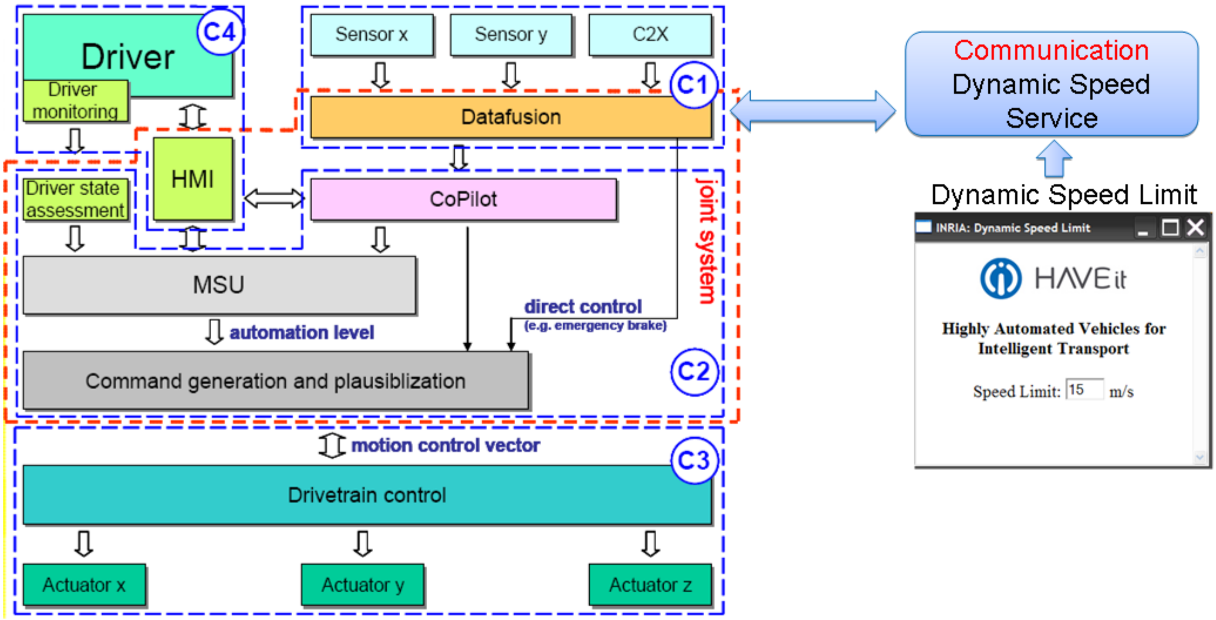


Figure 47: Interface between Joint System and Communication Component

4.3 Challenge 2.3: Actuators for by-wire systems

4.3.1 Brake-by-Wire actuators for the BbW truck (challenge 4.2)

Introduction

Current brake systems on heavy vehicles are pneumatically actuated which is a drawback in order to reach optimized performance under all road conditions. Even though the development of the pneumatically actuated systems have gone through several steps with a higher degree of electrification to reach increased performance, including ABS, EBS and ESP, still the actuation at the wheel-end uses the same principle as 30 – 40 years ago. In order to take a further step towards reduced stopping distance, energy optimized braking and increased performance in conjunction with advanced driver assistance systems (ADAS) a system based upon Electro Mechanical Brake (EMB) actuation is required. Brake systems together with steering systems are classified as safety relevant which means they must cope with a single failure in the systems in a fail-safe way.

The international braking regulation of the UNO Economic Commission for Europe – ECE-Regulation No. 13 gives the requirements for vehicle braking systems within the EU and can in principle be used for the type approval of an EMB system. Paragraph 5.2.1.2.7.2 of the regulation says that there must be at least two completely independent energy reserves, each provided with its own transmission likewise independent. In addition, paragraph 5.2.1.13.1 says that each of the energy reserves must be equipped with a warning device, detecting the energy level.

The aim of WP 4200 was to design and equip a 4x2 truck (Volvo FH12) with an EMB system according to the vehicle 2E architecture of HAVEit, meaning two electronic systems; no pneumatic system (see Figure 17). A safety case should be built. Thereafter pre-homologation of the vehicle to operate it on public roads should be started in cooperation with an independent institute (TUEV Nord) and the vehicle manufacturer (Volvo).

The functional requirements relate to redundancy / safety considerations. The vehicle platform design has to support both the basic brake function as well as the vehicle stability functions such as ESP and fully comply with the intention of ECE-R13. This regulation does not foresee electrical energy as primary source for brake actuation today; therefore equivalents needed be worked out.

Based on test results on high and low µ surface from an earlier test set up it was expected that a heavy vehicle using an EMB system will have a shortage of stopping distance of 15% in average compared to existing pneumatic systems on the market. The EMB truck system is using the principle of self-enforcement for generating brake force. This means very low energy consumption, compared to existing systems of today, when the braking process is achieved. As this system will replace the pneumatic actuators it was also expected that the vehicle braking will be more silent and thereby contribute to noise level reductions. Within the frame of HAVEit challenge 2.3, the brake-by-wire actuators were developed.

Brake-by-Wire architecture

Figure 17 shows the principal of the Brake-by-Wire system architecture. It mainly consists of following subsystems.

- The System Control Modules (SCM) covering the Power Distribution Management (PDM), Star Coupler, HMI and J1587 interface and ESP sensors.
- The XCCs having the functionalities brake system control and redundancy management.
- The gateway serving as a communication interface between vehicle CAN and FlexRay for the brake system.
- HMI for the service brake and parking brake functions, function switches and steering wheel angle.
- EMB wheel end units for brake actuation.

Details about the characteristics of the main subsystems are described in the sections below. The system layout is an x-split whereas the front left and rear right service brakes are connected to SCM Blue and front right and rear left service brakes are connected to SCM Red. By this set-up it is possible to incorporate the function for secondary braking performance within the service brake function. Today this is typically achieved by using the gradual hand brake valve, which its main function is the parking brake. Thereby a simplified parking brake activation and control can be done only by using a switch. In order to achieve fail safe operation the service brake and parking brake function at each wheel end respectively are powered and controlled by separate sources (Red and Blue).

a) *System Control Module (SCM)*

The SCM is the central module for power and communication distribution in the system. It is placed centrally on the vehicle chassis and includes also ESP sensors. Communication to other modules in the system (XCC, gateway and wheel-ends) is done with FlexRay. SCM also include interface to HMI and act as gateway for those signals (brake/steering demand) to / from the brake system application in the XCC´s (see Figure 48).

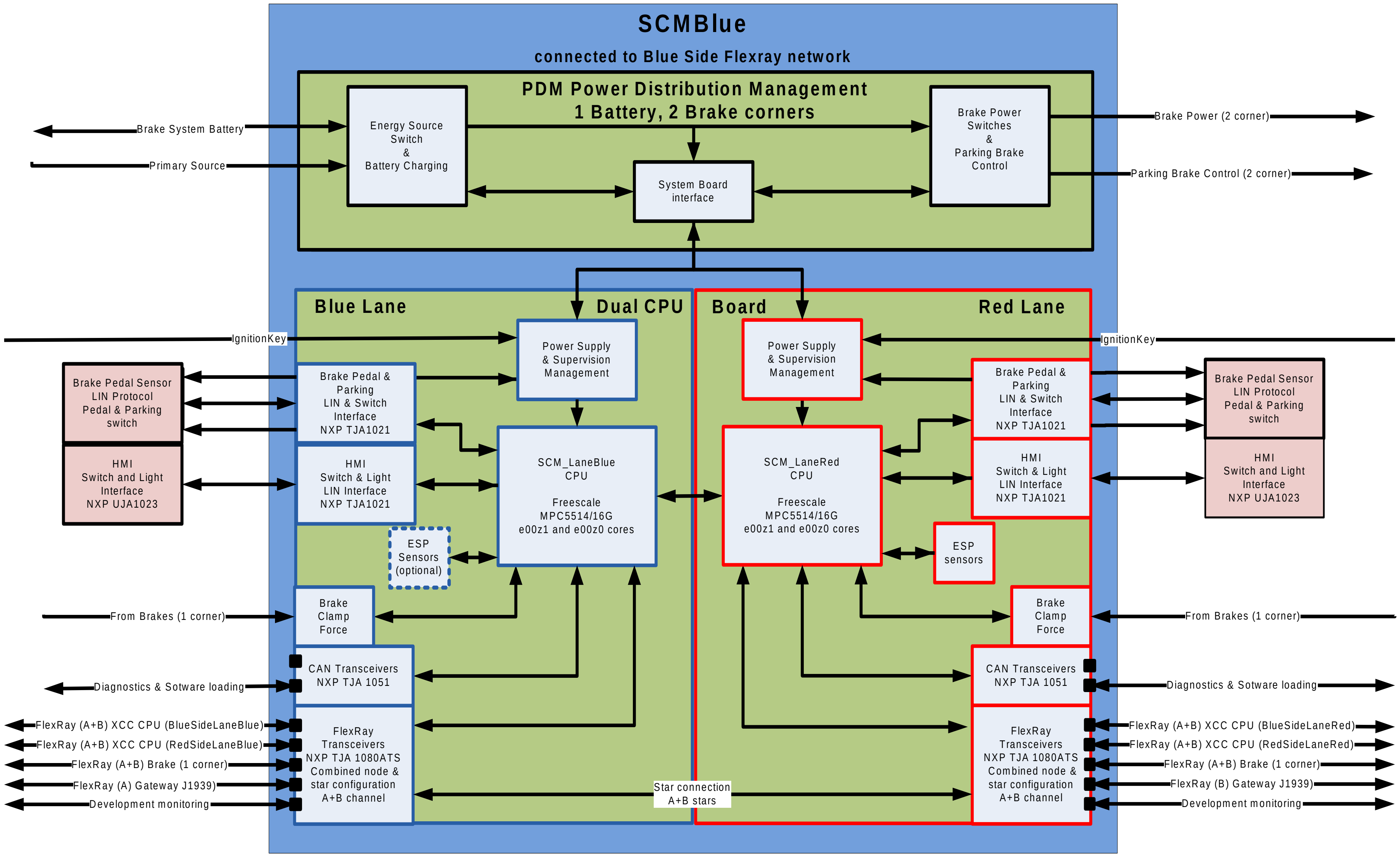


Figure 48: SCM schematic layout

SCM design priorities were:

- Design the ECU for future brake system where electric redundancy, system software redundancy and/or FlexRay communication is required.
- Adopting the system safety and redundancy level to the dual XCC design. This requires 2 SCM modules. One for the blue side and one for the red side of the XCC design safety and redundancy solution.
- Simplify the design by keeping the number of FlexRay transceivers and controllers low.

b) *Brake pedal HMI*

Today's existing EBS brake pedal has a dual-circuit pneumatic and a dual-circuit electrical structure. Actuation in initiation is recorded electronically by a switch. Further pneumatic redundancy pressure is delivered in circuits 1 and 2. In case of (electrical or pneumatic) failure of a circuit, the other circuits remain functional. If the electrical signals are aligned to each other (brake signal is OK) these signals will be used as the driver brake demand and the pneumatic signals will be suppressed.

Concerning the brake pedal for the HAVEit system (see Figure 49) no pneumatics are used but instead two dual electrical circuits. Each of the circuits connects each of the SCM modules. The sensors used are contactless Hall Effect sensors; TLE 4998, transmitting a PWM signal to the SCM´s. The counterforce to create a sufficient pedal feeling is generated by a spring package (see Figure 50). The signal characteristics (stroke vs. deceleration) can be varied depending on the preferred customer set-up. In the HAVEit project a characteristic was used which is rather aggressive in order to create a similar pedal feeling as in passenger cars.

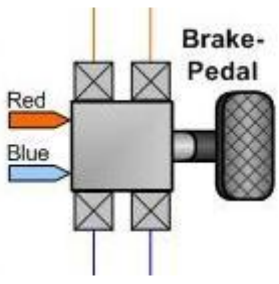


Figure 49: HAVEit brake pedal communication principle

c) *Parking brake HMI*

The existing EBS parking brake valve includes the function of the secondary braking performance according to the ECE-R13 paragraph 5.1.2.2. It controls both the front- and rear axle spring brakes on this particular vehicle. As already mentioned the HAVEit system is built as an x-spilt. Therefore, the parking brake HMI can be made very simple, as it does not have to include the function for secondary braking performance. In principle it is a push button having four digital switches, whereas two signals per side are foreseen for each of the SCMs (see Figure 51).

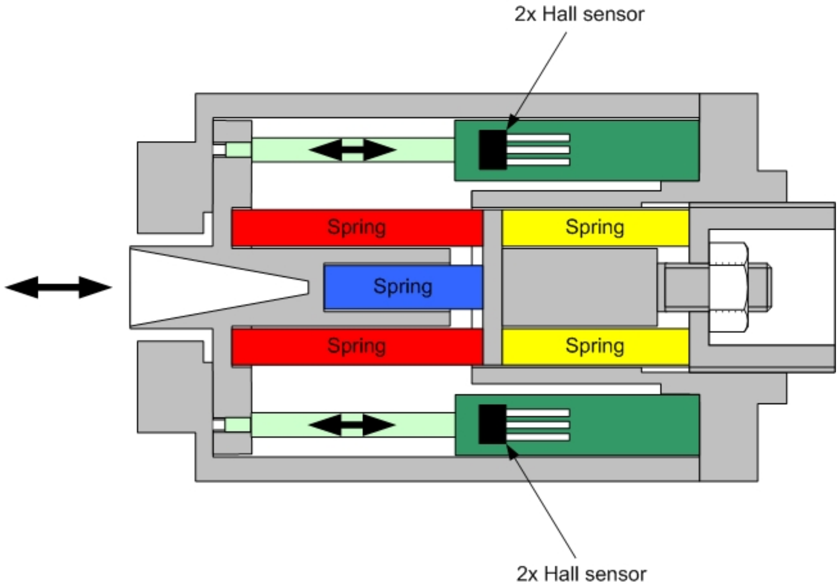


Figure 50: HAVEit brake pedal and counterforce arrangement principle

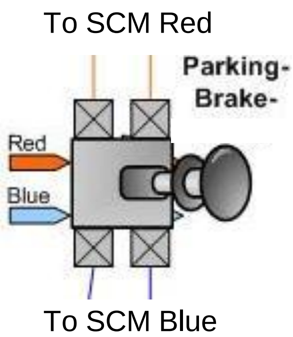


Figure 51: HAVEit parking brake switch principle

d) *Functional switch board HMI*

Certain vehicle functions can be manually activated by the driver depending upon the current driving conditions. The control of these functions is done by the braking system as the brakes can be used also to improve for instance traction control when starting on low μ surface. A separate PCB (see Figure 52) converts the push button signals to a LIN standard protocol, for communication with the SCMs. The functions in are Traction Control System (TCS), hill hold, trailer check and differential lock control.

e) *XCC*

The XCCs are described in detail in section 4.1.2 above.

f) *FlexRay-CAN gateways*

The FlexRay-CAN gateways are described in detail in section 4.2.2 above. A separate gateway between the vehicle CAN (J1939) and the FlexRay system of the Brake-by-Wire system was realised. It is primarily to handle the data communication to and from the engine ECU on the truck. For further information concerning communication matrices see HAVEit deliverable D22.1.

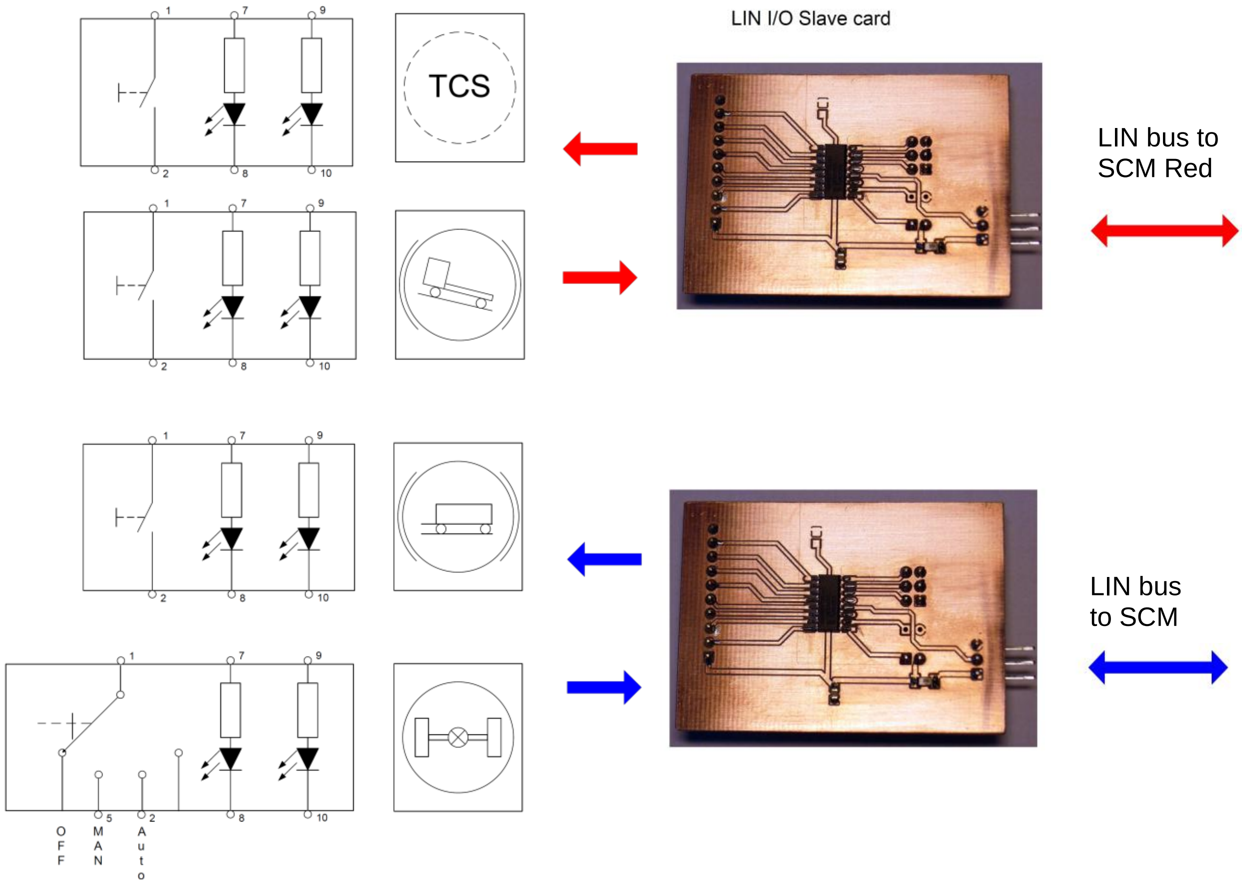


Figure 52: Functional switch board and communication to SCM

g) *EMB wheel-end modules*

The EMB wheel-end actuators and their design have a determining impact on the improvement in braking distance reduction and reduced power demand for generating brake force. The brake torque on each wheel-end can be individually monitored and controlled in a very accurate way which allows a better utilization of the existing tire/road friction conditions. In particular in emergency brake situations and during traction control on low friction conditions significant improvement can be achieved compared to a pneumatic brake system of today. A principal figure in form of a slip – µ curve shows the difference in control principle (see Figure 53).

The EMB wheel end is built to fit on existing axle installations for heavy vehicles. The design is modular which allows a high degree of freedom to fit for instance a separate package with a mechanical spring for the parking brake function (see Figure 54). To generate clamp force between brake pad and disc the mechanical design is based on the principle of self enforcement. This means only very low amount of external power is required to generate high clamp forces.

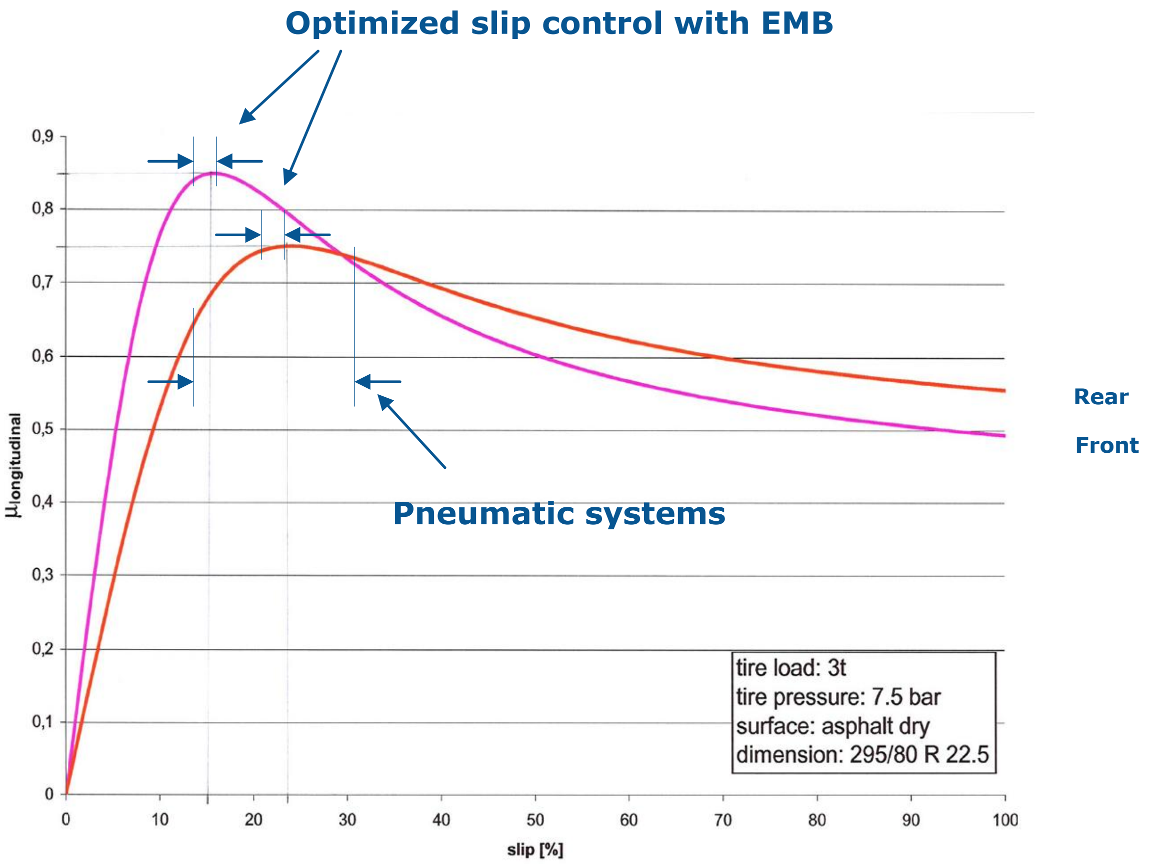


Figure 53: Slip - μ curve comparison of control principle. Slip control (EMB) vs. ABS control

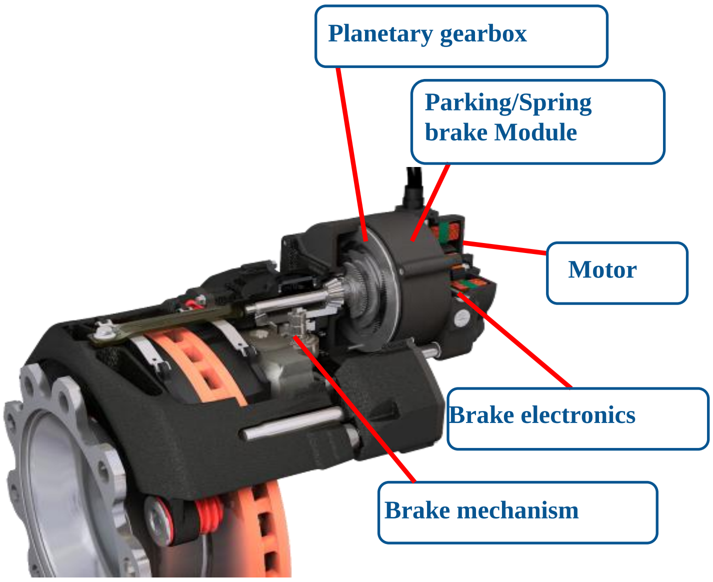


Figure 54: EMB wheel end actuator principle design.

When applying the brake the requested force level is sent from the SCM to activate the electrical motor at the wheel end. Via a mechanical gear and a ramp mechanism for amplification the motor movement generates clamp force. The applied clamp force is adjusted by the motor to correct level by the use of internal sensors (see Table 3 for key technical data and Figure 55 for picture of EMB prototype).

Min clamping force	<1 kN
Max clamping force	240 kN
Max brake torque	30 kNm (limited to 22kNm due to guidepins in prototype, can be exceeded in special test case)
Parking brake torque	min 18kNm
Emergency brake torque	min 18kNm
Clamping force accuracy	± 1 kN + 2% of requested force level
µ nominal value	0.38
µ variation normal operation	± 20 %
µ variation limited operation	µ-min 0.2 µ-max 0.6
Build-up time to 75% of max clamping force	<200 ms

Table 3: EMB wheel end actuator and technical data

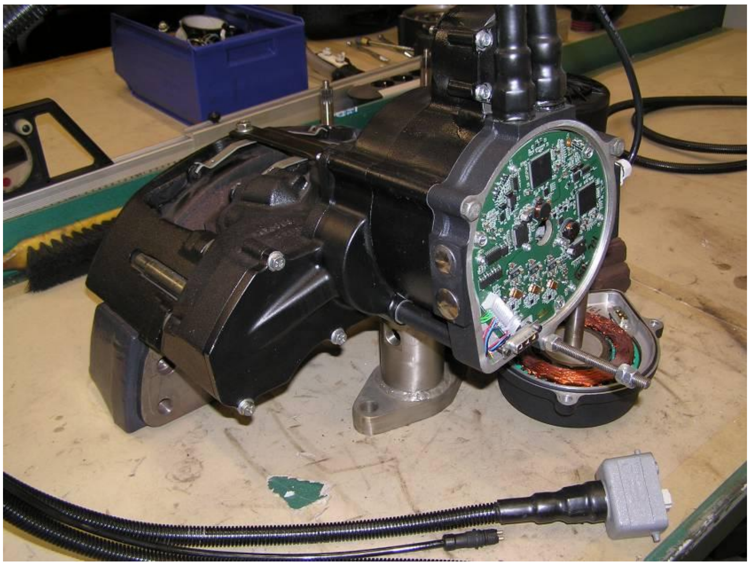
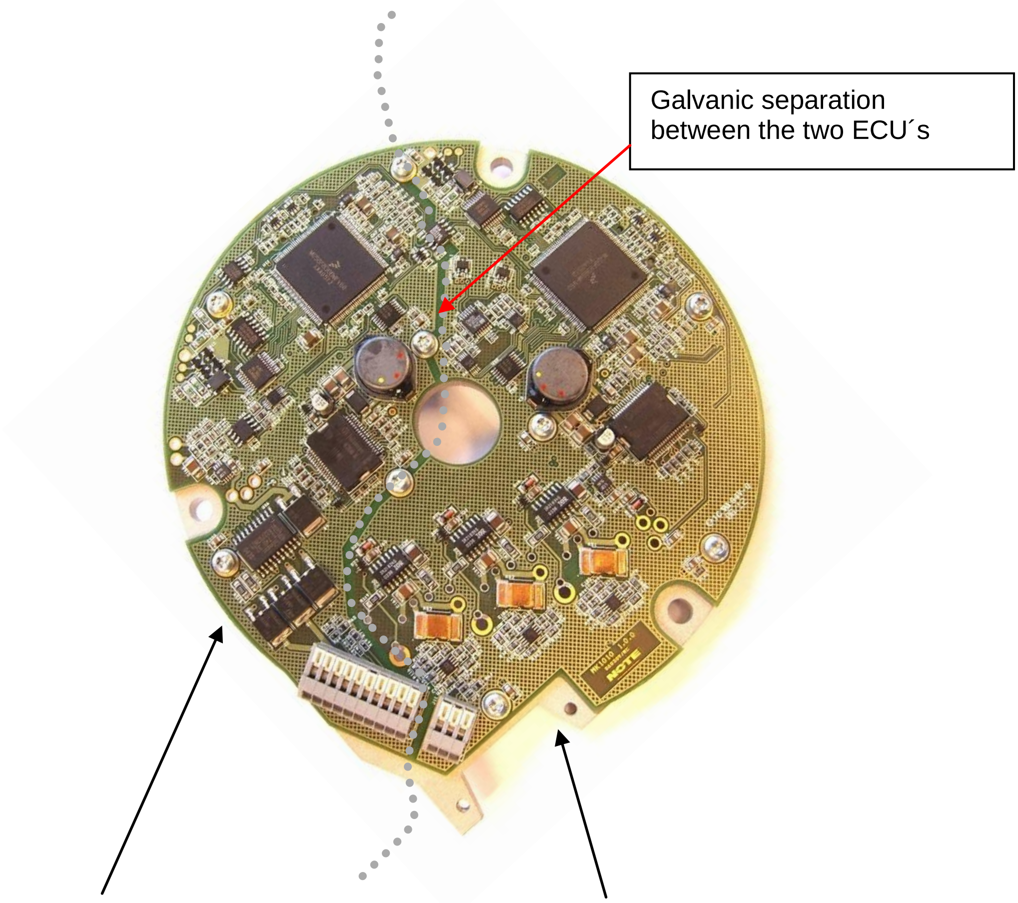


Figure 55: EMB wheel end actuator prototype build

Each EMB wheel end has electronics for motor drive and brake control. On one PCB two separate ECUs are placed galvanically separated. Besides several identical functions between the two ECUs (communication, clamp force measurement etc.) one ECU performs the motor drive control and the other takes care of the parking brake function. Each of the two ECUs are separately powered from PDM Blue respectively PDM Red (see Figure 56).

h) *Communication architecture*

The brake system communication is based upon two asynchronous dual-channel FlexRay networks named as "FlexRay Blue network" with channel a (FR(B,a)) and channel b (FR(B,b)) and "FlexRay Red network" with channel a (FR(R,a)) and channel b (Fr(R,b)). The FlexRay cycle time of each of the networks is 10 [ms]. The brake system application software is running in four CPUs in the two XCC units. The XCC units are named "XCC Blue" and "XCC Red". The two CPUs in each XCC unit are named "Lane A" and "Lane B". All four CPUs are connected to both asynchronous networks. For further information on the communication Brake-by-Wire architecture see also D22.1 chapter 1.3.



Wheel end ECU 1 (PDM Red)	Wheel end ECU 2 (PDM Blue)
Motor drive circuits/sensors	Parking brake spring circuit
Clamp force sensor	Clamp force sensor
FlexRay channel A communication	Wheel speed detection
Ignition key (KL 15)	FlexRay channel B communication
CAN for diagnostics/flash programming	Pad wear sensor
Freescale 56F8366 DSP hybrid controller	Ignition key (KL 15)
Freescale MFR 4200 FlexRay communication controller	CAN for diagnostics/flash programming
	Freescale 56F8366 DSP hybrid controller
	Freescale MFR 4200 FlexRay communication controller

Figure 56: EMB wheel end electronics and functional split

4.3.2 Steer-by-wire for Joint System Demonstrator (challenge 4.1)

The section describes the technical realisation of the steering actuator required for the demonstration vehicle of WP4100 (Extended Joint System Demonstrator). It contains the actuator description from the technical side and their integration in the vehicles. Also the basic functionality of the actuator was proven by demonstrating the measurement results of tests performed in the vehicle.

The steering actuator system mainly bases on three components, the serial steering system of the vehicle (opened for CAN control), the electromechanical clutch and the driver feedback actuator. The role of each actuator with respect to the steer-by-wire (SbW) system is described in detail in D22.1. A survey of their main components is represented in Figure 57. While the feedback actuator and the clutch are external devices to be installed, the serial steering system is part of the vehicle itself.

The main challenge was to replace the vehicle's original serial steering column by the new steering devices. Not only the mechanical integration was very challenging but also the electronic integration, due to the required integrity of the overall vehicle architecture which is needed for the functionalities that are not dedicated to the steer-by-wire system.

Between the driver's feedback actuator and the steering pinion the electromagnetic clutch is installed in the steering column. For safety reasons the clutch is kept open by an electro-magnetic coil, thus coupling automatically through a spring or permanent magnet when the electric power is shut off. Even in the unlikely case of a total power failure, the vehicle still stays controllable.

During the normal operation of the steer-by-wire system, the clutch is open and there is no mechanical connection between the steering wheel and the front axle. The feedback torque at the steering wheel is generated by the driver feedback actuator, while the steering task is performed by the steering actuator. In the WP4100 steer-by-wire vehicle, the steering actuator consists of the electric servo steering motor which is standard in the VW Passat. A software modification of the servo steering ECU allows to command the steering action via the CAN bus.

Figure 57: Feedback actuator, clutch and SbW-vehicle

Figure 58 gives an overview of the communication architecture for WP4100. The steer-by-wire system mainly consists of two steering actuators – separated from each other by a clutch – several input sources as well as the computing units (XCCs) which will perform on the one hand the control laws and on the other hand the redundancy management of the whole platform. In principle, the platform is divided into two sides (»red« and »blue«), each with their own FlexRay bus. Both FlexRay buses run asynchronously. Aggregates (actuators, gateways, interfaces) are generally connected only to one FlexRay bus, while the safety relevant duplex XCCs get access to both FlexRay buses.

The general functionality of the depicted system is as follows: The demanded steering angle is detected by the actuators themselves or provided by the Joint System. Steering commands are calculated within the XCCs and sent to the actuators. These commands are adapted regarding (1) the vehicle speed and (2) the current failure state of the system (e.g. in case of the loss of the servo actuator, the clutch is commanded to close and the remaining feedback actuator takes over the steering action, thus we could speak of a 2E/1M system architecture). In addition, in case of no failure, the feedback actuator is used to provide haptic feedback to the driver, initiated by the Joint System.

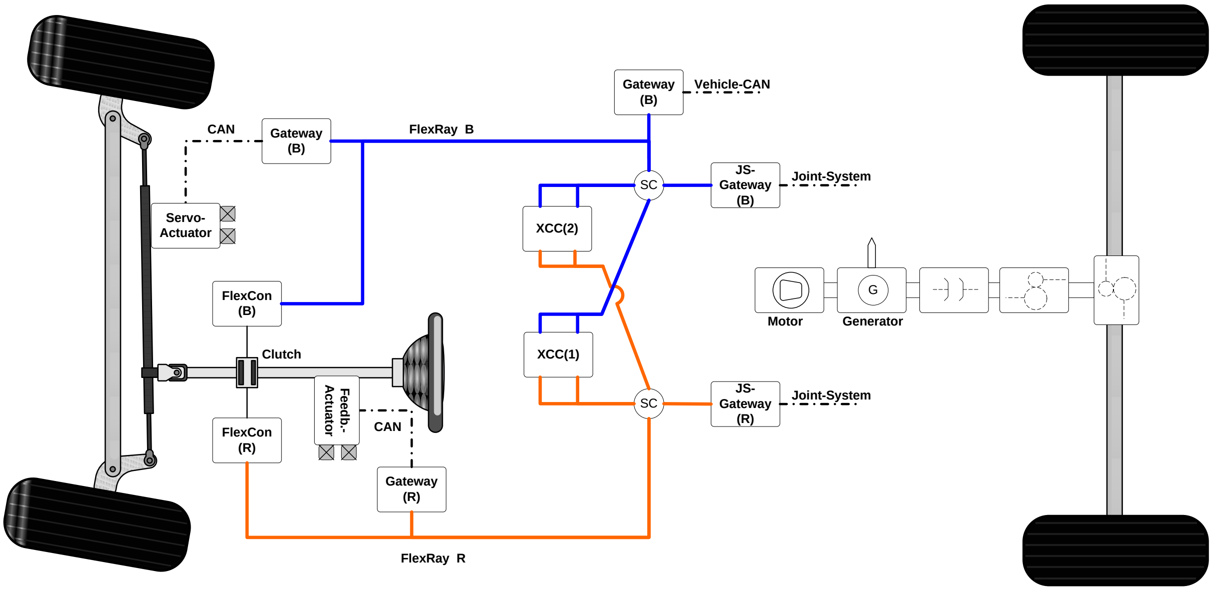


Figure 58: Extended Joint System Demonstrator SbW communication architecture

Driver feedback actuator

The driver feedback actuator (see Figure 59) is a near-serial device developed by TRW Conekt (TRW Conekt Active Steering Wheel System, Part-no. AC47133001). Its components were originally designed for the application in flight simulators but were rebuilt to use the technology for automotive applications. As result, the actuator can be obtained as a Commercial-off-the-Shelf (COTS) product for automotive simulators and for cars. This actuator allows the application of angle and force commands on the steering wheel during operation.

Figure 59: Driver feedback actuator by TRW Conekt

The feedback actuator requires a power supply capable of supplying 60A at 12VDC. The power supply should also be capable of absorbing power generated by the feedback actuator when the motor is driven by hand (using the steering wheel at high wheel turning rate). An alternative to a high rated power supply plus dump circuit is an automotive battery and charging circuit, the low impedance of the battery absorbing excess power when the feedback actuator is back driven.

In order to integrate the actuator into the vehicle, the standard mechanical vehicle installations were removed and additional mounting points were created. Figure 60 shows the integrated actuator. The actuator is placed between the clutch (on the left side) and the steering wheel on the right side (not visible here). Welding points were created in a way that the carpet could be closed after the mounting procedure so that the impression of the dashboard is not disturbed (see Figure 61).

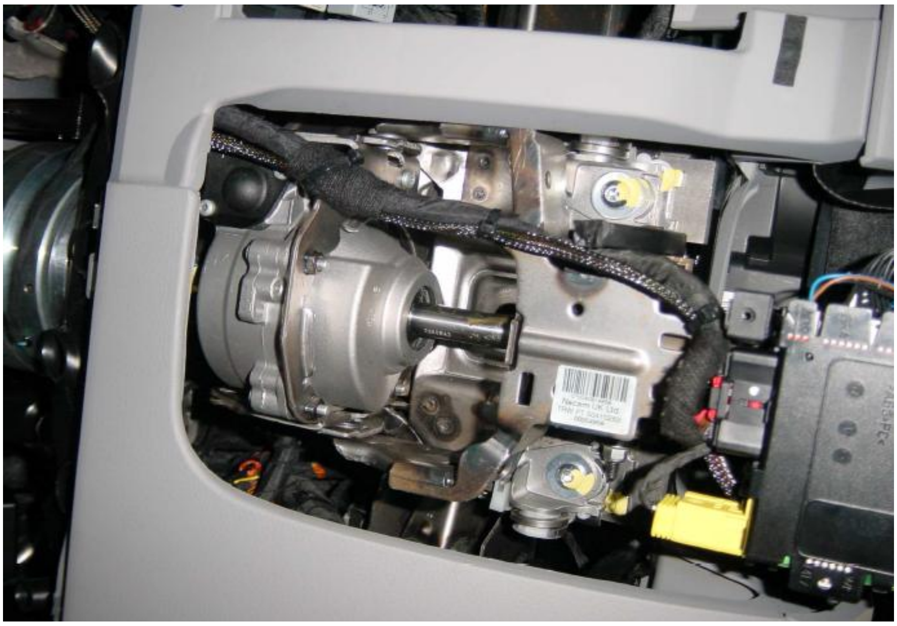


Figure 60: Feedback actuator integrated in vehicle

Figure 61: Feedback actuator integrated

Figure 62 shows the measured data recorded during a simple experiment to test the performance of the feedback actuator after installation in the vehicle. During this experiment the clutch was open and therefore the actuator mechanically decoupled from the steering axis. This setup complies with the normal mode when the steer-by-wire system is operational.

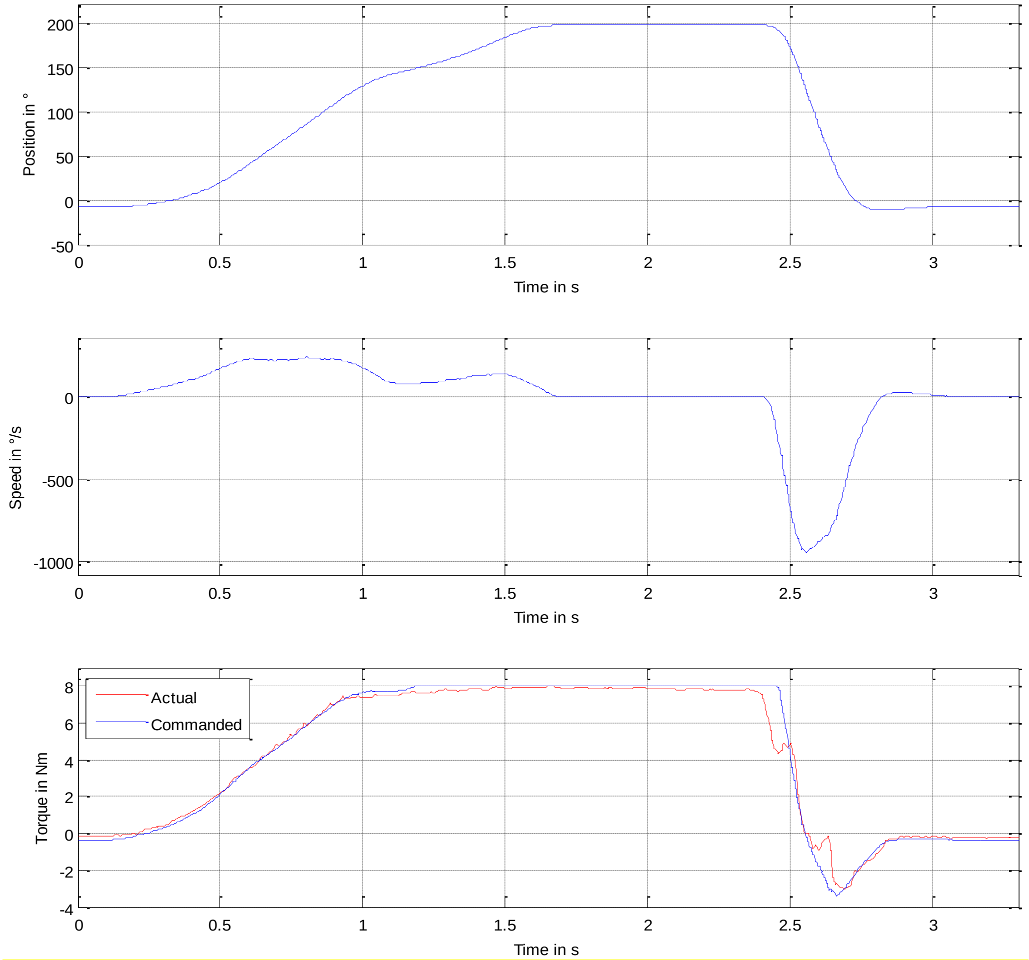


Figure 62: Recorded position and torque data from driver feedback actuator

In the experiment, the actuator was driven by a simple position controller with a linear gain of 0.05 Nm / ° (capped at 8 Nm) and a damping of 2.5 Nm s / °. It should be noted that these controller parameters were created strictly for testing purposes, they are in no way applicable for real driver feedback applications as the occurring torques and wheel speeds are much too high.

The steering wheel was manually turned 200° approximately between 0.5 s and 1 s against a torque rising according to the controller parameter. The feedback torque follows the deviation approximately in a linear way comparable to the behaviour in a real vehicle. At about 2.5 s, the steering wheel was completely released, therefore spinning back towards the zero-position with hands off the wheel. The oscillations in the falling slope are a result of the position controller that accelerates and decelerates the wheel in order to reach the zero position. These oscillations can not be detected by hands because they only occur in hands-off-mode.

The remarkable result of the experiment is that such a simple controller was able to guide the released wheel back to the zero-position practically without any overshoot even though the wheel was accelerated up to almost 1000°/s. This controller performance is accomplished mainly by two factors: The cycle time of the unit is with 2.38 ms very small compared to the time constants of the mechanical system and the turning speed of the steering column is directly measured rather than calculated using the time-derivative of the position signal. This greatly enhances the stability of the damping term.

Electromechanical clutch

The main requirements of the electromechanical clutch are low dimensions, a minimum torque of 50 Nm and mechanical closing in power-off. Due to the planned vehicle integration the diameter must not exceed 18cm and the full length has to be adapted between the steering actuator and the Cardan joint that connects the system to the vehicle steering actuator. An additional challenging requirement is the displacement of the steering angle encoder from the original steering wheel to the bottom of the steering column in order to keep the vehicle angle control loop working. A product that achieves these requirements is the Combinorm-T supplied by the German manufacturer KEB (Karl E. Brinkmann GmbH, 32677 Barntrup).

Figure 63 shows the mechanical integration. The clutch is completely mounted and fixed between feedback actuator and Cardan joint. The only electrical connecting interface to the clutch is the field coil. With a signal of 24 VDC the clutch closes and without voltage it opens again automatically and disconnects the feedback actuator mechanically from the steering shaft. After the mechanical installation work the construction was closed with the original carpet as shown in Figure 61.

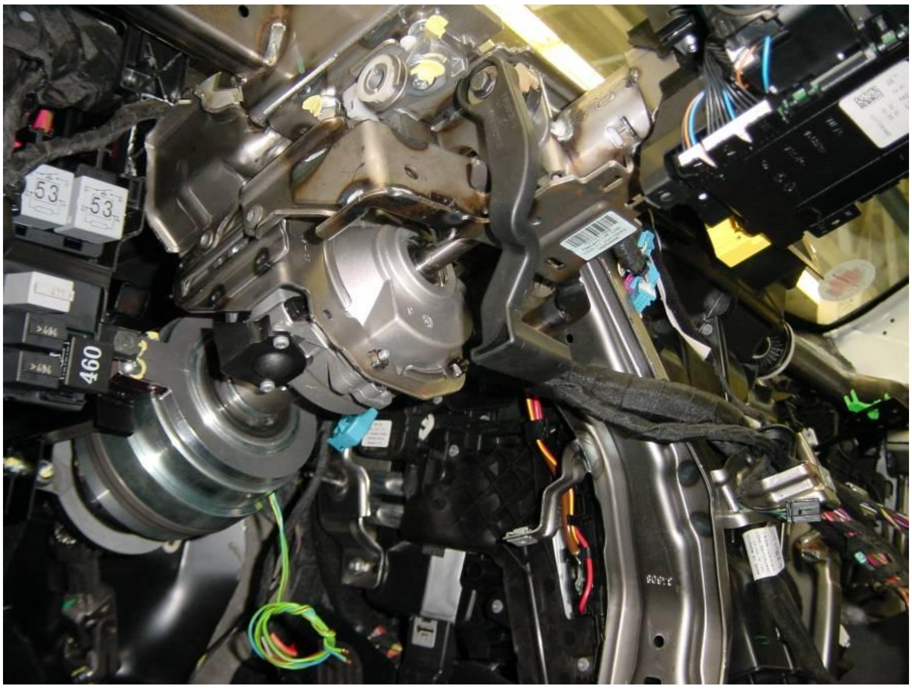


Figure 63: Steering column (clutch and feedback actuator) integrated in the vehicle

Steering actuator

The main requirements for the steering actuator were the fail-tolerant operation behaviour and the integrity of the device. A device with these properties was found in the serial steering actuator of the VW Passat in which the actuator usually acts as a power assisted steering device in order to support the drivers manual steering angle command. For the steer-by-wire purpose an electronic communication channel had to be created directly into the actuator (preferably CAN or FlexRay) to enable the system to carry out a steering angle command by itself.

An answer to this request was found in the serial electrical steering actuator of the VW basic vehicle that was extended with two main functionalities: The serial vehicle electronic architecture was modified in a way that a torque or angle command can be applied via CAN bus. The general commands are described in Table 4. The maximum torque to steer the front wheels was extended compared with the serial application. In tests it was proven that even for a standing vehicle the tires can be moved with satisfactory dynamics. The extension of the maximum torque was motivated by two reasons: First to increase the dynamic behaviour and second to be sure that the tires can be moved even when the vehicle is standing.

Signal name	Remarks
Checksum	Checks integrity of communication
Cycle counter	Lost message detection
Status	Operational, errors, mode selection (torque, angle), etc…
Steering torque demand	Demand that will be applied on the wheels
Torque	Actual value
Steering angle demand	Demand that will be applied on the wheels
Steering angle	Actual value

Table 4: Commands from/to steering actuator

For the separate communication channel into the power steering, the watchdog functionality and for the simulation of the vehicle bus three CANlog devices were built in the luggage compartment (see Figure 64). Additionally, some other devices like power supply components, a D-GPS platform and the corresponding cable installations were integrated.

Figure 64: Electrical integration of electronic modules

An overview about the devices and communication channels that are added to the basic vehicle architecture is represented in Figure 65. The yellow CAN channel is dedicated to the steer-by-wire system while the others are addressed to further drive-by-wire functionalities like E-gas, E-brake, gear shift and the execution of other requests by the co-pilot system.

In order to prove the feasibility of the actuator, a set of tests was conducted at DLR. Of particular interest and importance for the success of the steer-by-wire system implementation is the step response test. Results are shown in Figure 66.

From the zero-position, the steering actuator is commanded to move to 200° (corresponding to a wheel angle of 12.5°) after 0.1 s. At the time of the test, the vehicle was at zero speed, not supported by an auto-hoist, standing on an anti skid floor in a garage. Under these circumstances the necessary forces at the steering rod are much higher than during normal driving conditions. The commanded position is nevertheless reached within 0.3 s, which implies that the actuator is capable of performing the steering task during steer-by-wire operation without problems.

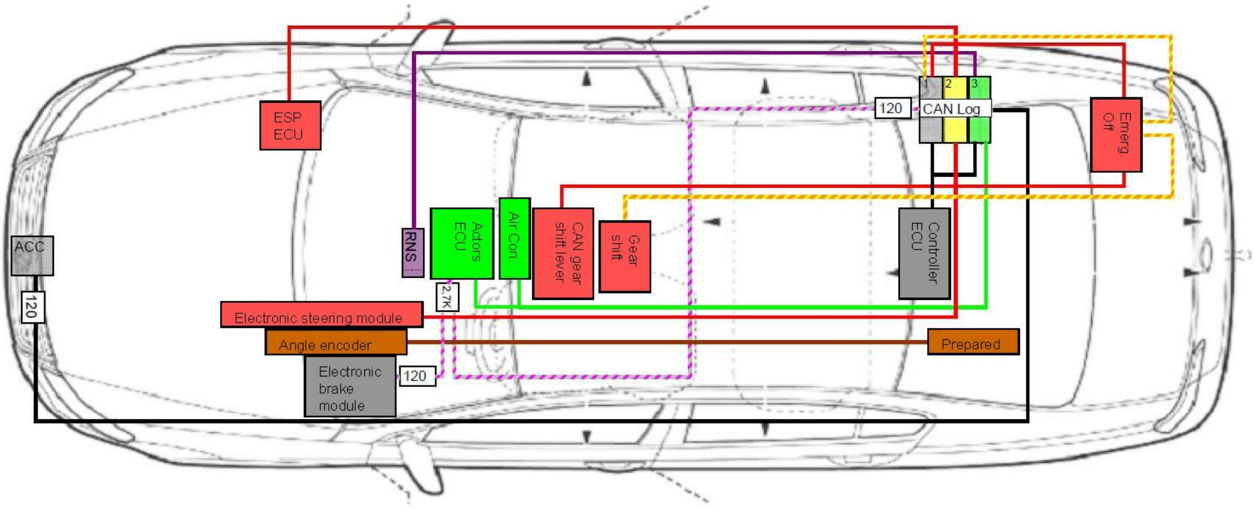


Figure 65: CAN architecture and ECUs in the SbW vehicle

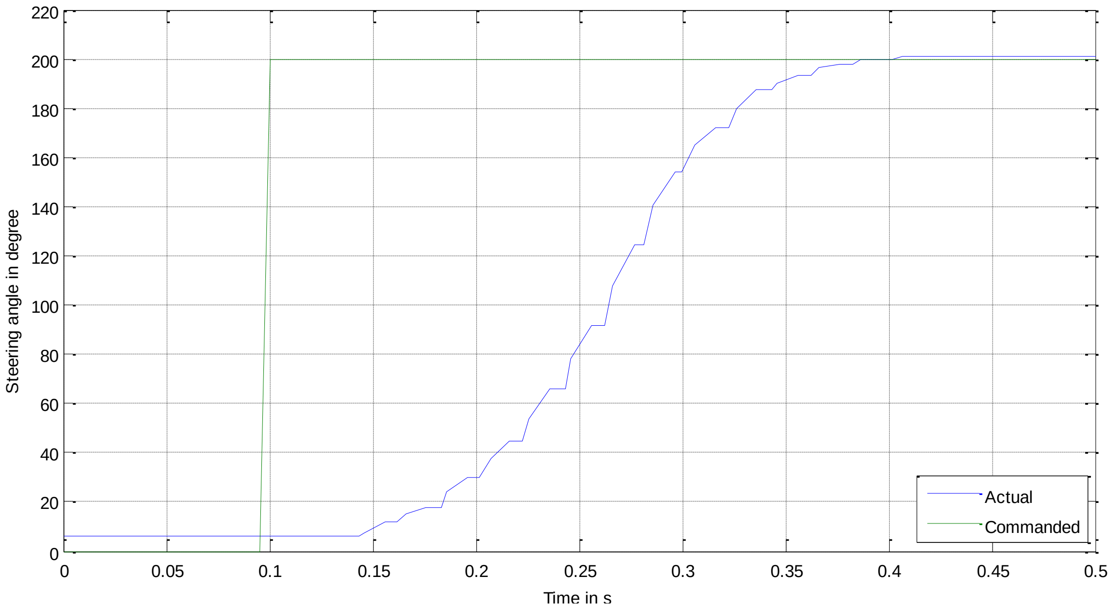


Figure 66: Step response of the steering actuator

4.4 Conclusions

From sections 4.1-4.3 it can be concluded that all requirements concerning the challenges clustered under HAVEit sub-project SP2000 “Safety Architecture Implementation” were met. Components developed were built, tested and supplied to partners for use in the HAVEit demonstrators (vertical challenges).

5 Horizontal Challenges: Joint System Driver / Co-System

In this section, results achieved within the frame of HAVEit's horizontal challenges in cluster 3 are described in detail. To summarize:

- Challenge 3.1: Co-system
- Challenge 3.2: Driver state assessment
- Challenge 3.3: Joint system design: driver / co-system

5.1 Challenge 3.1: Co-system

This challenge covers the design of the co-system. According to Figure 3 the co-system constitutes of perception, maneuver planning, trajectory computation and command generation. INRIA, LCPC and DLR contributed to the development of the Co-pilot system in collaboration with the following partners: ICCS as data fusion provider and SICK as a sensor provider.

5.1.1 Perception and data fusion

5.1.1.1 Low level fusion of Infrared laserscanners (SICK)

Overview and architecture

The laser scanner system developed and provided to HAVEit consists of multiple laser scanners and a processing unit called laser scanner ECU. The typical system has the following architecture as depicted in Figure 67.

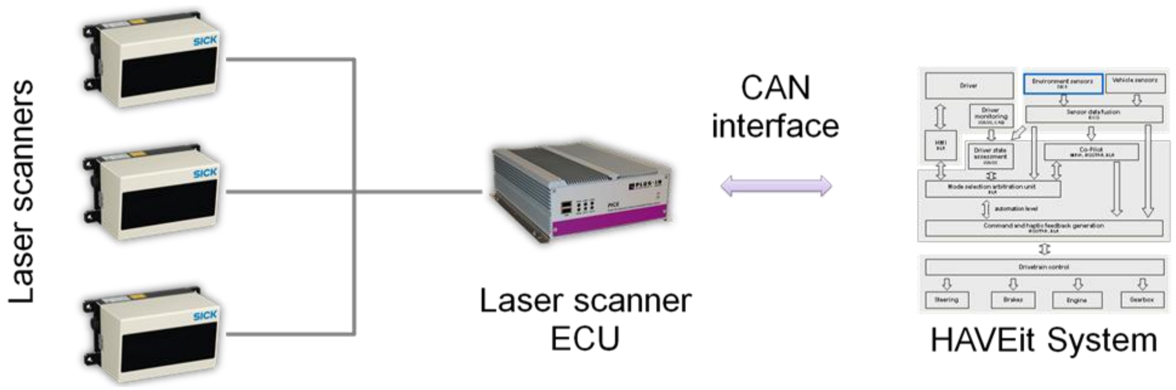


Figure 67: Laser scanner system architecture

The laser scanners used are modified "ibeo LUX 3" sensors with specially adapted and optimized FPGA firmware and DSP firmware for HAVEit. The scanners are connected via Ethernet (utilizing the TCP/IP protocol) to the ECU. The ECU is a Plug-In PICE 3110 model running on Linux based firmware which has been developed by SICK in the course of the project. The ECU provides the aggregated, processed and interpreted fusion data to the "High-Level-Fusion" of HAVEit via CAN.

Laser scanner integration

The laser scanners have been installed into the front bumper of

- the DLR Passat (3 laser scanners),
- the VW Passat (3 laser scanners),
- the VTEC AQuA truck (3 laser scanners) and
- VTEC AGD bus (2 laser scanners).

Figure 68 is showing a typical installation. The central laser scanner (blue) is facing forward while the left (red) and right (green) sensors are facing outwards. The measurements of the laser scanners are indicated as colored dots. The red dots are measurements of the left laser scanner, the blue dots of the central laser scanner and the green dots of the right laser scanner. By varying the orientation of the left and right laser scanner, the field of view can be increased or decrease resulting in a smaller or larger area in front of the host-vehicle realizing redundant measurements. In other words, a smaller total field of view increases the overlapping area of the sensors' individual field of view and therefore increases the robustness against failures of a single sensor. An additional benefit is the increased number of measurements per object which increases accuracy and reliability of the whole system. On the other side, an increased total field of view decreases the blind areas and consequently increases or enables object detection in the vicinity of the host-vehicle. Based on the specific needs of the sub-projects, the left and right sensor is facing outwards between 20 to 45 degrees resulting in opening angles of up to more than 180 degrees.

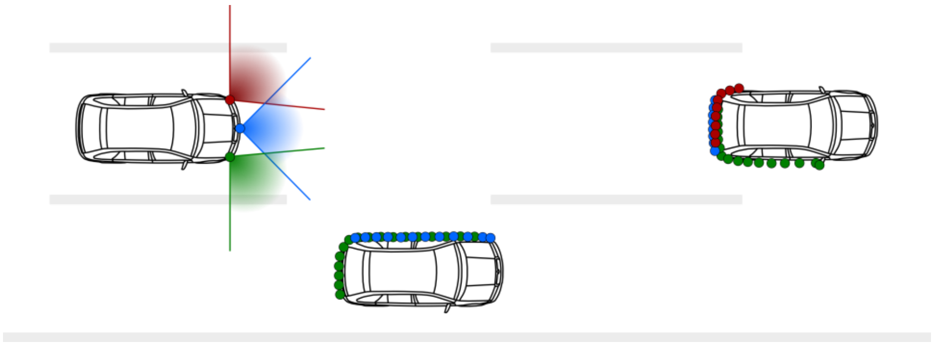


Figure 68: Typical laser scanner fusion system consisting of 3 sensors

In order to compensate the pitching motion of the host-vehicle as well as to detection lane markings, the laser scanners have 4 layers each (see Figure 69), with a vertical opening angle of 3.2 degrees. This technology ensures that the surrounding road users will be detected by the laser scanners even though the host-vehicle accelerates (negative pitch) or brakes (positive pitch). Additionally, the road surface is measured which is required for the relative positioning of the host-vehicle to its lane (see below for further details).

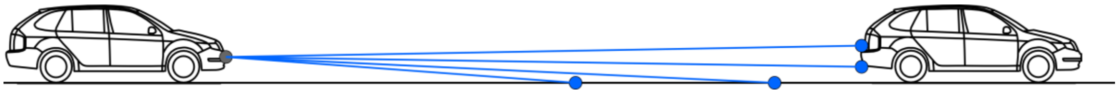


Figure 69: Multi-layer technology enables pitch compensation and lane detection

Data processing

The data processing graph is shown in Figure 70. The laser scanners gather a range profile of the environment. In order to perform a precise low level fusion, the scanners are synchronized in their scanning frequency as well as their measurement direction. The range profile is then sent to the laser scanner ECU. In a first step, the data are pre-processed. This processing detects and labels measurements of dirt, rain and snow. Additionally, the pre-processing estimates the host-vehicles state based on vehicle state data provided via the vehicle CAN bus.

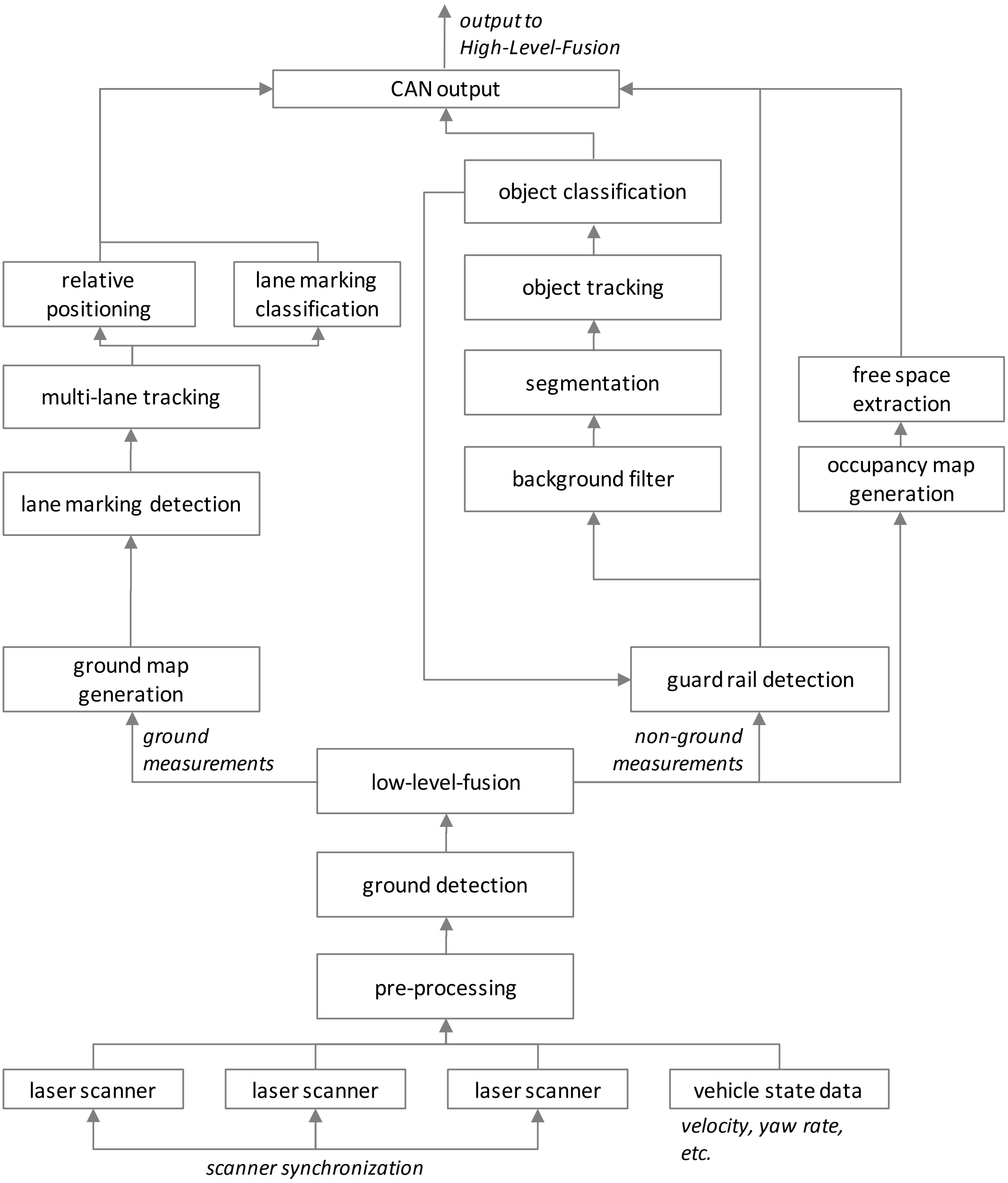


Figure 70: Laser scanner data processing graph

After the pre-processing, the raw measurement data of the laser scanners is separated into those scan points reflected off the ground (road surface) and those, reflected by any vertical object such as road users of any kind and guard rails. The separation is realized by labeling every single measurement (scan point) as required.

The subsequent Low-Level Fusion combines the pre-processed scans of the individual sensors to one single scan. From this point on, the data is treated as if it was gathered by one single sensor. This enables a flexible and modular architecture and makes the whole system robust against a failure of one or two sensors.

After the Low-Level Fusion, the data stream is split into the relative positioning path and the object detection path. The latter starts with the guard rail detection: The algorithm utilize expert knowledge about the scenario and object interpretation as well as a feedback from the object classification. By this feedback, a false interpretation of a long truck or bus as a guard rail is avoided.

Based on the knowledge about the relative position and orientation of the host-vehicle towards the guard rails (indicated as red lines in Figure 71), a background filter is applied, marking all measurements behind guard rails as not relevant for further interpretation.

The subsequent step is a segmentation of the fused range profile. This is a specialized clustering algorithm optimized for laser scanner data. The resulting segments are then analyzed for size, orientation, edges and corners as well as further properties relevant for the subsequent tracking algorithm. The tracking itself is using adapted Kalman filter techniques to follow the objects over time. The objects provided by the tracking are then classified based on their properties like size, speed, age, orientation and history. Object classes provided are truck/ bus, passenger car, motor bike, pedestrian and “unknown object” for anything else but the previously mentioned road user classes. An example screenshot (Figure 72) of the laser scanner data visualization shows tracked objects as well as background (yellow) – foreground (red) separated scan data. To support the reader of this document interpreting the scan data, a camera image is provided (Note: This camera image is not used for any calculations or data interpretation within the algorithms.).

The relative positioning path starts with the generation of a ground map which is continuously updated and shown as the black areas in Figure 71. This ground map is analyzed for lane markings (green lines), representing the measurements for the subsequent multi-lane tracking. This lane tracking utilizes separate Kalman filters for each lane: the host-vehicles ego-lane as well as the adjacent lanes to the left and right, if present. The presence of the adjacent lanes is also estimated and provide to High-Level fusion. This information supports the automated lane change maneuver.

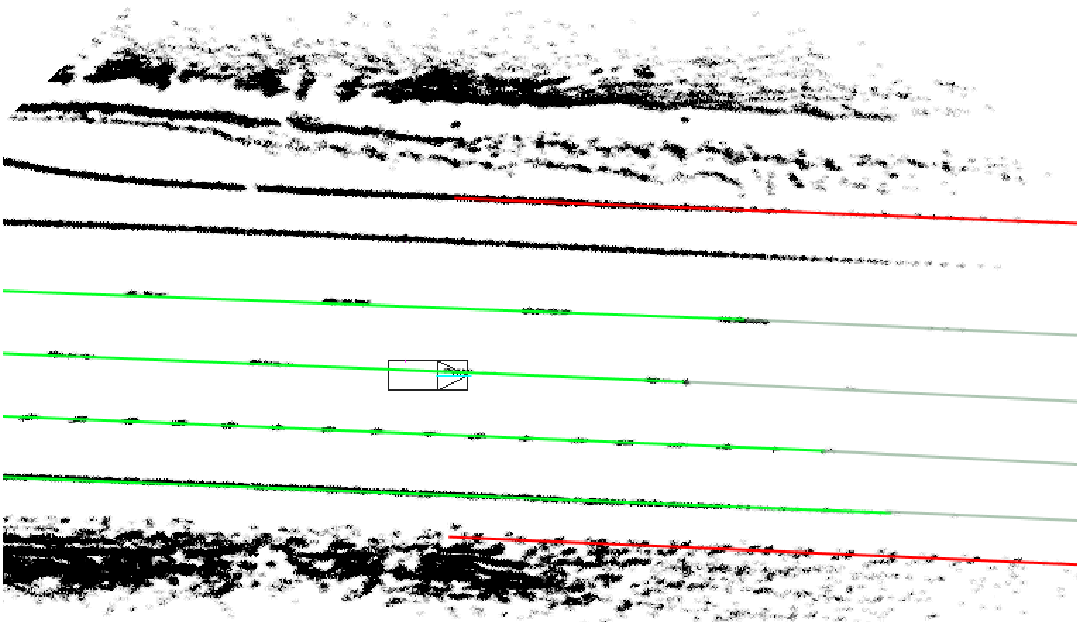


Figure 71: The relative positioning of the host-vehicle to its lane and adjacent lanes (green) as well as the guard rails (red) is based on on-line map data.

Based on the filtered lane information, a robust and reliable relative positioning of the host-vehicle to the lanes is calculated. In parallel, the lane markings are classified as solid lines or dashed lines.

The third path calculates an occupancy map from the non-ground scan data. This map is used to describe geometrically the free space (drivable area) in front of the host-vehicle which has been required by the Temporary Auto Pilot (Volkswagen demonstrator).

Figure 72: Tracked and classified road users, based on laser scanner data

Conclusion

The laser scanners' hardware and software has been adapted to the needs of HAVEit. The realized laser scanner based Low-Level Fusion system, developed within HAVEit has been integrated successfully into the Volkswagen demonstrator, both VTEC demonstrators and the DLR demonstrator vehicle.

The system provides a consistent environment model containing all relevant road users and objects in the field of view of the laser scanner system. Field of view covers an opening angle of more than 180 deg and a maximum detection range on cars and trucks of 200 m.

This environment model is augmented by the provision of a free space map, which holds a geometrical description of the drivable area in front of the vehicle. Additionally, the relative position of the host-vehicle to the ego-lane and adjacent lanes is provided. Furthermore, the lane marking type (dashed or solid) is provided. The system is able to detect lane markings up to 30 m in front of the host-vehicle.

5.1.1.2 High level fusion (ICCS)

In general, data fusion developed by partner ICCS is executed in two distinctive levels: the sensor level processing and the situation refinement. Figure 73 shows the lower level structure of the algorithm. Important blocks are presented in this section.

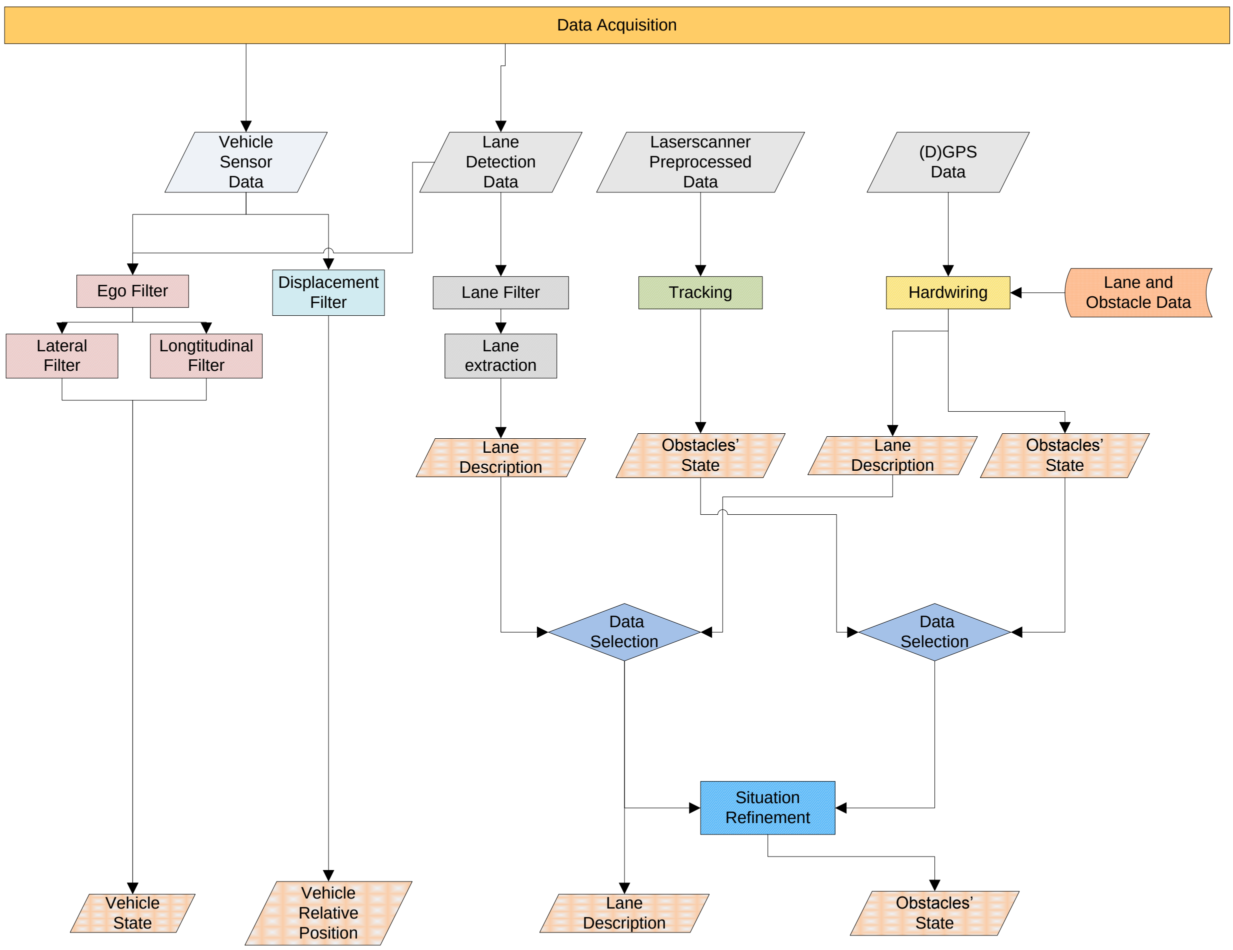


Figure 73: Data Fusion System Architecture

- Ego Filter

  When the ego filter is called, two separate Kalman filters are executed. These two filters form the ego filter's state vector. The longitudinal model makes use of the CTRA (constant turn rate and acceleration) model. The lateral filter processes lane and vehicle data and with the use of the CA (constant acceleration) model, the lateral dynamics of the vehicle are estimated. The ego filter algorithm is depicted in Figure 74.

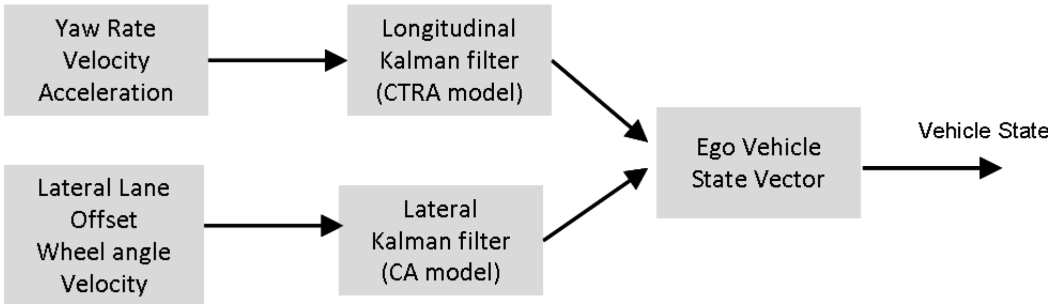


Figure 74: Ego filter algorithm

- Lane Filter

  Lane detection camera data are filtered with the use of an additional Kalman filter. The filtered output is used to extract the lane geometry. In the extraction, lane markings are used in order to identify the types of adjacent lanes. The geometry interpolation is done under the assumption that road follows the clothoid model equation. The lane filter algorithm is depicted in Figure 75.

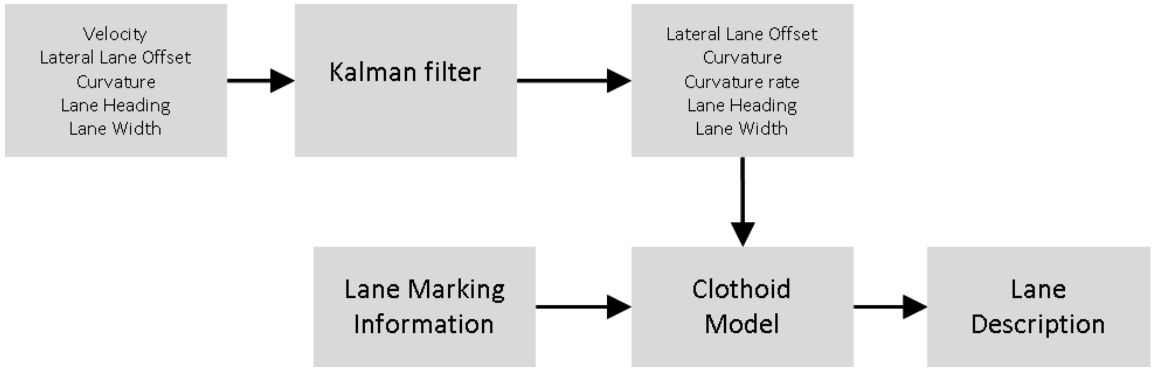


Figure 75: Lane filter algorithm

- Target Tracking

  Target tracking involves the procedure of assigning sensor measurements to tracks and the estimation of tracks respective states from these measurements. The tracker algorithm's data flow is illustrated in Figure 76. Tracks are associated with measurements after the gate computation, using the Global Nearest Neighbour data association. Afterwards, the track states are estimated after they have been updated with the associated measurement. Track management finally updates the tracks list, by deleting old tracks or confirming new ones.

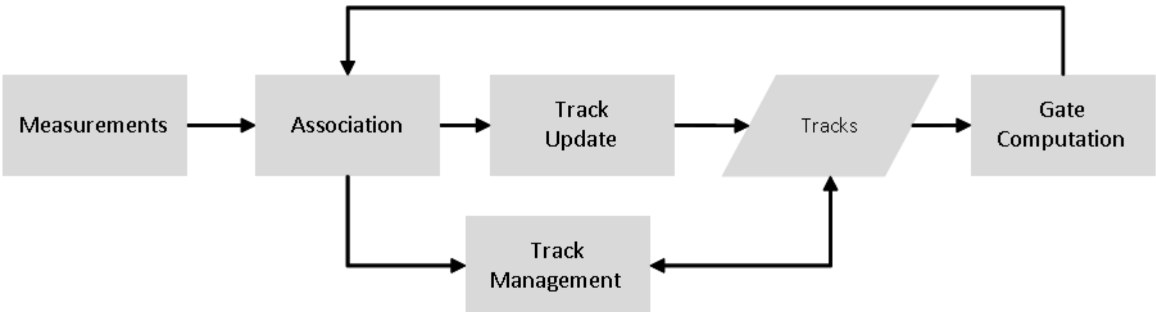


Figure 76: Tracker flow diagram

- Measurement to track association

This module has to associate the track list stored in the tracker from the previous scan, with the sensor measurements received from the current scan. First, for each measurement that falls inside the gate of a track, the Mahalanobis distance d_{ij} between this measurement y_j and the corresponding track $\tilde{x}_i$ in scan k is calculated. If d_{ij} is smaller than the gate size G_{ij}, the track score a_{ij} is calculated, otherwise the track score is zero.

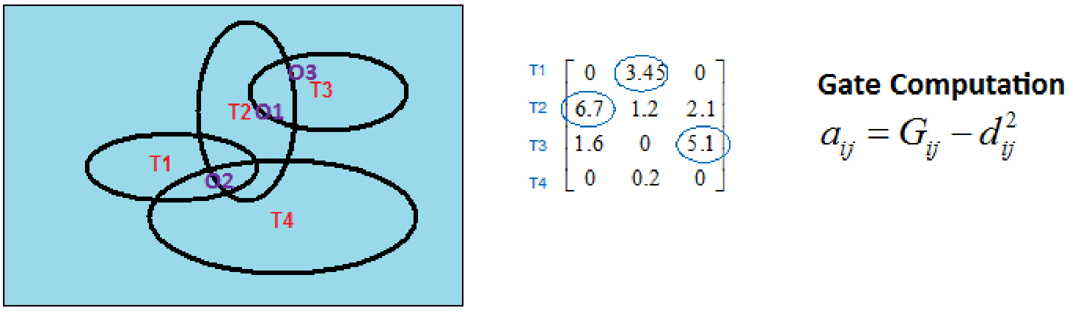


Figure 77: Gate Computation

The result of the gating is a matrix as the one shown in Figure 77. The ellipses represent the gate formation and measurements that fall inside the gate form tentative track-observation assignments. Afterwards, the assignment problem has to be solved: This is a profit maximization problem and the auction algorithm is used for solving it. The elements x_{ij} are either 0 or 1, whereas the constraints indicate that one measurement can only be associated with one target, and one target can only be associated with one measurement.

- Track management

The track management algorithm is based on the use of rules for track confirmation. The results of the data association are used. The un-associated measurements are initialized as tentative tracks and the corresponding track is assigned with a unique ID number. The tracks (tentative or confirmed) that have been associated with a measurement have their ‘hit’ value increased by one, whereas the unassociated ones have their ‘miss’ value increased by one. Afterwards the track management checks the ‘hit’ and ‘miss’ values for each track. Three track cycles were found to be the optimum between detection quality and computation time.

- Track State Update

The track state update process is the final step of the tracking algorithm. In this step in the track list there are tracks that are associated with a sensor measurement from the current scan, and also tracks that remain un-associated. If a track is associated, a Kalman filter algorithm estimates the track state at the current scan. For a non-associated track, the state estimation is predicted using a motion model and the track state from the previous scan.

- Track Fusion

  The track fusion module takes as input the track lists of the local trackers and gives a single track list in the output. The track-to-track association module identifies which tracks from different tracks list represent the same object. The procedure is similar to the track to measurement association, except that the Mahalanobis distance of the two tracks is calculated differently:

  As described, the track fusion gives to the global tracker a single track list. The global tracker's tasks are similar to the local's. The difference lies in the input, where the global tracker uses a fused track list while the local one uses the sensor measurement list. The track fusion algorithm is visualized in Figure 78.

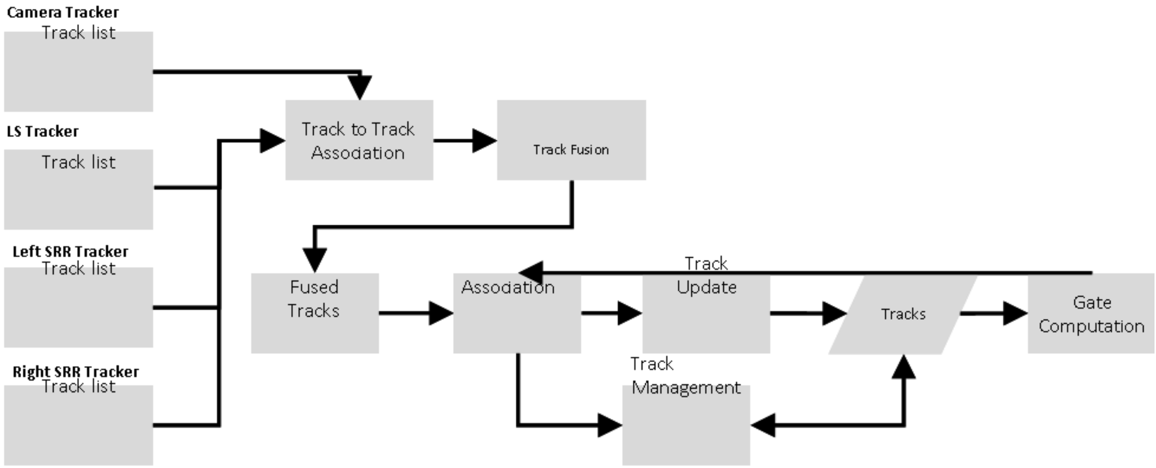


Figure 78: Track fusion algorithm

<u>Performance measurement</u>

As the data fusion function is closely linked to the hardware of the vehicle, the performance measurement of its algorithms is done through a field test on a vehicle. We isolated the component from the influence of other functions by manually driving the vehicle along one of the scenarios.

After the drive session, the following output variables with a direct link to the performance of this function are analyzed with respect to their correctness (error with respect to a known reference) and stability. A set of scenarios was tested in order to evaluate the following parameters:

- Tracking stability (ID maintenance)
- Object to Lane assignment
- False alarms presence in the lanes of interest (adjacent and ego lanes)
- Lane estimation accuracy
- Displacement estimation accuracy
- Calculation time

The coordinate system used in the *performance measurement* sections in this deliverable corresponds to the ISO 8855 norm, Figure 79. It is attached to the ego vehicle.

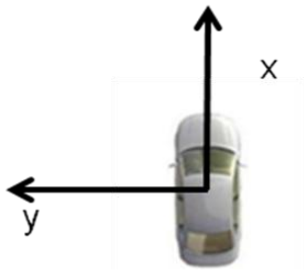


Figure 79: The ISO 8855 coordinate system

Following measurements present perception results achieved in certain scenarios using the sensor setup of the Joint System Demonstrator test vehicle from DLR.

- Two lane changes without any objects in the lanes

  In this scenario the ego vehicle executed two lane changes with no objects present in any of the lanes. In this scenario the existence of any false lane assignments in the adjacent or ego lanes can be identified. The closest distance of objects in the adjacent and ego lane are logged, with a value of -1 if no object was detected. As it can be seen from Figure 80, there are no false or clutter tracks assigned to the lanes of interest.

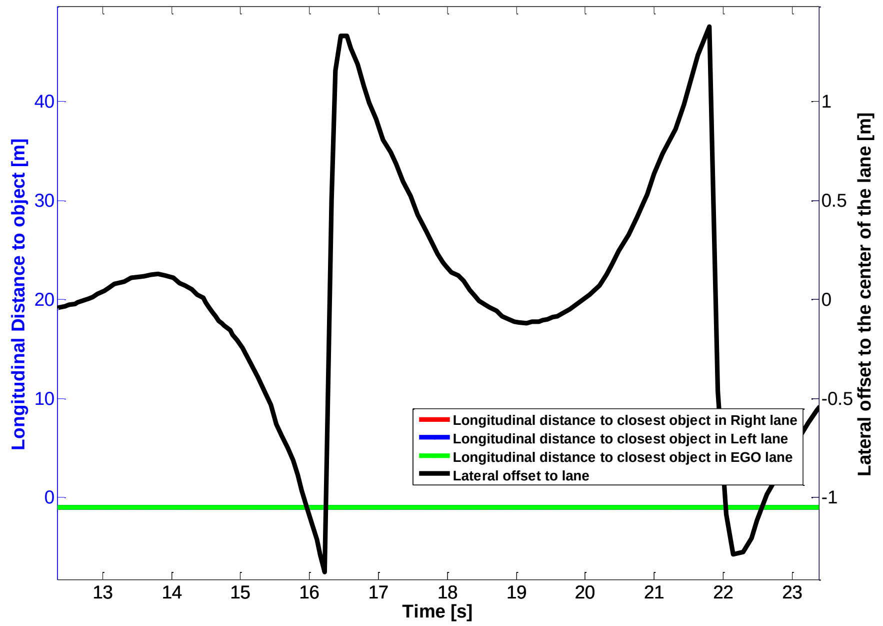


Figure 80: Closest distance of objects in adjacent and ego lane (2 lane changes)

- Left overtaking of a standing object

  In this scenario a standing obstacle was placed in the right lane. The ego vehicle approached the object and executed a lane change. The results of this scenario can be seen in Figure 81. The standing object is correctly lane assigned throughout the tracking period. Figure 81 indicates that initially the standing object is assigned to the ego lane, whereas after the lane change (transition of lane offset from a negative to a positive value) the standing object is assigned in the right lane. Apart from that object, no other track is assigned to any lane during the scenario.

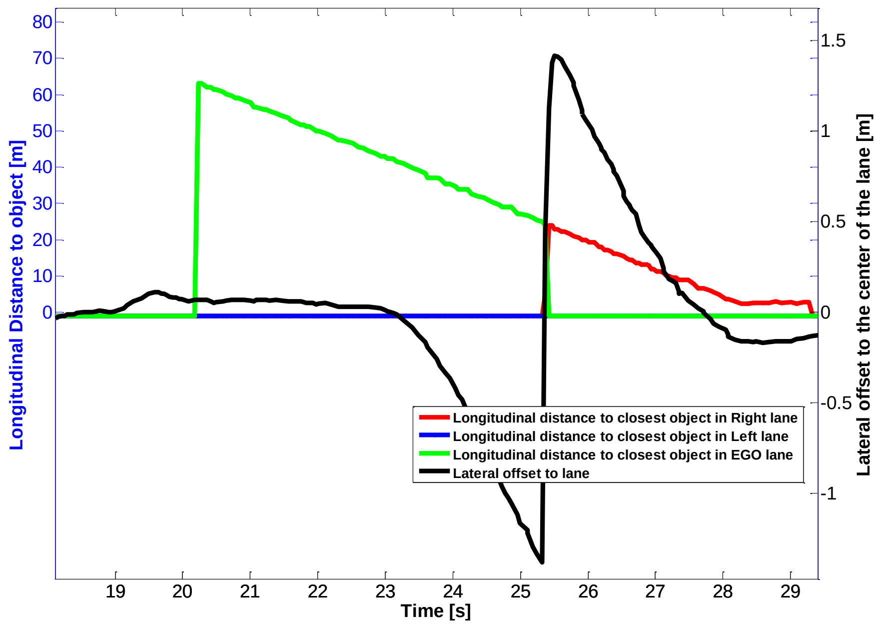


Figure 81: Closest distance of objects in adjacent and ego lane (Left overtaking of a standing object)

- Tracking of moving object in the same lane and lane change

  In this scenario, another vehicle is moving in front of ego vehicle (in the same lane) with nearly the same speed. The ego vehicle executed an overpass maneuver by lane changing to the left and accelerating, the results of this scenario can be seen in Figure 82 and Figure 83.

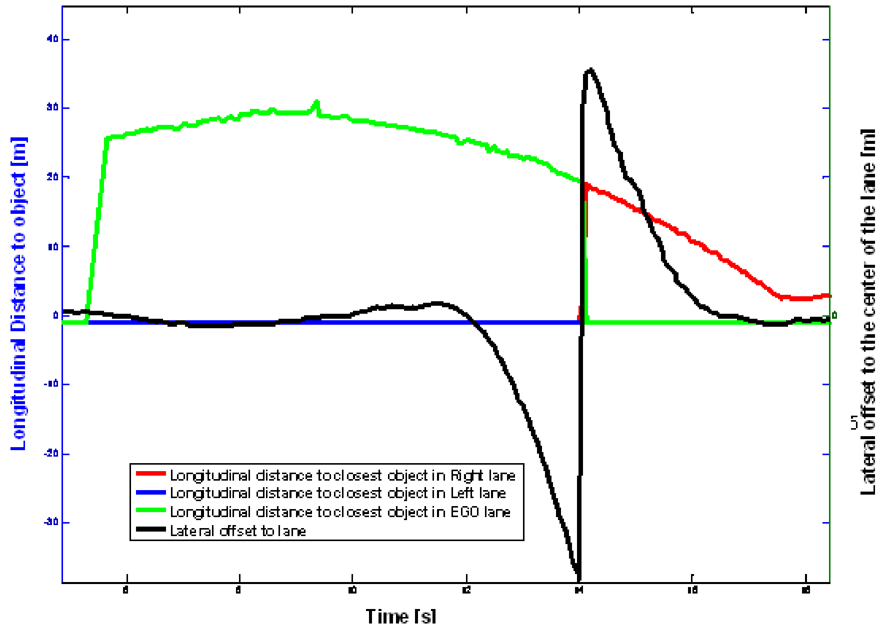


Figure 82: Closest distance of objects in adjacent and ego lane (Tracking of moving object in the same lane and lane change)

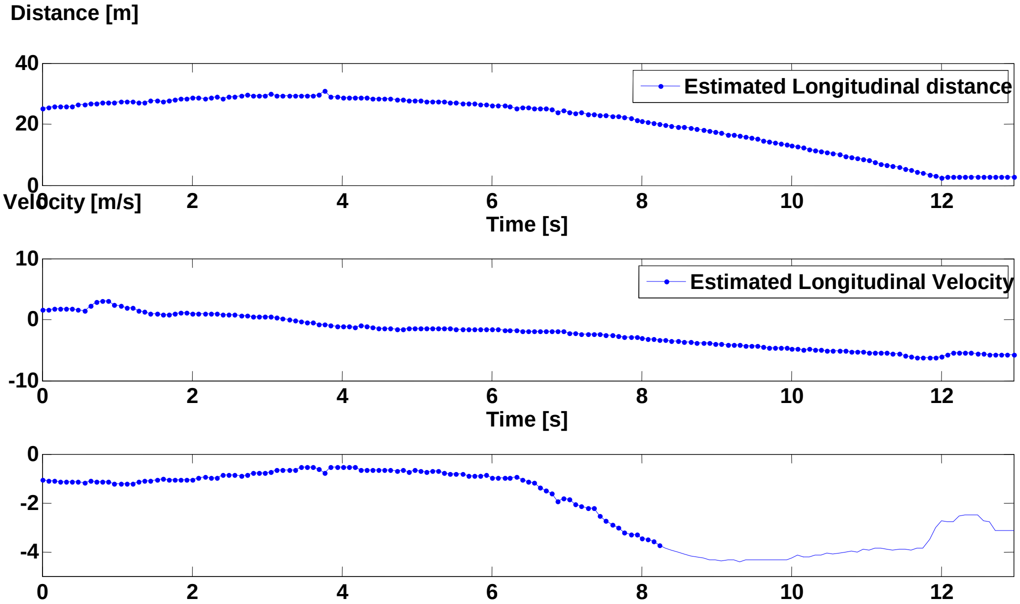


Figure 83: Tracking of object (tracking of moving object in the same lane and lane change)

- Lane Estimation Validation

In this test, the estimated lane geometry (coming from the lane detection camera only), was compared with the lane geometry coming from a very accurate map and DGPS position. Even if the errors of the DGPS (position and heading) and the map are accumulated to the overall calculated error, this comparison gives an estimation of the upper bound of error. As shown in Figure 84, up to a range of 65 meters the lateral estimation error is below 1m.

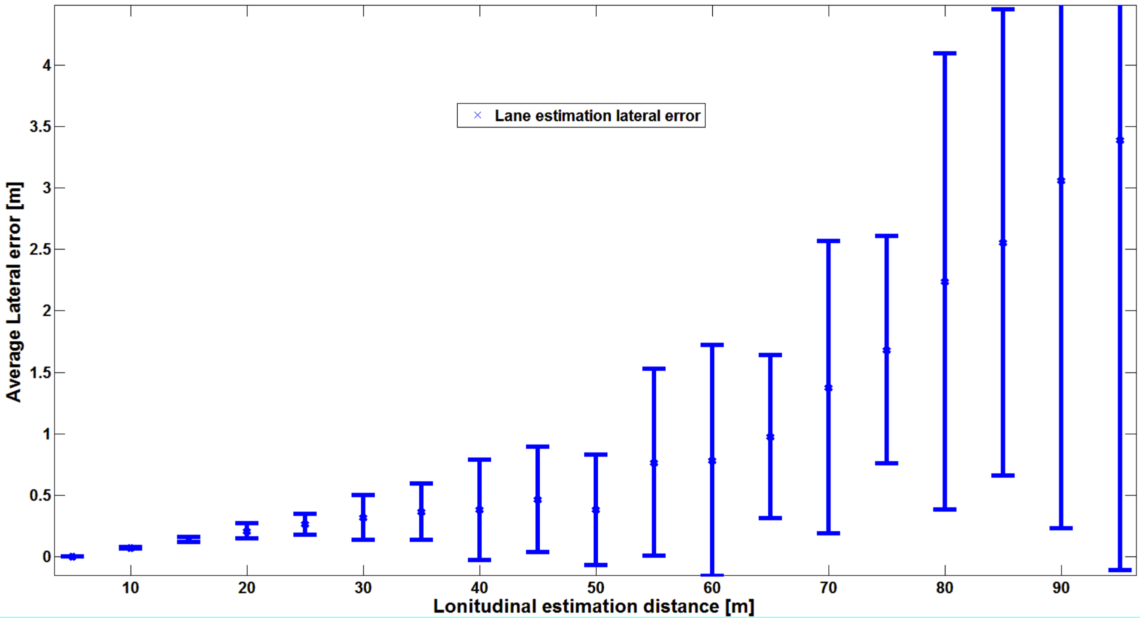


Figure 84: Lateral lane offset estimation error

The co-pilot algorithm needs accurate object to lane assignment, which means that the lateral lane offset estimation should be accurate enough in order to take appropriate action depending on object's position relative to the lanes. To improve accuracy for higher distances fusion of the lane tracker camera with the digital map information was implemented.

- Calculation time

For measuring the computation time of each process, the data fusion was analyzed with the profiling software AQtime 6. The fusion platform used the VTEC sensor setup of the AQuA demonstrator. The fusion algorithm consists of 4 local trackers, a global tracker and finally some filters. The results showed that the maximum calculation time (Max Time with Children) of the fusion algorithm (CFusionModule::RunFusion) is 24 milliseconds. The worst case can be calculated by adding all the fusion component execution times, except from the RunFusion that calls all the fusion modules, which results in a calculation time of 35 milliseconds.

5.1.2 Co-Pilot module of the Joint System

The Co-Pilot module is fundamentally intended to support the driver by identifying the current driving situation and providing a recommendation of the action to be done next. The coming action is a manoeuvre that has to be executed by the driver or by the vehicle controllers in a highly automated mode. There is also an evaluation whether the present situation can be mastered by the technical system or not. Hence, the Co-pilot is a piece of software that integrates several algorithms computing the safety envelope and the coming motion vectors.

This subsection summarizes the co-pilot’s architecture, its functioning and its operational scheme and interactions with the other sub-systems of SP3000. A technical description of the co-pilot system is also included as well as the algorithms used to generate safe and optimized trajectories. The selected trajectory is then used by the “Command Generation and Validation” sub-system in order to generate the command vector used by the vehicle controller to realize the feasible trajectory.

General objective of the “Co-Pilot” and “Command Generation and Validation” systems

From the results of the multi-sensor based fusion module, the “co-pilot” and “command generation and validation” algorithms identify the type of situation the vehicle is in and generate the best drive vector to handle the situation. If an emergency situation occurs, this vector will have a risk value attached to it. In practice, the system determines the optimal strategy to achieve the current driving objective accounting the driver state and the driving context.

The strategy determination is based upon the analysis of the current driving situation (based on both environment sensor information and future estimation). The assessment of the danger level of the current driving situation may lead the system to select a contingency strategy for the sake of safety. What is meant by “strategy” is a set of feasible optimal manoeuvres to be realized by the vehicle or the driver. Those are described in terms of vector commands and a geometric geo-positioning in space.

The optimality of the trajectories should be understood in terms of collision risks, respect of driving rules, optimal driving comfort, fuel consumption etc. Based on the calculation of the driving strategy, the co-pilot generates a trajectory. This is a path expressed in terms of spatial coordinates for given timestamps. It is effectively executed thanks to command generation and validation algorithms. On the other hand, the decision of executing the generated trajectory will be depending on the Mode Selection and Arbitration Unit (MSU) through the driver interface. These algorithms were evaluated on the Joint System Demonstrator (JSD) developed by DLR.

Inputs/outputs of the Co-Pilot module

The Co-pilot system needs sensors for the environment perception (vehicles, pedestrians, lane markings, traffic signs, obstacles, etc.) as well as for the driver state assessment (drowsiness, attentiveness, etc.). The following list summarizes the information involved with the co-pilot system[31].

Input data

- Data coming from the data fusion system
  - Ego vehicle information: Timestamp, kinematic state information, variance of the estimated values, position relative to the point the trajectory was calculated. geodetic position, lane assignment, standstill detection, sensors that provided the data for extracting the information
  - Perception model:
    - Objects: track information: ID and track lifetime, position and kinematic state estimation, variance of the estimated values, object classification and dimensions (width, height, object type), obstacle probability, lane assignment, number of detected obstacles, sensors that provided the data for extracting the information for each object.
    - Lanes: Position: lateral and longitudinal position relative to the ego vehicle, curvature and curvature rate, width, lane type (soft, emergency, hard etc.), left and right lane marking type, observed as traversable; observer from map, lane reliability
    - Road information: road type (unknown, highway, rural, urban), road type additional (none, ramp, additional map-based information), gradient, warning signs, sensors that provided the data for extracting the information for each object.
    - Trip information: time and trip duration
- Data from the driver interface system
  - Driver's primary task command: position of the acceleration pedal, steering wheel angle, steering wheel rate, steering wheel torque, vehicle speed, yaw rate, longitudinal acceleration, lateral acceleration, lateral speed

Output Data

- List of ranked geometric trajectories: Each trajectory is a set of geo-referenced and locally referenced 2D positions of the vehicle. It is also described by a set of basic controls (linear speed, steering or yaw angle).
  - Relevant targets and key environment information (application specific)
  - Ranking of trajectories (ranking of each trajectory, cost (and sub costs) associated to each trajectory)
  - Vehicle positioning, current executed trajectory and trajectory limits (current local position : lateral position; current local position : heading control; current trajectory; lateral boundary of the trajectory
  - Limits of the free space and limits of the health horizon

[31] For more detailed description of the interactors please refer to the architecture deliverable D12.1.

- Set of ranked trajectories with descriptions (maneuver description, longitudinal position in the lane, lateral position with respect to the lane, time position, length of the trajectories, ranking of each trajectory, cost (and sub-costs), lateral position of the lane, longitudinal position of the lane, desired velocity, desired lateral position in the lane

5.1.2.1 Co-Pilot algorithms - driving strategy and trajectory

The co-pilot is an advanced path planner that aims at constantly elaborating an “optimal” trajectory for the vehicle. Optimality is to be understood as a compromise that takes into account several needs and constraints: comfort, fuel consumption, respect of driving rules, safety, the length of the path, the speed along the path, etc. The main output of the co-pilot sub-system is a trajectory that is described in terms of a set of points and positions, endowed with the intended vehicle dynamics.

In order to achieve a strong cooperation with the driver, irrespective of the automation level, the co-pilot process is achieved using two main functionalities:

- *The definition of a driving strategy* at a high level, which is described using a manoeuvre language (e.g. Loeper & Flemisch 2009[32]). Three different approaches are evaluated and then fused in order to give a more reliable result.
- *The definition of trajectory* at a lower level: This function uses the previously selected manoeuvre to define a trajectory.

To these functionalities, a third not less important function is the generation of the motion command vector to be transmitted to the drivetrain (actuators of the execution layer) in order to execute effectively the computed trajectory.

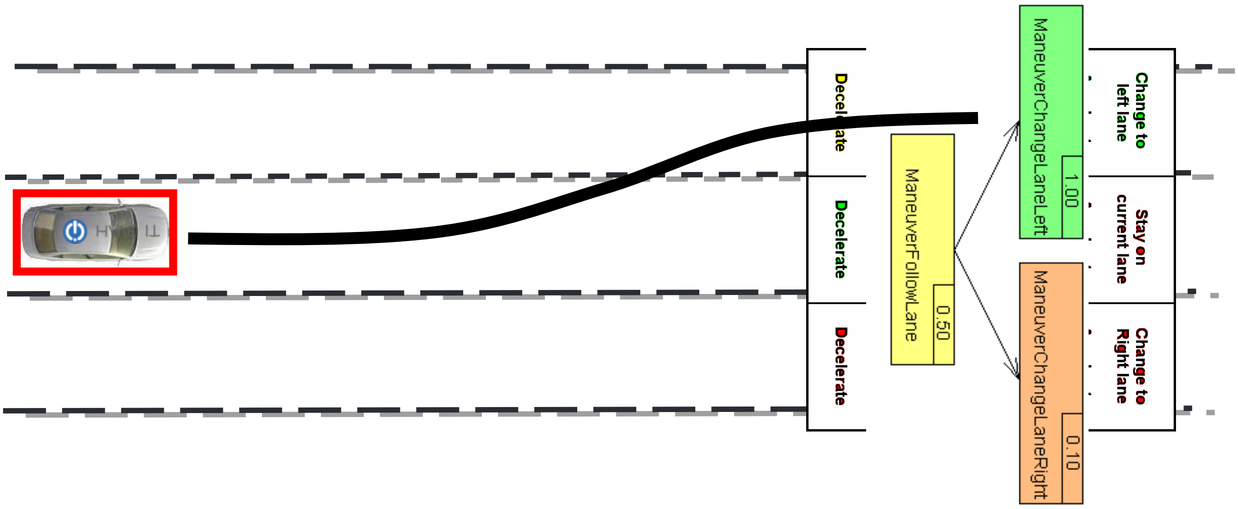


Figure 85: Coarse planning (manoeuvre level). Here, the next manoeuvre to execute is to change lane to the left and accelerate

[32] Löper, C.; Flemisch, F.: Ein Baustein für hochautomatisiertes Fahren: Kooperative, manöverbasierte Automation in den Projekten H-Mode und HAVEit; 6. Workshop Fahrerassistenzsysteme; Löwenstein; 2009

5.1.2.2 Coarse planning – the strategy level

The *definition of a driving strategy at a high level* sub-module is based on fast and simple algorithms that evaluate the possibility of performing several predefined manoeuvres. The aim of this high level is to quickly eliminate a part of the search space, thus reducing the calculation time of the *definition of trajectory* at low level. It also allows a high level communication towards the driver, in the form of a *manoeuvre grid* or a *manoeuvre tree.*

Nine manoeuvre cases can be identified by combining following basic actions: Three actions in longitudinal direction (accelerating, decelerating and staying at constant speed) and three actions in lateral direction (staying in the same lane, going to the left and to the right). Some examples of these manoeuvres are “stay in the same lane and decelerate” or “change to the left lane and accelerate (Figure 85)”.

To these basic manoeuvres, an “emergency brake” manoeuvre is added, corresponding to a full brake till standstill, and a “minimum risk state manoeuvre”, corresponding to a user- or system-initiated slow-down on a dedicated lane. This gives a total of eleven manoeuvres. Each manoeuvre gets a performance indication which is called *valential.* What is meant by performance is a global qualitative and quantitative evaluation of the manoeuvre accounting given criteria. The valential is used to directly discard certain areas in the solution space and to give a clear overview of the situation in the human machine interface (HMI).

There are two ways to represent these manoeuvres: the *grid* which is a 3x3 matrix, giving 9 cases for the 9 basic manoeuvres plus 1 case for the emergency brake manoeuvre and 1 for the minimum risk state manoeuvre. Each case is coloured according to its valential (red to orange to yellow to green for increasing valentials). The *tree* representation gives the current situation, and it visualizes the possible actions with their valential as the branches of this tree. In the following sub-sections we will describe the different manoeuvre algorithms as well as the fusion of the three algorithms developed by partners IFSTTAR and DLR (e.g. Loeper 2011).

Manoeuvre grid calculation (IFSTTAR)

The manoeuvre grid builds a solution space as the combination of three longitudinal actions and three lateral actions (Figure 86). On the longitudinal side the vehicle can be decelerated, accelerated or held in the current speed range. In the lateral direction, the vehicle can change lanes to the right, to the left, or can stay in the same lane. To these nine manoeuvres, a minimum risk state (MRS) manoeuvre is added, which corresponds with stopping on the right most lane, and an emergency manoeuvre, corresponding with maximal braking till standstill.

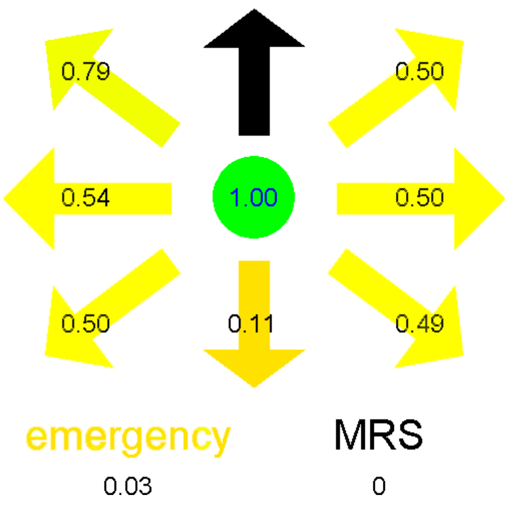


Figure 86: Manoeuvre grid

The manoeuvre grid algorithm attributes a performance to each of the presented manoeuvres. This is done through the evaluation of a set of performance indicators that are linked with the aspects of good driving. The algorithm measures the risk of collisions with other objects if the manoeuvre would be executed. This includes the risk of slipping in curves, the risk of going off the road, the speed proposed by the manoeuvre, the offences against the speed limits, the accelerations and decelerations needed to execute the manoeuvre. Moreover, it checks whether full lane marks are crossed and if other vehicles are right overtaken. The total performance of each manoeuvre is the weighted sum of the different performance indicators. Manoeuvres that correspond to a fast and smooth drive without risk or offence against the traffic rules are promoted in the ranking of the manoeuvre grid. The total performance indicator is equivalent to the valential presented above; it is used by the manoeuvre fusion algorithm (Figure 89). The weights used to calculate the total performance indicator allow tuning the character of the co-pilot. The algorithm has been tested both in simulation and on a real vehicle. The calculations are very fast (3 to 4 milliseconds).

Manoeuvre grid calculation II (IFSTTAR)

The aim of this solution is to build a faster manoeuvre grid algorithm than the one specified above, by only considering the collision risk. The output of this grid is a ranking of the manoeuvres and the target speed for the best manoeuvre. It does not compute an absolute risk. The risk is evaluated as the product of two components: The possibility of the collision and the gravity of the collision if it occurs:

- *Evaluation of the gravity*: The gravity is evaluated using the scale (MAIS[33]). We have to compute the Equivalent Energetic Speed of the collision (EES).
- *Evaluation of the possibility*: In order to cover all the possibility space, we use two situation descriptors: The first one is Time-To-Collision (TTC), the relative distance divided by the relative speed. This indicator mainly addresses fast approaching vehicles; it is unable to give a possibility of collision when the vehicles are close and approximately at the same speed. So the second indicator is the Inter-Vehicular-Time (TIV) which is good at describing dangerous situations when vehicles are close. A transfer function associates from the TIV and TTC a possibility of collision. These functions are defined using the results of previous projects.
- *Evaluation of the risk*: The risk is directly evaluated as the product of the two parameters gravity and possibility. It is evaluated for a large range of speeds of the ego vehicle (from 0 km/h up to 150 km/h) in the lane, so considering the front and rear vehicle. The output is a curve that describes for each possible speed of the vehicle the associated risk. The process is the same on the adjacent lane, considering a virtual vehicle at the same level.

The risk curves are defined relative to the value of the manoeuvre that is being executed. At the end of the algorithm, the risk curves are plotted in a second manoeuvre grid. The risk is converted into a “valential” to allow an easy comparison with the first manoeuvre grid and the manoeuvre tree[34]. The algorithm has been tested in simulation and on the vehicle. The calculations are very fast (1-2 milliseconds) and also speed up the trajectory modules by eliminating highly risky zones in the solution space of the trajectory planners.

[33] Mills and Hobbs; Mills, P.J.; Hobbs, C.A.: “The probability of injury to car occupants in frontal and side impacts”. In: Stapp Car Crash Conf, Chicago, IL, pp. 223–232, 1984

[34] For detailed algorithm description, see: Vanholme, B.; Glaser, S; Mammar, S, Gruyer, D.: “Manoeuvre based trajectory planning for highly autonomous vehicles on real road with traffic”, In Proceedings of ECC’09, Budapest, Hungary, 23-26 August 2009

Manoeuvre tree calculation (DLR)

The manoeuvre tree (see Figure 87) offers an integrated representation of the current action of the vehicle and possible future actions regarding to the current situation, starting with the current manoeuvre as the root of the tree. Feasible manoeuvres which can possibly follow the current manoeuvre are located as leaves in the tree. A quality rating called valential is assigned to all feasible manoeuvres to show the preferences of the automation. The manoeuvre with the highest valential is the preferred manoeuvre of the automation. If the valential of the current manoeuvre is greater than zero the automation can also continue executing this manoeuvre. The definition of a manoeuvre is compatible to a human driver's definition of a manoeuvre.

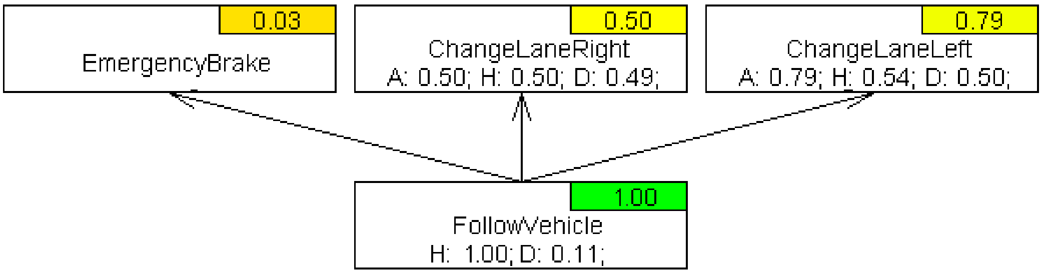


Figure 87: Manoeuvre tree

Figure 88 shows a block diagram of the algorithm used for generating the manoeuvre tree. Because the algorithm is based on fuzzy logic the first step is the fuzzification of the relevant input data. The recognition of the current manoeuvre of the vehicle is done by using a fuzzy rule set containing one rule for each manoeuvre. For the determination of the feasible manoeuvres from the automation's point of view also a fuzzy rule set with one rule for each manoeuvre is employed. For every current feasible manoeuvre a valential is calculated. The calculation is based on a paired comparison of the feasible manoeuvres by means of fuzzy rules. The last step of the algorithm is the assembling of the manoeuvre tree. The current manoeuvre of the vehicle is placed as root manoeuvre; the other feasible manoeuvres are located as leaves in the tree.

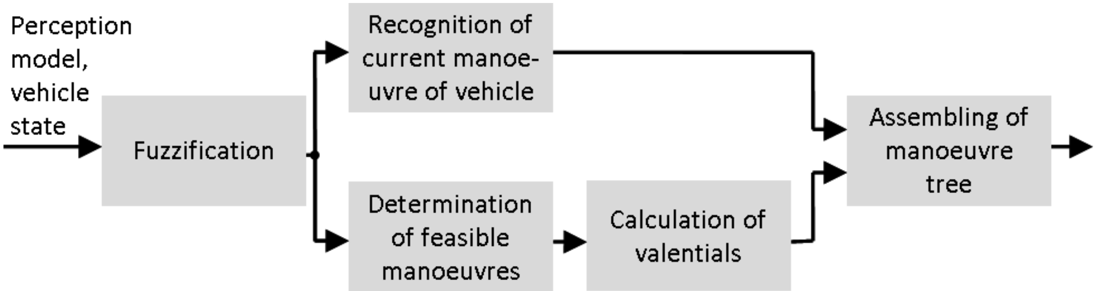


Figure 88: Block diagram of the algorithm for manoeuvre tree computation

Fusion of the manoeuvre systems

The two LCPC algorithms and the DLR algorithm run in parallel and at the same frequency. After comparing the performance of both systems, it was decided to fuse the information coming from both systems according the scheme described in Figure 89. The output of these three algorithms is fused by a voting module into a single output. The very different nature of these three algorithms creates true robustness and reliability to this safety relevant component.

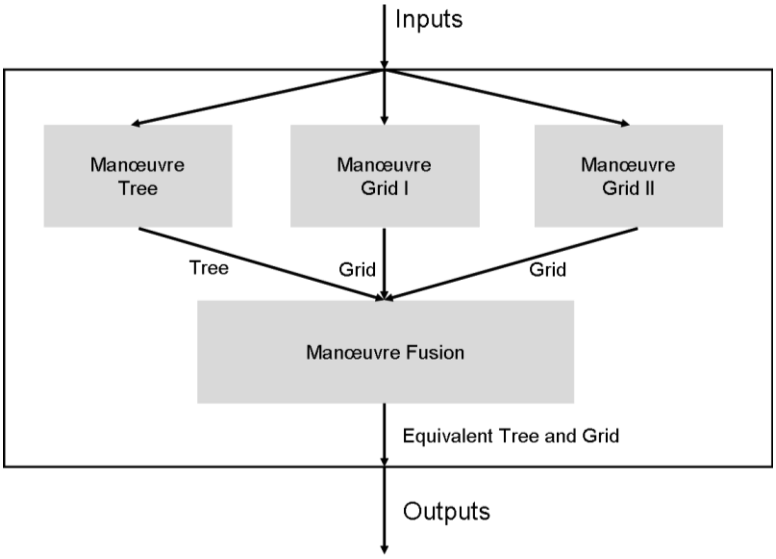


Figure 89: The fusion of the three manoeuvre algorithms

As there is a unique translation between the tree and the grid format, the fusion module output can be given in both representations. Both representations are fully equivalent and both can be delivered to the client modules.

Performance measurement

Performance measurements of these algorithms were done in a simulation environment. We isolated the component from the influence of other functions by providing an ideal simulated data fusion and manually drove the vehicle along the relevant use cases. After the drive session, the following output variables with a direct link with the performance of this function were analyzed on their correctness and stability:

- The longitudinal and lateral component of the best manoeuvre
- The valential of the best manoeuvre

Figure 90, Figure 91 and Figure 92 show the performance plots of the (fused) manoeuvre algorithms during a complex scenario that incorporates several HAVEit use cases, such as avoidance of right over passing of obstacles, left over passing of obstacles, driving on the right most lane and speed limitation. The functionality for all use cases was met. The plots show a consistent grid and tree representation of the best manoeuvre; the “pink” grid and “black” tree plots overlap completely. The longitudinal and lateral components of the best manoeuvre were stable. Their valential was kept under control and close to the maximum value of 1, meaning that the algorithms always found a good solution.

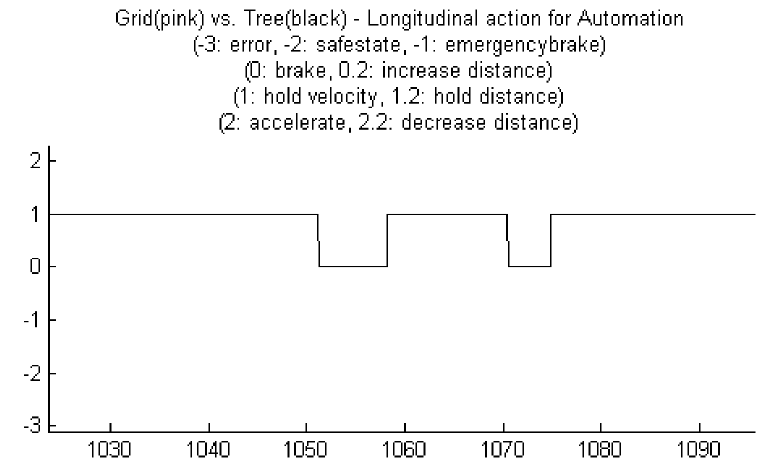


Figure 90: Manoeuvre grid and tree: longitudinal component of best action found by automation

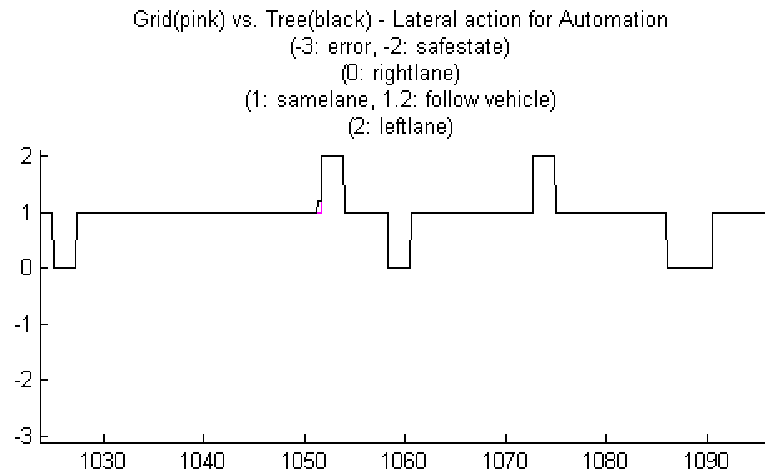


Figure 91: Manoeuvre grid and tree: lateral component of best action found by automation

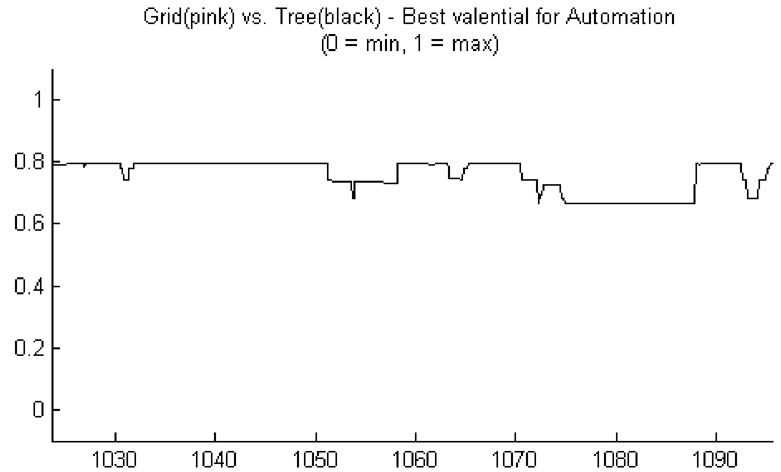


Figure 92: Manoeuvre grid and tree: valential of best action found by automation

5.1.3 *The trajectory planner*

The trajectory sub-module uses the output of the manoeuvre sub-module to generate a detailed spatio-temporal description of the optimal trajectory. A total of 3 trajectories (one per lane) are generated in the current implementation. The trajectory that will be used by the controller corresponds to the manoeuvre with the biggest valential given by the manoeuvre algorithm. The trajectory sub-module will not decide to change the manoeuvre performed by the controller even if it detects a collision since that task is supposed to be assigned to the manoeuvre sub-module.

In practice, once a coarse plan has been defined, a specific motion plan is assigned to the vehicle. This motion plan defines the state of the vehicle for the future time instants. This sequence of desired states in time is the so-called trajectory.

Partial motion planner – PMP (INRIA)

In order to guarantee the safe motion of the vehicle, when computing the trajectory, the vehicle has to correctly consider its own limitations and the future movement of the other vehicles. The approach taken in HAVEit follows the work described by Petti[35], Fraichard[36] and Benenson[37].

Since the vehicle has a limited visibility, its plans can only reach a limited horizon. Since a wall (traffic jam, road blockage, etc...) could exist on the front of the unobserved areas all trajectories are required to stop before reaching the end of the visibility region. This is treated as an end of road situation (dead end). When the observed region is updated, the trajectory is also updated. Because of the partial nature of the provided trajectory, we call this approach *Partial Motion Planning*. In order to ensure that the trajectory is feasible by the vehicle, trajectory generation strategy is based on a search in the command space (Figure 93).

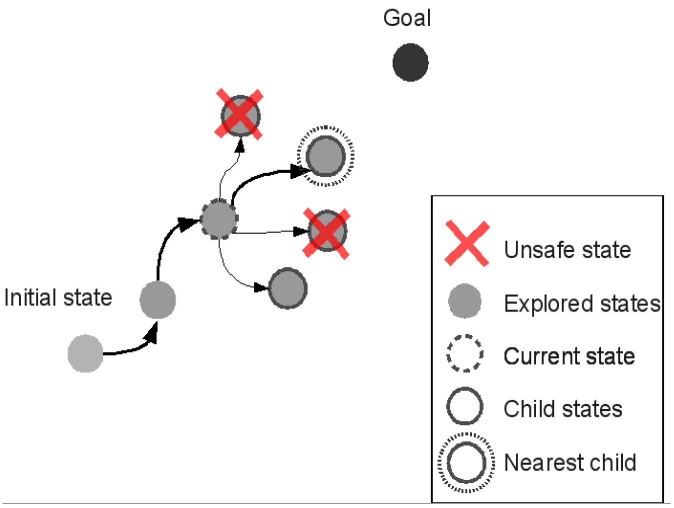


Figure 93: Construction of the sequence of states in time in PMP

Given an initial vehicle state (position, speed, steering), the PMP algorithm searches the set of commands that will allow to reach at best the goal. The model used to integrate the effect of a sequence of commands takes into account the saturation of the vehicle in steering and

[35] Petti, S.: "Safe navigation within dynamic environments: a partial motion planning approach," Ph.D. dissertation, Ecole des Mines de Paris, 2007.
[36] Fraichard, T.: "A short paper about motion safety," in Proceedings of the IEEE International Conference on Robotics and Automation, 2007.
[37] Fraichard, T.: "A short paper about motion safety," in Proceedings of the IEEE International Conference on Robotics and Automation, 2007.

acceleration. Also, for any given state of the partial trajectory it is verified that the vehicle is capable of stopping without having a collision.

By doing so, the PMP algorithm ensures that at any time the solution available will not actively cause a collision. In order to provide this guarantee it is necessary to use a conservative prediction of the vehicle's surroundings. Using a full search on a discrete commands space, using a continuous curvature distance metric to reach a specific goal and doing brute force collisions checking has been shown to provide satisfactory results.

However, the HAVEit project presents specific needs and previous work requires some adaptation. First, the driving in HAVEit is modeled as operations on lanes. The goals and obstacles are defined as presence on lanes. This provides a coarser (faster) spatial sampling for collision checking and simplifies the distance metric to the goal “How far are we from reaching the centre of the desired lane?” The diagram in Figure 94 illustrates the data flow of the simplified Partial Motion Planning algorithm.

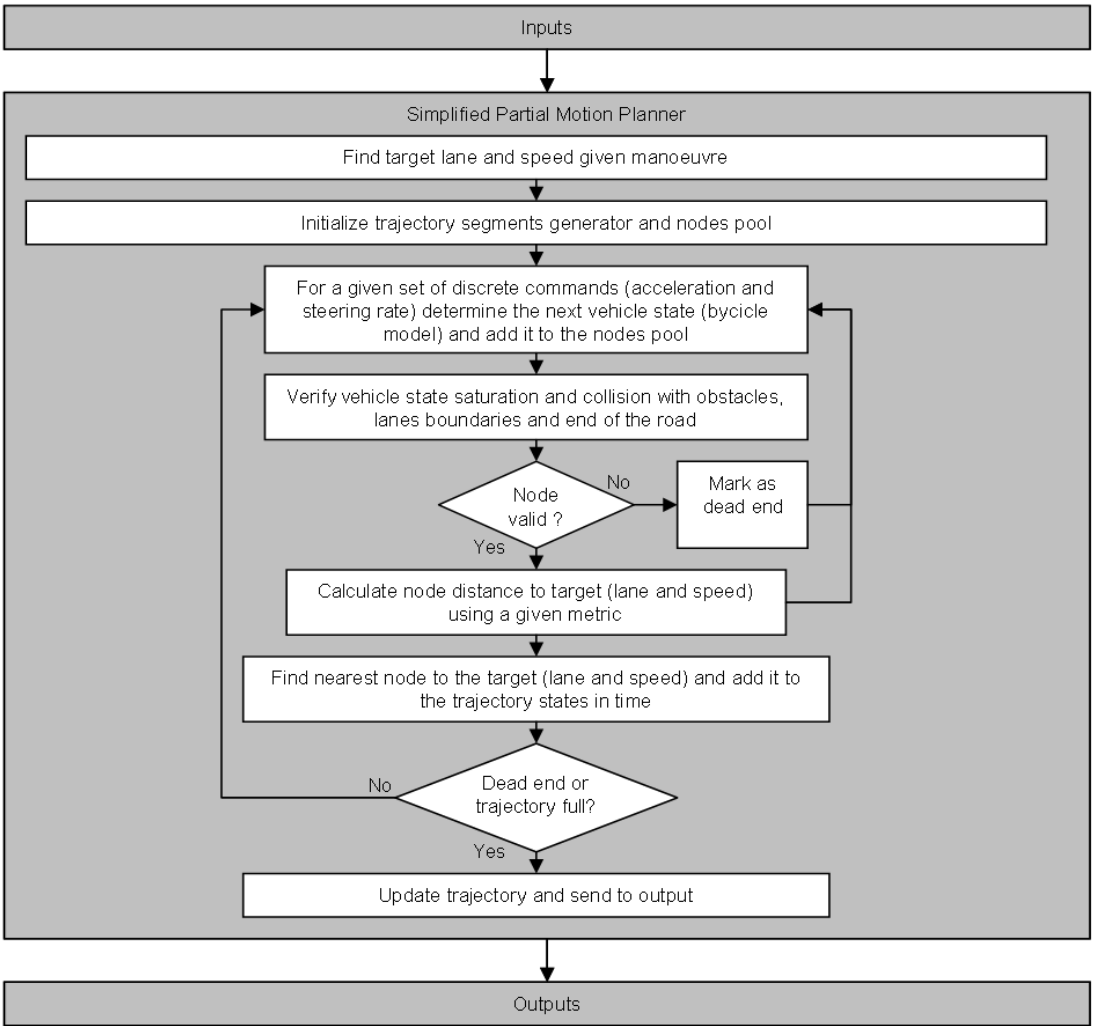


Figure 94: Simplified partial motion planner block diagram

Instead of searching a trajectory that avoids the obstacles and reaches the goal as best as possible by any means, a simplified approach is used: The trajectory goes straight towards the desired lane and stops if any obstacle is present. The circumvention of obstacles is prohibited since this responsibility is delegated to the strategy level (manoeuvres) that will decide the sequence of lane changes required to circumvent an obstacle. These simplifications allowed a more efficient implementation, in code size, memory usage and computation time.

The described algorithm is implemented in pure C with static memory allocation. When running on a 2 GHz computer the simplified implementation of the partial motion planner provides full trajectories (see Figure 95) in up to 40 milliseconds, enough for real time operation.

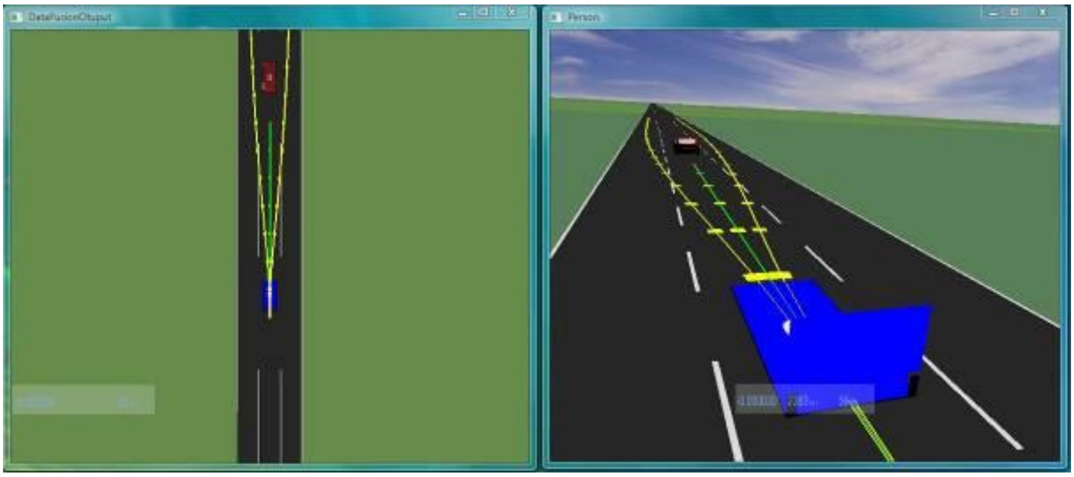


Figure 95: Simplified Partial Motion Planner trajectories in the HAVEit JS Framework

Quintic polynomials (INRIA)

This algorithm uses a mathematical function that provides a geometric modeling (polynomial) of the vehicle trajectory that responds to the realistic demands of the manoeuvre to be performed. The advantage of this approach is that it is faster to run; however, it can lead to behaviors that cannot be achieved by a vehicle. To model the geometric path during a lane change, literature often shows approaches using models of fifth degree polynomials[38,39].

By choosing a 5th degree (quintic) polynomial, a third degree behavior is assured for the longitudinal and lateral accelerations. A function of third degree is the minimum degree that can ensure realistic behavior of the two acceleration components. So the position of the vehicle must follow a function of 5th degree in the direction longitudinal X and lateral Y. The following Figure 96 shows an example of a typical lane change relative to a system of axis of reference [X, Y].

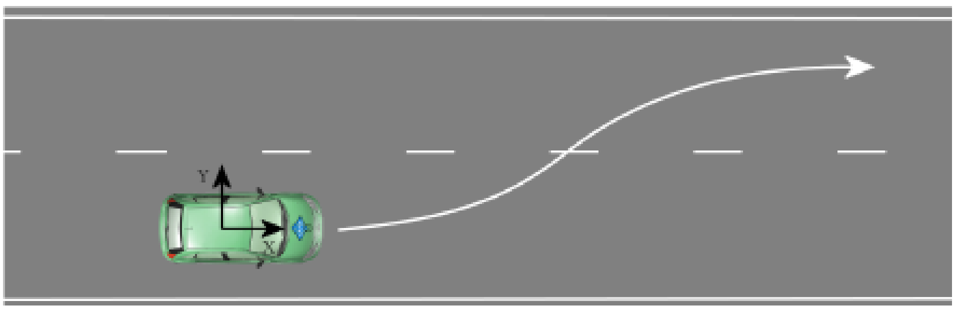


Figure 96: Example of a lane change

[38] Papadimitriou, I.; Tomizuka, M.: "Fast lane changing computations using polynomials", American Control Conference, 4-6 June 2003

[39] Shamir, T.: "How should an autonomous vehicle overtake a slower moving vehicle: design and analysis of an optimal trajectory", IEEE Transactions on Automatic Control, April 2004

X(t) and Y(t) will have the following formula functions of time t:

$$X(t) = A_5 t^5 + A_4 t^4 + A_3 t^3 + A_2 t^2 + A_1 t + A_0$$

$$Y(t) = B_5 t^5 + B_4 t^4 + B_3 t^3 + B_2 t^2 + B_1 t + B_0$$

The equation coefficients (A's and B's) are determined by specifying limit conditions for the lateral and longitudinal values of position, velocity and acceleration. After the geometric model coefficients are determined, X(t) and Y(t) are calculated. These points are calculated in a way that they are spaced of half of the length of the vehicle (maximum) to ensure that there is no free space between two consecutive states. This will be useful for the collision checking verification to ensure that there are no collisions between states.

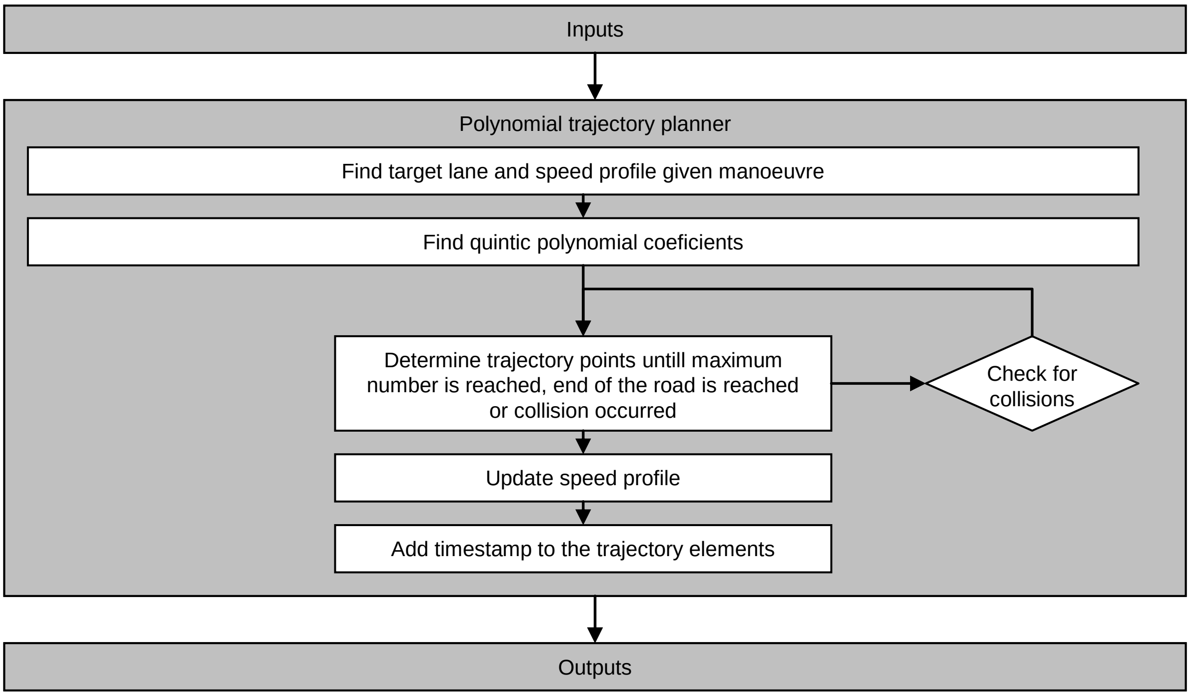


Figure 97: Polynomial trajectory planner block diagram

This calculated points are then added to the trajectory until the trajectory is "full" (maximum number of elements is reached), "lane target" or "end of road" is reached. If the lane target is reached and the trajectory is not full then the remaining trajectory elements are filled in with the target lane points.

The collision checking is done while filling in the trajectory with the points, and the speed profile is updated adequately according with the manoeuvre and trajectory constraints (e.g. lateral acceleration). In the end, a timestamp is attributed to every element of the trajectory according to the trajectories elements position and velocities.

The diagram in Figure 97 illustrates the data flow of the polynomial trajectory planning algorithm. The algorithm described is implemented in pure C-code with static memory allocation. When running on a 2 GHz computer the polynomial trajectory planner provides full trajectories in a few milliseconds which is perfect for real time operation. Figure 98 presents an example of trajectory calculation based on 5th order polynomials.

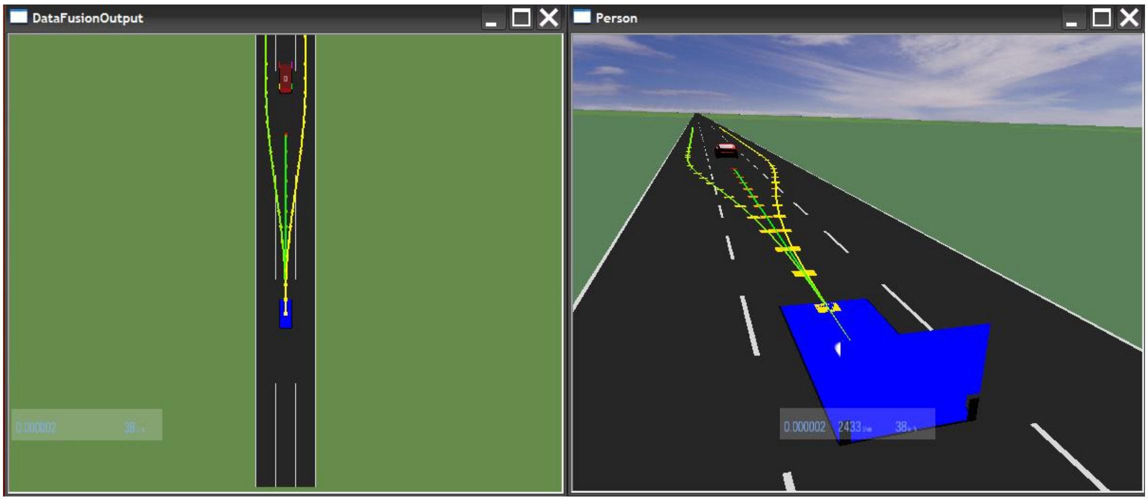


Figure 98: Polynomial trajectory planner trajectories in the HAVEit Joint System software framework

Multi-criteria polynomial planner (IFSTTAR)

The manoeuvre grid I algorithm described above leads directly to a multi-criteria polynomial trajectory planner. The manoeuvre grid can be seen as a limited solution space with eleven trajectories. Internally, each manoeuvre is indeed represented as a polynomial trajectory that connects the actual vehicle state to a future vehicle state, with a target speed and a target lane, corresponding to the manoeuvre. The calculation of the performance indicator "Valencial" is done through these polynomial trajectories.

For the trajectory planner, the solution space is refined from eleven trajectories to one hundred. This means that the solution space is still discrete, but it is fine enough to find a (sub-)optimal trajectory with a good performance and to have soft transitions between the different trajectories.

The algorithm has been tested in simulation and on the vehicle. The calculations are very fast (10 to 20 milliseconds) and are able to handle the complete set of SP3000 scenarios. Figure 99 shows the algorithm output on the Joint System HMI. The optimum is indicated in green. Other trajectories can be outputs for testing purposes, e.g. an emergency trajectory which is shown on the left side of the optimal trajectory in this figure.

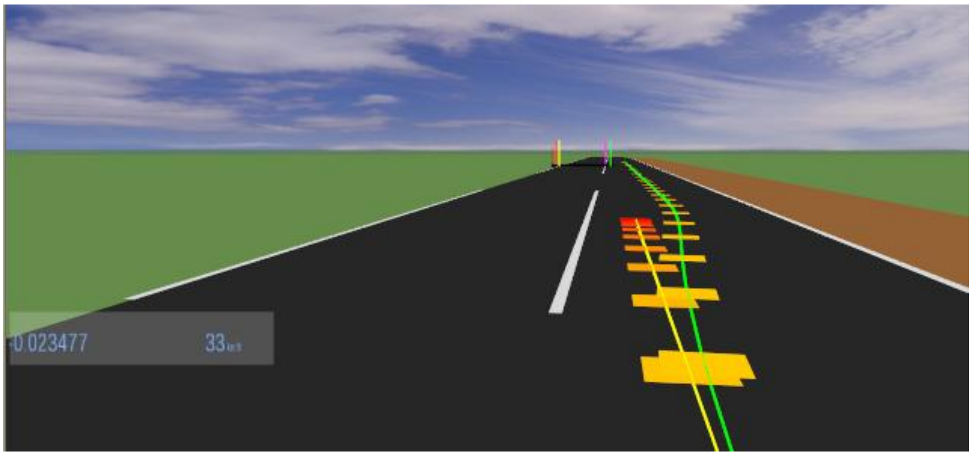


Figure 99: The multi-criteria trajectory planner: the emergency trajectory (left, yellow), the optimal trajectory (right, green)

Performance measurement

Performance measurements for the algorithms were made in simulation environment. The trajectory planner was isolated from the influence of other functions by providing an ideal simulated data fusion. The best manoeuvre has been chosen manually and the vehicle was manually driven along the different use cases.

After the driving session, the following output variables with a direct link with the performance of the trajectory planner ware analyzed with respect to their correctness and stability:

- Consistency of the trajectory with the chosen manoeuvre
- The number of spatio-temporal points that describe the trajectory (indicating that a full trajectory has been found)

Figure 100 and Figure 101 indicate the good consistency of the best (or chosen) trajectory and the best (or chosen) manoeuvre and the number of trajectory elements during the scenario described for the manoeuvre performance measurement.

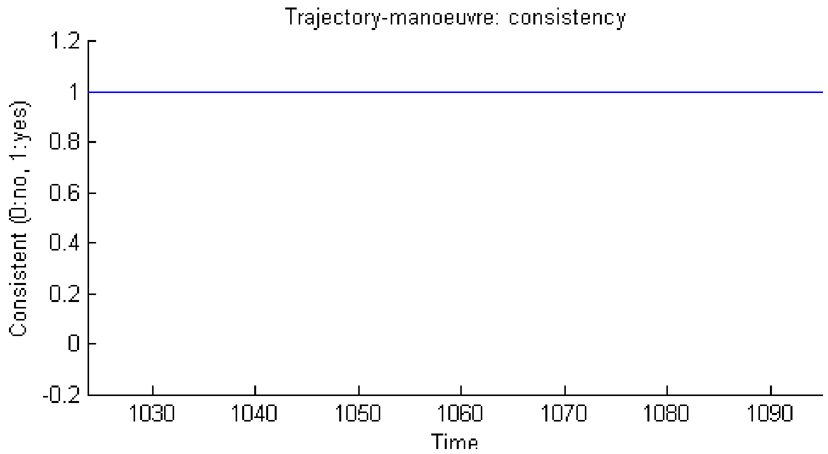


Figure 100: Consistency of the trajectory with the chosen manoeuvre

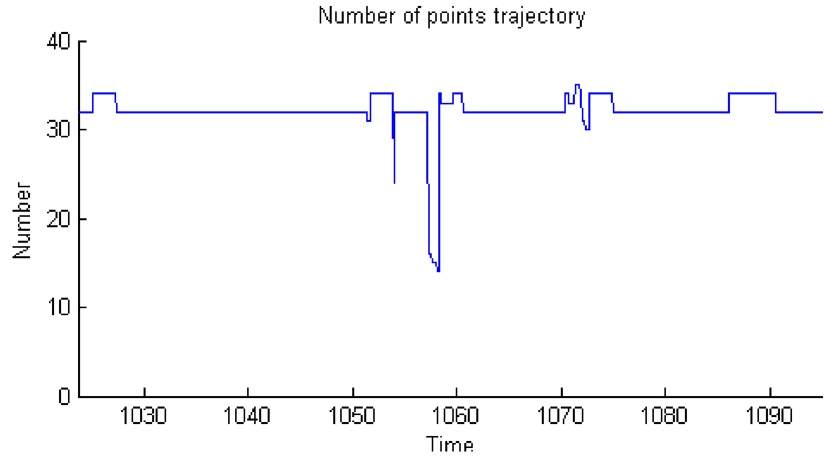


Figure 101: Number of spatio-temporal points that describe the trajectory

5.1.4 Command generation and validation

The algorithms presented in this section compute the vehicle actions to be carried out in order to achieve the required automation (or warning) task according to all inputs (automation level, driver automation level request, set of trajectories, relevant detected targets, vehicle positioning, trajectory & state limits, vehicle state). The outputs of this function are primarily the acceleration and the steering angle (or torque) to be applied to the vehicle actuators.

A straightforward idea consisting of decoupling the longitudinal dynamics and lateral dynamics, under some simplification hypothesis, can lead to a substantial simplification of the controller synthesis phase. Indeed, with these hypotheses, the vehicle model can be divided into two linear sub-models, longitudinal and lateral, each of which is controlled by a separate control organ, acceleration and braking pedals to control the longitudinal dynamics and the steering wheel to control the lateral dynamics (see Figure 102). In this case linear robust controller synthesis techniques can be used.

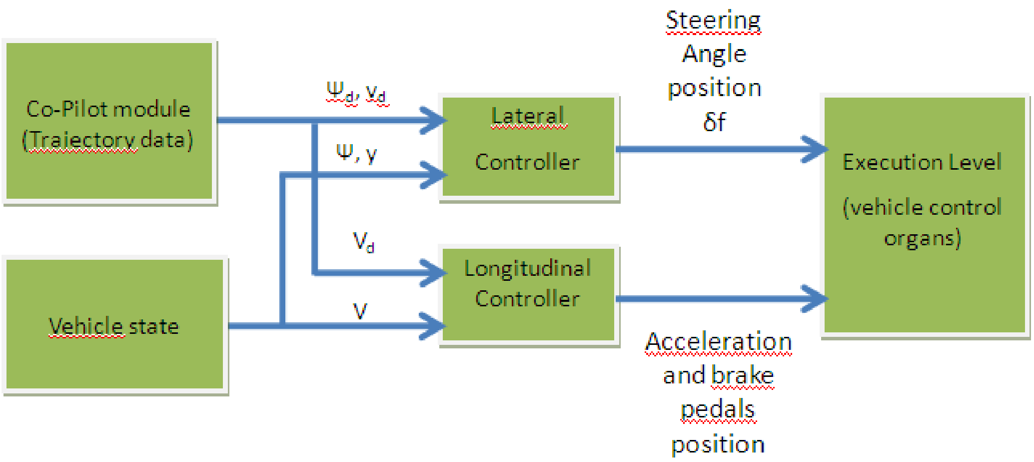


Figure 102: A non-coupled longitudinal / lateral control

As the function is also aware of the driver desired inputs, it computes the distance between the two vehicle commands (from the automation and from the driver). Moreover, the function indicates through an interactor output the action to be committed. Figure 103 shows the block diagram and data flow of the component needed to compute the vehicle actions in order to keep the vehicle close to the planned trajectories.

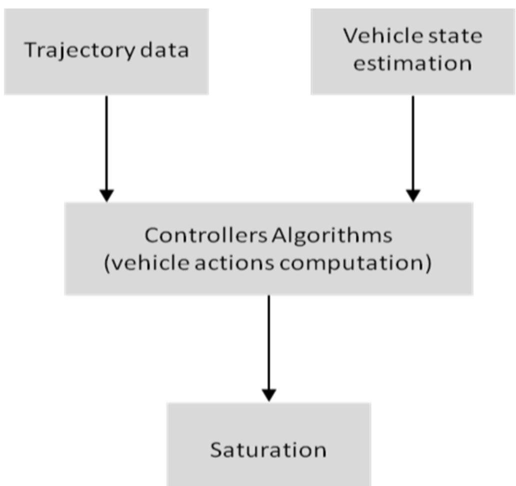


Figure 103: Block diagram of computing vehicle action process

This needs to estimate the vehicle state, in particular lateral position, orientation and velocity, and compare them with their desired values provided by the trajectories planner module. The relative error between the actual state of the vehicle and where it should be is computed. Hence, the adequate controller actions needed to bring the vehicle state to the desired one are generated. Before transmitting the computed controller actions to the actuators, they are passed through a saturation module in order to constrain their magnitudes and also their rate of change.

Common estimation algorithm

This algorithm estimates the position and the heading of the ego vehicle relative to an initial coordinate system (X_1, Y_1). The main use of this module is for estimating the position of the vehicle with respect to a ground fixed trajectory shown in Figure 104.

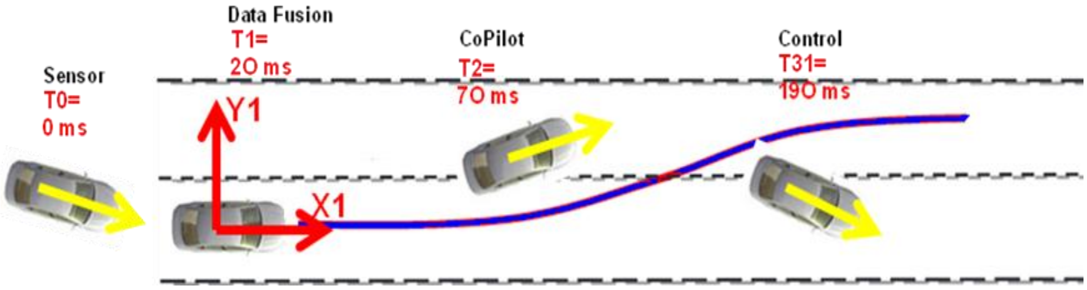


Figure 104: Overview of coordinate system and timing

The co-pilot module adds a timestamp T1 to its trajectories, which corresponds to the start of the calculation of the environment by the data fusion. The position of the ego vehicle at that time is expressed by the coordinate axis (X_1, Y_1), which serves as a reference axis for all calculations. Every time the displacement filter is called with an input timestamp T>T1, the estimated position relative to the coordinate system (X_1, Y_1) is given. Since the calculated trajectories are also expressed in the (X_1, Y_1) coordinate system, the controller can calculate the vehicle's position relative to the trajectory.

The displacement filter in Figure 105 and Figure 106 is used to estimate the position of the vehicle, relatively to an initial coordinate system. The coordinate system is set every time the filter is reset. When the filter is called, a Kalman based algorithm estimates the position and the heading relative to the initial coordinate system. In addition, the filter can be updated by the sensor data (speed, yaw rate) so that the estimation remains accurate even if the vehicle executes highly dynamic manoeuvres in the longitudinal or the lateral axis.

Longitudinal control

The controller (Figure 107) is designed to keep the vehicle speed close to a desired value. Regarding to this problem, it seems to be sufficient for the first time to use a proportional and integral (PI) controller which can reach the desired velocity and eliminate the steady error. Nevertheless, the problem of this controller is that the integral action can make the controller reaction time large, especially when the actuator saturates, i.e. windup phenomena. In order to prevent this drawback, an anti-windup action is integrated to this controller to supervise the integral action and to constrain the computed controller action to stay close to the actuator saturation values. So, the anti-windup action is enabled when the computed controller action reaches the actuators limits.

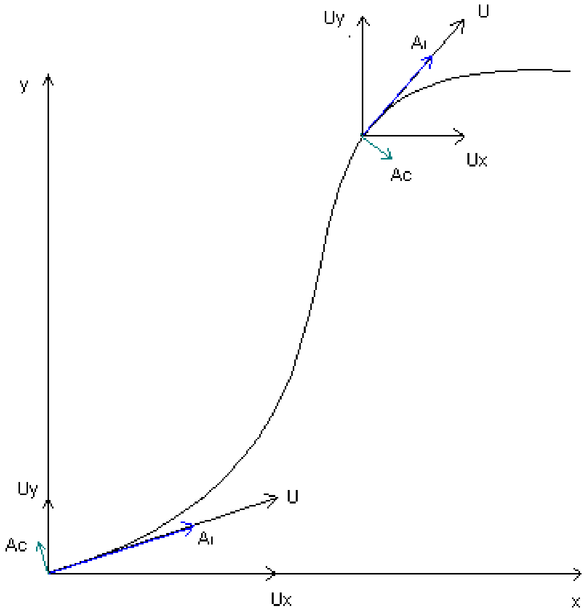


Figure 105: Displacement filter

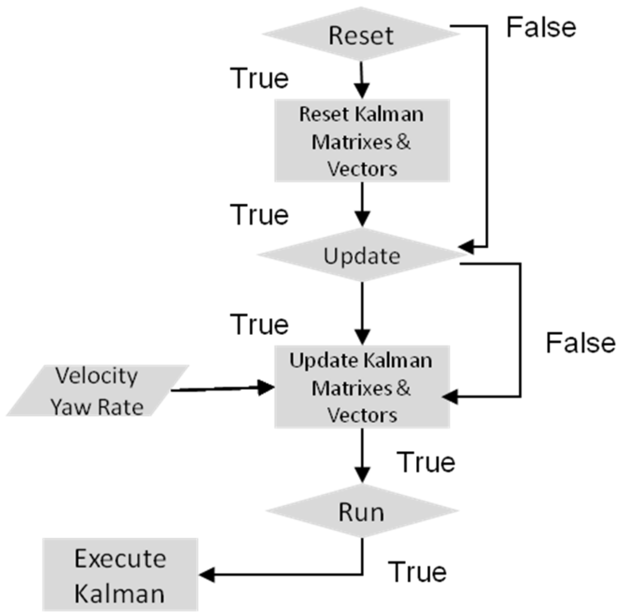


Figure 106: Displacement estimation algorithm

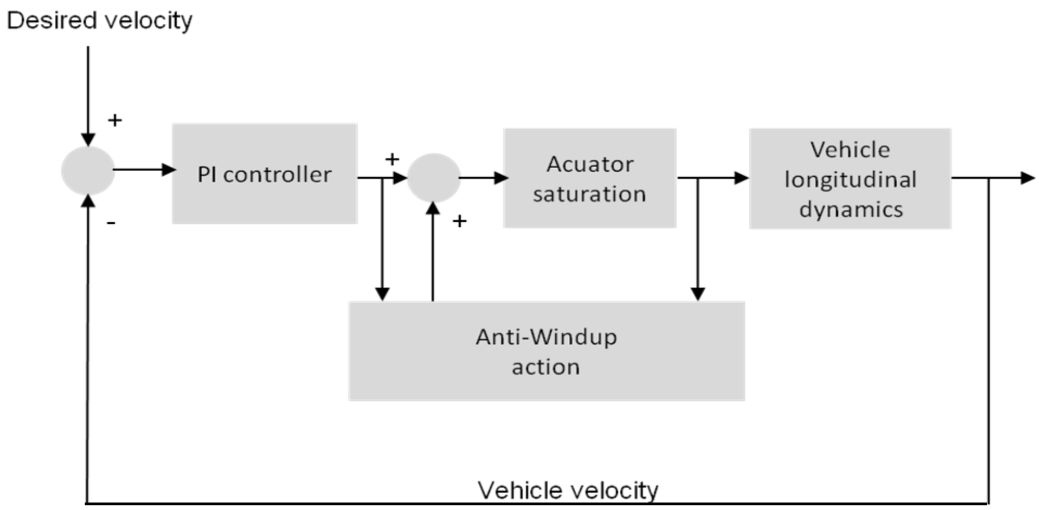


Figure 107: PI controller with anti-windup longitudinal controller architecture

Lateral control

For many stable systems with a smooth non-linearity, a simple proportional feedback control law may be sufficient to achieve the control objectives like proposed by Papadimitriou[40] and demonstrated by Mammar[41]. A control law that describes a simple proportionally feedback control is given by the following equations were u_1 and u_2 are respectively the longitudinal and lateral control laws:

$$u_1 = v = v_{ref} - (1 - k_1 \cdot \Delta x)$$

$$u_2 = \dot{\varphi} = \varphi - (k_2 \cdot \Delta y + k3 \cdot \sin \Delta\theta)$$

[40] Papadimitriou, I.; Tomizuka, M.: "Fast lane changing computations using polynomials", American Control Conference, 4-6 June 2003

[41] Mammar, S.: Assisted and automated lateral control of vehicles: Robust control approach; Habilitation thesis of Evry-Val-d'Essonnes University, 2001

The value v_{ref} is the reference velocity given by the open loop trajectory calculated by the trajectory planning module. The parameters k_1, k_2 and k_3 are tuning coefficients of the control law. These parameters were tuned empirically, and during these experiments a bound was defined on the probable tracking error. During the execution this tracking error is monitored.

This simple proportional control method provides asymptotic convergence to the desired state. However, when the state evolves in time, the error does not converge to zero. Since the vehicle moves according to a predefined trajectory that is feasible by design, such constant tracking error is avoidable.

Common saturation algorithm

In order to protect the vehicle actuators from damage and also to protect the vehicle from unexpected abrupt control actions, a saturation module was added to control the quality of the computed actions. In fact, this module constrains both, the magnitude of the computed controller action and also its rate of change. The effectiveness of this module can be seen in the experimental phase where the value of the authorized control action amplitude and rate of change can be adjusted progressively to their maximum values.

5.1.5 Summary

The trajectory calculation and command generation and validation systems described in this section were completely developed, integrated and tested in the Joint System simulation tool and in the Joint System Demonstrator (see HAVEit deliverable D41.1). More details and the description of the other co-system algorithms developed in HAVEit are available in deliverables D31.1, D31.2 and D33.4 and D33.5.

5.2 Challenge 3.2: Driver State Assessment

The idea of the HAVEit system is that automation is adapted to the intentions and limits of both of the two members in a Joint system - the driver and the co-system. Based on this information the current appropriate automation level is selected. If either the co-system or the driver is not able to manage the situation, then the automation level has to be changed. This could mean either a transition back towards a higher responsibility for the driver or a transition towards a higher responsibility for the co-system. A main precondition for applying this dynamic task repartition is to constantly know the potentials and limits of both members of the Joint System.

From the human's perspective in this concept, a driver model is necessary in order to assess if the driver is able to safely manage the driving task under all various conditions and automation levels. This is required as the automation will not be 100% reliable and will not cover the whole driving task. Instead, the system will produce errors and have some limits. So the driver is required to "stay in the loop" and has to react appropriately on system limits and system errors. Limits of the driver's performance capabilities are mainly set by driver's current physiological and psychological state. To identify negatively influencing factors on the driver state, a Driver State Assessment is required which is able to detect long term evolving driver drowsiness / fatigue which impairs the general arousal level of the driver and short term driver distraction which impairs task-oriented attention.

The output of this Driver State Assessment (DSA) module can be used to identify driver's need for automation and to make decisions when automation has to be up- or downgraded. For the assessment both driver related and driving related measures are used to derive a model of driver's behaviour (see Figure 108). Driver related measures refer to direct measures of driver state using a camera based system (Driver Monitoring System DMS) provided by Continental

Automotive France (CAF). It observes driver's eye movements, blinking patterns and gaze direction. A detailed description of the DMS software is provided in deliverable D32.1 "report on driver assessment methodology". Driving related measures (developed by WIVW) are indirect activity and performance measures which can be used to draw conclusions about the driver's state, e.g. reduced steering activity or decreased lane keeping performance. Both inputs are combined to derive a driver model that can be used for detecting driver drowsiness and driver distraction.

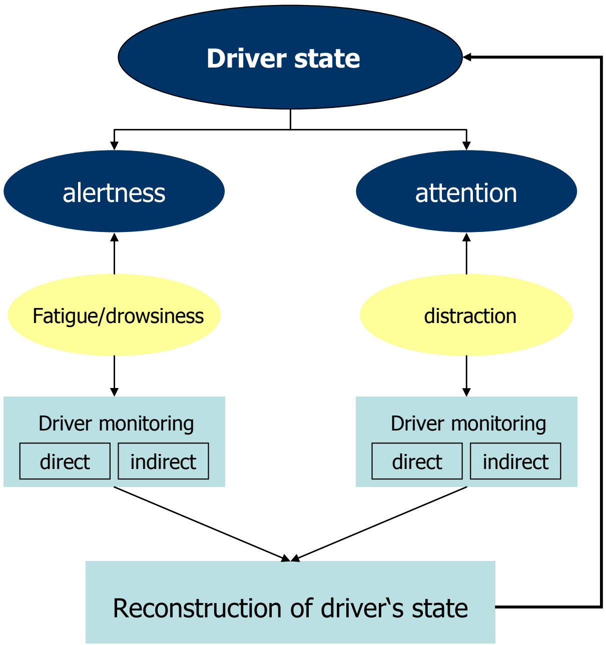


Figure 108: General concept for driver state assessment

5.2.1 Special issues of driver state assessment in highly automated driving

When assessing driver's state online in assisted and automation driving some aspects have to be kept in mind that will be relevant for the development of driver state algorithms.

Expected changes of driver state and attention allocation with assistance and automation

When thinking about driver state assessment in highly automated systems we get into the dilemma, that parallel with the increasing difficulty of driver state assessment on higher automation levels, the importance of this assessment also rises. It has to be considered that driving with assistance or automation itself may change the attention allocation to the driving task, driver's performance and driver's state.

For example, a badly designed automation may itself further promote the development of drowsiness due to the changed role of the operator from an active interactor to that of a passive observer. Vigilance decrements due to the low demands may be the result. Buld[42] et al. studied the effects of automation (ACC) in the driving simulation. Their results reveal a decrease of attention with increasing system availability (defined as increasing level of automation) and

[42] Buld, S., Krüger, H.-P., Hoffmann, S., Kaussner, A., Tietze, H. & Totzke, I. (2002). Wirkungen von Assistenz und Automation auf Fahrerzustand und Fahrsicherheit, *Veröffentlichter Abschlussbericht Projekt EMPHASIS: Effort-Management und Performance-Handling in sicherheitsrelevanten Situationen (Förderkennzeichen: 19 S 9812 7)*. Würzburg: Interdisziplinäres Zentrum für Verkehrswissenschaften an der Universität Würzburg (IZVW)

increasing time on task. Driver's state within a short time of automation was comparable to that of long-term manual driving. The effects of vigilance decrements were observable in physiological (EEG and eye lid closure measurements), psychological parameters (subjective ratings of attention and boredom) as well as performance measures (reduced accuracy, increased reaction times).

Another question is how supporting one subtask of driving affects other subtasks which still have to be performed by the driver him/herself. It could be expected that releasing the driver from one task will free resources for other subtasks, leading to performance improvements. When driving with an ACC, for example, longitudinal control is taken over by the system whereas the task of lane keeping remains with the driver. This should positively influence lane keeping performance. However, taking into account the present studies, this doesn't always prove to be true: either SDLP was not influenced or even increased when driving with ACC (e.g. Nilsson & Nabo, 1994; Peters, 1996; Rothengatter & Heino, 1994, Hoedemaker et al., 1998[43]). This result indicates that the strict distinction between longitudinal and lateral control is not valid from a psychological view. However, the stabilisation task has to be understood as one entity being performed with a constant accuracy. Decreased attention to one part might unwantedly lead to impairments in the other part.

Furthermore, driving with a badly designed automation may decrease driver's situation awareness (Endsley, 1988[44]) as the driver is removed from the loop and is no longer actively involved in the task. The result will be that the operator might get problems with detecting a problem requiring some interventions or will need at least additional time to determine the state of the system and to sufficiently understand what is happening (e.g. Hogema et al., 1997; Nilsson & Nabo, 1994; Nilsson, 1995 cited by [130], Stanton, 1997[45]).

An often reported positive effect of automation (e.g. ACC) is the subjectively perceived decrease in workload (according to a literature review by Buld et al., 2002). This may mislead the driver to interact more with other in-vehicle activities in order to increase one's activation level (Brockmann et al., 1995; Fancher & Ervin, 1998; Risser & Lehner, 1997[38]). The attention allocation towards distracting tasks may again lead to severe problems when system limits are reached on which the driver has react to. Reaction times to specific events are clearly increased under visual as well as cognitive distraction.

Available signals when driving with higher levels of automation

The major problem of assessing the driver's state in the HAVEit system is that with increasing levels of automation available parameters which can be used for driver state prediction will diminish. This will especially affect indirect driving related parameters (see Table 5). As the driver is released from performing parts of the driving task he is no longer required to make primary task commands, such as steering and activating the pedals. The resulting vehicle state is now an effect of system interventions (e.g. lateral and longitudinal control parameters as SDLP and speed or distances to lead vehicles) and is no longer related to driver's state.

Compared to manual driving using a Lane Departure Warning System (LDW) drops out information about lane exceedings as the system now actively prevents the driver from leaving the lane (a solution would be here to use the number of system's interventions instead). Measures as SDLP, steering activity, longitudinal control measures as well as reactions to specific events remain available. In case of using an Adaptive Cruise Control (ACC) measures

[43] Deram, P. (2004). *Vehicle-based detection of inattentive driving for integration in an Adaptive Departure Warning System- Distraction Detection*. Master Thesis. Royal Institute of Technology, Department for Signals, Sensors and Systems, Signal Processing, Stockholm

[44] Endsley, M.R. (1988). Situation Awareness global assessment technique (SAGAT). In *Proceedings of the National Aerospace and Electronics Conference* (NAECON) (pp. 789-795). New York: IEEE

[45] Stanton, N.A., Young, M. & McCoulder, B. (1997). Drive-by-wire: the case of driver workload and reclaiming control with adaptive cruise control. *Safety Science, 22*(3), 149-159

of longitudinal control and pedal activity drop out as the system now takes over the control on speed and distance behaviour and the driver no longer has to accelerate and decelerate. If a combination of ACC and LDW is used SDLP, steering activity and reactions to specific events will remain as useful parameters. In highly automated driving where also lateral control is fully controlled by the system and no more steering activity is required the only remaining information from driving are reaction times to specific events.

Measurable parameters	Driver only	Assisted (LDW)	Semi-Automated (ACC)	Semiautomated (ACC + LDW)	Highly automated
Indirect driving related measures					
Longitudinal control	x	x			
Pedal activity	x	x			
Lateral control	x	SDLP	x	SDLP	
Steering activity	x	x	x	x	
Reactions to specific events	x	x	x	x	x
Indirect measures referring to additional in-vehicle activities					
Driver's use of on-board systems	x	x	x	x	x
Direct driver related measures					
Vision-based measures	x	x	x	x	x

Table 5: Available parameters for online driver drowsiness and driver distraction detection dependent on the current automation level

To sum up, the only techniques that seem to safely deliver data independent from the current automation level will be reaction time to specific events, indirect measures referring to additional in-vehicle activities (e.g. driver's use of onboard systems) and direct driver monitoring meaning the observation of driver's eye closure and head or gaze direction via camera.

The conclusion for driver assessment is that these will be the most reliable and important information sources. The problem of the assessment of reaction times is that they require a trigger event. Therefore they are only useful for a continuous measurement of driver's performance abilities if the event is explicitly introduced by a secondary task.

The question whether and how many additional information can be included from driving related measures depends on the current automation level. For the development of suitable algorithms especially for driver drowsiness this information has to be taken into account: A kind of step-up/step down routine has to be developed for selecting the best algorithm based on the current incoming signals.

Need for perception model and manoeuvre detection

In contrast to direct driver related parameters such as driver's eye movements, indirect parameters seem to be highly influenced by the driving situation. Especially, parameters of lane keeping behaviour and steering activity proved to be strongly dependent on road characteristics and driven speed[46]. Green (2004[47]) revealed that SDLP is slightly decreased with higher speeds and that the measure differs on various road types (e.g. expressway, rural, urban roads) and in various traffic densities. In addition the mean SDLP slightly varies dependent on the context of the specific study in which the parameter was assessed. The values varied from mean=.18 m for real road (sd=.04m), mean=.23 m for simulator studies (sd=.13m) and mean=.21m (sd=.05m) for test tracks. However, these differences were not significant. The author states that the slightly elevated mean in the simulator may be due to the lack of pressure to attend to the primary driving task, mainly because the consequences of not attending to driving are less severe. Another explanation is the lack of feedback in the simulator (e.g. due to an uneven surface, unsteady crosswinds).

Steering activity is highly dependent on the environment (e.g. road curvature; Donges, 1975[48] and the performed driving manoeuvre, e.g. lane changing or turning at an intersection. For the manoeuvre detection, information about the use of the indicators can be taken into account. If the system detects the driver using the indicator, one can expect that he is currently performing a lane change or a turning manoeuvre or is planning to do this in the next few seconds. At this point, it has to be kept in mind that the indicators are not very confident as many drivers use them only rarely or even never. Another possibility is the use of high-pass filters. They allow filtering out steering signals coming from road characteristics or driving manoeuvres compared to micro steering corrections that can be directly linked to driver's alertness or attention level.

Variations in following distances can only be interpreted in a meaningful way if position and speed behaviour of the leading vehicles is available. Interpretation of variance in speed requires knowledge about the frequency of changes in externally valid speed limits. In order to detect some special events the driver has react to, both static information, e.g. about preceding curves, intersections (derived from map data) or relevant dynamic situations e.g. hard braking leading vehicle, has to be collected.

In order to reliably infer driver's state from indirect driving parameters it is essential that a perceptional model of the environment including information about road curvature, road geometry, driven speed, behaviour of surrounding vehicles, lane width and relative position on the road, is build up. This requires first, a certain amount of sensors measuring vehicle state and the environment, second, an adequate fusion of data delivered by these sensors and third, the calculation of meaningful parameters out of this data (e.g. parameter number of lane crossings derived from vehicle's lateral position on the road and lane width).

Algorithms using the mere data without any knowledge of the surrounding environment and without driving manoeuvre detection may produce a high false alarm rate. Furthermore, some additional information can be used to increase the confidence level of driver state classifications. For driver drowsiness detection trip information about time of day (driving around lunch time or during the night increases the probability of getting drowsy) and the duration of the drive (time on task increases probability of getting drowsy) is suitable.

Definition of criteria of impairment

A very important factor for driver state assessment is the definition of criteria when driver's state is said to be impaired. In general, two different approaches can be applied: absolute vs. relative

[46] Knappe, G., Keinath, A. & Meinecke, Ch. (2006). Empfehlungen für die Bestimmung der Spurhaltegüte im Kontext der Fahrsimulation. *MMI-Interaktiv, 11*. Available at www.useworld.net/ausgaben/12-2006/02-Knappe_et_al.pdf [06.06.2007]

[47] Green, P., Cullinane, B., Zylstra, B. & Smith, D. (2004). *Typical values for driving performance with emphasis on the standard deviation of lane position: a summary of the literature*. Report SAFE-IT Task 3a (Safety Vehicles using adaptive interface technology).

[48] , E. (1975). Experimentelle Untersuchung des menschlichen Lenkverhaltens bei simulierter Straßenfahrt (Teil 1). *Automobiltechnische Zeitschrift, 77* (5), 141-146

criteria (Brookhuis et al., 2003[49]). For defining absolute criteria, fixed values have to be set defining the absolute red line of demarcation for impaired driver behaviour which should be valid for the global driving population under all circumstances. The problem of this approach is the high variation among drivers resulting in either a high number of false alarms (threshold set too low) or a high number of missings (threshold set too high) for some drivers.

In contrast to absolute criteria, relative criteria relate currently observed behaviour to an individual baseline. This approach is evaluated as more favourable due to a comparable low intraindividual variance in driving behaviour within one driver. So, deviations from a baseline should be clearly interpretable as impairments. Brookhuis et al. (2003) suggest a number of values for the definition of absolute and relative criteria for impaired driving behaviour (see Table 6). The absolute values are conformable with definitions of driving errors or abnormal driving behaviour used in driving behaviour observations (see Reichart, 2001[50]).

	Absolute change	Relative change
Following too close:		
Time headway to lead vehicle (TTC)	<0.7 s	−0.3 s
Straddle lanes:		
Steering SD	>1.5°	+0.5°
Lateral deviation (SD) of the vehicle	>0.25 m	+0.04 m
Minimum time-to-line crossing (TLC) right lane	<1.3 s	−0.3 s
Minimum time-to-line crossing (TLC) left lane	<1.7 s	−0.2 s
Median TLC (right lane)	<3.1 s	−0.7 s
Median TLC (left lane)	<4.0 s	−1.4 s
Driving too fast:		
Vehicle speed	limit+10%	+/− 20%

Table 6: Definition criteria for following too closely, bad lane keeping behaviour and driving too fast (Brookhuis et al., 2003)

Conclusions for the Driver State Assessment in HAVEit

The goal of Driver State Assessment in HAVEit is to measure if the driver is able to safely manage the driving task under all conditions and levels of automation. This includes cases where he or she is required to take over the driving task from the automation because the system fails or reaches its limits. Driver state is assumed to be an indirect measure of his or her error management ability - if driver state is impaired, the driver will no longer be able to adequately react to these events. An optimum arousal level and a task-oriented attention are identified as the two components of driver state. Therefore, the main goal of driver state assessment must be, on the one hand, to measure the degree of drowsiness which highly impairs alertness level and is expected to be a more long-term process, and on the other hand, the degree of short-term distraction which highly impairs driver's task oriented attention.

The most promising measures for the driver state assessment are vision-based measures which have the big advantage that they directly assess the driver's state by analyzing driver's eye closure behaviour for drowsiness detection and driver's head or gaze direction to detect distraction (=direct driver monitoring). The other group of parameters try to use driving behaviour measures to indirectly infer on the driver's state (indirect driver monitoring). These two data sources have to be combined within both one driver drowsiness and one driver distraction algorithm which are able to detect or even predict driver's state with sufficient

[49] Brookhuis, K.A., de Waard, D. & Fairclough S.H. (2003). Criteria for driver impairment. *Ergonomics, 46 (5)*, 433-445

[50] Reichart, G. (2001). Menschliche Zuverlässigkeit beim Führen von Kraftfahrzeugen. *VDI Fortschritt-Berichte, Reihe 22,* Nr 7. Düsseldorf: VDI-Verlag

confidence. Vision-based measures have to be proven as very effective in detecting driver's impaired state due to several aspects: One sensor, a camera, provides information both for driver drowsiness and driver distraction detection using various parameters which can be extracted from driver's eye and head movements. The technique allows an unobtrusive measure without any physical contact with the driver. The measurement is able to distinguish between different levels of driver impairment (especially for drowsiness detection: distinction between alert, slightly drowsy, drowsy and sleepy; for distraction: distracted vs. non-distracted). The measure is mainly independent from road characteristics and the gathered data are independent from the current automation level.

Indirect measures from driving behaviour are much more difficult to interpret and seem partly not very suitable to infer from them on driver's state. Analyzing the literature about the effects of drowsiness and distraction on driving behaviour reveals as most promising measures parameters of lateral control, especially standard deviation of lateral position (SDLP) and number of lane crossings. These parameters increase both under low vigilance and distraction conditions. The other variables can be derived from the steering activity of the driver: With increasing drowsiness and distraction the number of micro-corrections seems to be decreased, while large and faster corrections after phases of nearly no steering interventions increased. Reactions to specific events are highly sensitive to impaired driver state but much more difficult to assess compared to the observation of continuous parameters. Variation in speed and following distance seem to be moderately suitable to clearly refer to an impaired driver state.

However, the collection and the correct interpretation of indirect measures require knowledge about the surrounding environment. For the calculation of SDLP, lane markings have to be assessed by sensors. As SDLP proved to be, for example, influenced by lane width and road curvature, also information about road characteristics must be available. Also steering activities can only be interpreted correctly if influencing factors from the environment (e.g. driving through curves) or driving manoeuvres (e.g. turning, lane changing) are considered. As the HAVEit applications will mainly be used on freeways, this aspect seems to be less relevant, but still relevant. Assessing following distances and the correct interpretation of variations in speed requires the knowledge about surrounding vehicles and valid speed limits. Also information about the trip (time on task and time of day) represents helpful indicators for the probability of impaired driver's state.

Concerning the detection of driver distraction, results from the literature review revealed that especially visual distraction will be detectable. Here, analyses of driver's gaze and head position provide the most reliable information. In addition, knowledge about driver's use of onboard entertainment systems will be taken into account. However, the increasing number of nomadic devices in the car complicates the gathering of these data. Much more difficult is the assessment of cognitive distraction. Some approaches use a combination of visual data and lane keeping performance data: a decreased visual field (interpreted as attentional tunnelling) together with an increased SDLP is said to be an indicator for cognitive distraction.

The most urgent problem of driver state detection within highly automated driving is the fact that with increasing automation level the number of available parameters for indirect driver assessment will diminish. On the highest automation level, the driver is no longer required to make any inputs in the vehicle. The resulting vehicle behaviour is no longer influenced by him / herself but fully controlled by the system.

5.2.2 Driver State Assessment in HAVEit

The algorithms for indirect driver monitoring were developed in Python. The final software was written in C language and implemented on the hardware platform CSC. The driver state assessment includes direct driver monitoring (developed by partner CAF), indirect driver state assessment developed by WIVW. Both subsystems deliver information on the driver state and confidence levels that are fused to generate the final drive state and confidence information.

Direct driver monitoring

The Driver Monitoring System (DMS) provided by CAF is a vision based system analyzing the face of the driver in order to provide information about his/her state degradation. The system thereby provides support not only in the event of drowsiness, but also during actions which temporarily divert attention from the driving task, such as inserting a CD or operating the navigation equipment. Thanks to a unique compact camera including a CMOS sensor and pulsed NIR light (invisible light) DMS monitors driver's drowsiness (DDM) and inattentiveness (meaning “distraction” in HAVEit, DIM). A distant processing unit analyzes the image flow provided by the camera to extract information about the driver's eyelid and blinking patterns and face characteristics. The system electronic functions use these parameters to reliably determine driver's drowsiness and inattentiveness. The system works by night and day taking into account that drowsiness is not just a night-time phenomenon and is also likely to occur during the morning or the afternoon. The system is fully automatic; it doesn't require any input or specific behaviour from the driver. DMS provides multi-level information which allows implementing various warning strategies combining text message, seat or seat belt vibrations and acoustic warnings.

Indirect driver monitoring

Model for driver drowsiness assessment

Driver drowsiness monitoring has to be carried out on multiple levels with reference to the underlying energetic processes that occur in different time frames and correlate with different performance levels of the driver. It is assumed that a differentiated set of indicators is required to reflect this multi-level concept. The following levels of this drowsiness development and the related consumption of energetic resources have to be distinguished in this model:

- In the “awake” state full resources are available. The driver’s behaviour is not influenced in any way.
- On the next “slightly drowsy” level some resources have to be invested to maintain a certain arousal level. This should be measurable by first behavioural changes, e.g. in the blinking pattern. DMS (driver monitoring system) will provide an output “slightly drowsy” at this stage. Driving performance will still remain uninfluenced at this level. Therefore, no indicators will be derivable from driving behaviour on this state.
- On the next “drowsy” level the driver has to invest a high effort to stay awake and to maintain an adequate driving performance. This effort can be measured again by the observation of the eye blinking behaviour (a higher frequency of medium and long blinks). DMS will provide an output “drowsy” at this stage. On this level also first hints from the observation of driving behaviour (over longer time intervals) will be available. Results from the re-analysis of driving simulator data (detailed description see D32.1) revealed that suitable indirect parameters for detecting the “drowsy state” will be the standard deviation of lateral position (SDLP) and several parameters derived from steering activity averaged over a longer time interval (e.g. mean amplitudes of steering wheel reversals)[51].

[51] Other parameters for the driver activity, such as brake and accelerator pedal usage proved to be less sensitive for detecting a drowsy driver. In addition, they are much more influenced by the current traffic environment and are the

- On a higher level, energetic resources are exhausted and performance capabilities are exceeded. This "sleepy" state will be observable in micro-sleep events and the accumulation of single relevant attention lapses in driving. DMS will provide an output "sleepy" at this stage. Results from the re-analysis of driving simulator data revealed that e.g. the number of lane crossings rises heavily especially at the sleepy state. The driver seems to be no longer able to maintain a safe lateral control. Also, very fast steering corrections occur only on higher drowsiness states and therefore seem to be a reliable indicator for a really sleepy driver.
- On the final level, a complete collapse of the energetic system occurs accompanied by a full breakdown of performance. The driver has fallen asleep and does not longer respond to relevant driving situations (e.g. lane departure or imminent collisions with a lead vehicle) and e.g. take-over requests by the system (stage "unresponsive").

Model for indirect driver distraction assessment

The observation of distraction can only be made within a short-term range of several seconds where it has to be monitored, whether the driver is not looking at the road and/or is operating some other tasks inside the vehicle, e.g. navigating within a complex information system or using the cell phone. It can be expected that the distraction level increases when performing more demanding secondary tasks for a longer time interval.

The idea of the indirect assessment of driver distraction detection is the following: If we know that the driver is performing some additional tasks during driving, e.g. if he/she is using in-vehicle devices we can expect that he/she doesn't pay enough attention on the driving task in that moment- that is the distraction potential is very high and impairments in driving performance will occur. This method will complement the assessment of visual distraction by camera observation. Even if the driver is not looking at the device we can assume that he/she will be cognitively and physically distracted by the engagement in the task.

In order to evaluate the distraction potential, all available information about driver's activities on in-vehicle devices and other vehicle controls have to be collected (beside the information about driver's visual distraction by the camera). Therefore, a number of sensors must be provided that gather such data about additional tasks. The next step is to classify them into meaningful groups with respect to their distraction potential (classes of additional tasks). The main dimensions for this classification are the visual, cognitive and motoric demands of the tasks. But also the expected duration, the time urgency and the initiator of the tasks have to be considered. There are a couple of tasks that require only short, single button pressed on the steering wheel. Those tasks should be less distracting with reference to the visual, cognitive and motoric demands. In contrast, others demand navigating through a menu system and repeated inputs by a more complex controller over a longer period of time. Those are expected to require more attentional resources. The first challenging task will be to define adequate weights according to the distraction potential for the several task groups. The output will be a distraction weight for each class of tasks.

The second one is to observe the time sequence of one or more activities. For longer tasks, the probability that the driver is not attending to the driving task even if he shortly looks back to the road within the single steps of the tasks heavily increases.

Not only the number of single steps but also the time delay between them has to be considered. As it can be assumed that distraction increases with the time the driver is occupied with an additional task without adequately interrupting it, several single steps within a certain time period have to be accumulated (stepwise increase of distraction score). From the literature

first ones who will drop out when the driver is driving on higher automation levels, e.g. driving with ACC. Therefore, it was decided to rely more upon parameters that describe the steering activity and the driver's lateral control performance.

about gaze behaviour it is known that glances longer than 2 seconds (e.g. AAM guidelines, 2003) away from the road are very dangerous for driving safety. This reference can also be applied to the use of in-vehicle devices. If the time delay between two single operations is less than 2 sec, then the distraction weights are summed up. The result is a continuous distraction score over time. Based on this distraction score and some defined thresholds, adequate interventions can be applied to bring the driver back into the loop and to prevent him from severe safety risks.

The relevant computations that are performed in the DSA software are:

- Classification of distracting tasks: eyes on/off the road (output of the DIM[52] module by DMS system), hands-off driving, engaging in secondary in-vehicle tasks
- Definition of distraction weights per group according to their distraction potential
- Definition of a continuous distraction score by observing time sequence and task switching strategies between distracting tasks and driving task
- Definition of a discrete distraction diagnostic (distracted vs. not distracted) by setting thresholds for unacceptably long distraction, adaptable to current automation level and task demands (e.g. driven speed)

<u>Main principles of driver state assessment</u>

The main principles of the DSA software architecture can be summarized as follows:

- Long-term drowsiness vs. short-term distraction are assessed separately
- Direct (output of the DMS) and indirect measures (internally calculated parameters from driving performance and driver's activity) are fused within a model of driver behaviour
- Driver's state is assessed on multiple levels with a differentiated parameter set for each level
- In a calibration phase the "normal" driving behaviour of each individual driver is assessed in order to adapt the algorithms (especially for the calculation of some indirect parameters)
- A manoeuvre detection and classification (e.g. lane changes, sharp curves) is included in order to define the appropriateness of calculated parameters (especially indirect ones from driving behaviour) and to decide on their inclusion in the output
- The current automation level (hands-off vs. hands-on) is considered for the decision on the inclusion of parameters in the output
- Signal quality (e.g. detection of lane markings) is considered for the definition of the confidence of the final output

The software delivers the following outputs

- 2 distraction states: distracted vs. not distracted (+ confidence level)
- 6 drowsiness states: undefined, awake, slightly drowsy, drowsy, sleepy and unresponsive (+ confidence level)

[52] Driver inattentiveness monitoring

The following information is used for the classification of the several outputs:

- For state "distracted":
  - Information about the driver looking on/off the road (output of the DIM module by Direct Driver Monitoring System DMS)
  - Information about driver's use of onboard systems, buttons pressed on the steering wheel, using the cell phone etc.
- For the multiple states of "drowsiness":
  - Direct: output of DMS diagnostic
  - Indirect: time of day, time on task, relevant lane keeping behaviour, relevant steering wheel activity, relevant distances to a lead vehicle, relevant duration of unintended hands-off driving, inadequate reaction times to take-over requests etc.
- Direct and indirect measures are currently fused by simple disjunctions on lower drowsiness levels and for the distraction diagnostic (if either the direct or the indirect monitoring detects a drowsy, sleepy or distracted driver the respective output is given by the DSA). On the unresponsive level currently a conjunction of both measures is used (if both direct and indirect monitoring detects an unresponsive driver the respective output is given by the DSA).
- The confidence level of the outputs will be mainly derived from signal quality within the observed time buffer. This will be dependent both from vehicle performance measures (e.g. lane detection quality) and camera performance measures (e.g. face tracking performance-output by DMS).

5.2.3 Validation of Driver State Assessment

<u>Evaluation of direct drowsiness detection in real driving conditions</u>

The evaluation of the DDM module has been done in real conditions. The processing algorithms have been developed in a C++ environment. The camera and NIR units have been integrated into an experimental vehicle, in the car instrument panel, thus observing the driver's face through the steering wheel. During the experimentations all the information described in the previous sections are recorded in real time. A wide variability of drivers has been tested (racial phenotypes, age, hair/skin colour, women, men, etc.).

Specific experiments have been set up to evaluate the performance of the drowsiness diagnostic. During the experiments a technical supervisor and a medical team (a doctor) accompanied the drivers. Drivers are instrumented with physiological sensors to record their Electroencephalogram (EEG) and Electro-Oculogram (EOG). Drivers are asked to self rate the evolution of their state. In addition the technical supervisor annotations are recorded. At last videos of the driver head and upper part of the body as well as, front and back videos of the road are recorded. The images provided by these camera, plus the image of the driver face from the sensing device with diagnostic (from Driver Monitoring PC) are mixed into a QUAD and then recorded by a VCR. In addition to the DMS parameters some vehicle parameters are recorded: Vehicle speed, steering wheel angle, in-vehicle temperature and yaw.

Experimentations have been performed on motorway, in real driving conditions, 11 drivers following a strict experimental protocol. Each driver drove about 360 km for each experiment. The driver is asked to stay on the right lane and not to exceed 100km/h in order to limit interaction with the traffic (no overtakes) but also to enhance the occurrence of drowsy situations. The objectives of these experiments were to compare the performances of the diagnostic with the expert references from the electrophysiological expertise provided by the medical team or by the driver's auto evaluation.

Each of the experiment duration was about 3 hours. For each experiment, a table of correlation between diagnostic and expertise is built up (see Table 7). The t_{ii} variables are calculated by analyzing the experimental results. The t_{ij} are representing the duration where expertise is rated as "state i" and diagnostic is rated as "state j". For example t_{11} represents the data points where the expertise is rated as sleepy and diagnostic as sleepy. Then from this set of variables some statistics can be calculated. The HIT corresponds to the case when the expertise detects a degradation of the driver state and this degradation is well detected by the diagnostic (True Positive). The PASS corresponds to the case when the expertise does not detect any degradation and that is confirmed by the diagnostic (True Negative). The MISS corresponds to the case when the diagnostic underestimates the expertise (False Negative). And the FA corresponds to the case when diagnostic overestimates the expertise (False Positive).

	Expertise				
Diagnostic		Sleepy	Drowsy	Slight. Drowsy	Alert
	Sleepy	Hit t_{11}	FA t_{12}	FA t_{13}	FA t_{14}
	Drowsy	Miss t_{21}	Hit t_{22}	FA t_{23}	FA t_{24}
	Slightly Drowsy	Miss t_{31}	Miss t_{32}	Hit t_{33}	FA t_{34}
	Alert	Miss t_{41}	Miss t_{42}	Hit t_{43}	Pass t_{44}

Table 7: Table of correlation for Drowsiness diagnostic

Then two statistics can be derived from these data:

- The sensitivity refers to the proportion of people with state degradation who have a positive test result.

$$Sensitivity_{driverN} = \frac{HIT_{driverN}}{HIT_{driverN} + MISS_{driverN}}$$

- The specificity refers to the proportion of people without state degradation who have a negative test result.

$$Specificity_{driverN} = \frac{PASS_{driverN}}{PASS_{driverN} + FA_{driverN}}$$

The final results are reported in Table 8. This table shows that the sensitivity is quite good for all the subjects (greater then 80% for most of the subjects) with an average value of 86%. Otherwise the specificity is very high for most of the drivers except for driver D04. The false alarm rate is 4% compared to a biomedical expertise based on EEG/EOG.

Drivers	01	02	04	05	06
Sensitivity	0.58	0.85	0.92	1.00	1.00
Specificity	1.00	0.75	0.49	1.00	1.00
Drivers	07	08	09	10	11
Sensitivity	0.81	0.95	0.75	0.89	0.87
Specificity	0.96	0.99	0.70	0.97	1.00

Table 8: Diagnostic performances for the set of experimented drivers

Evaluation of direct distraction detection in laboratory and real driving conditions

Evaluation of the DIM Pose Accuracy

The evaluation of the DIM has been performed in laboratory and real driving situations. The DIM Pose Accuracy has been evaluated in laboratory by comparison with a Continental head pose ground-truth system. Subjects are equipped with a helmet associated with a dedicated image processing software that extract the exact head pose. The data base includes 12 image sequences of 12 subjects collected in static conditions.

Initialization	Detection is performed at frame rate	
Refresh rate:	>10 Hz	
Position Range:	>95% eye ellipse	
On-road Rotation Range	Average Yaw range:	29°
	Average Pitch range:	26°
Off road detection rate	Out of the 100% on-road area	92,7% σ =2.95
	out of the 2.25x100% on-road area	96.44% σ=2.38

Table 9: Evaluation of the DIM Pose accuracy

Evaluation of the DIM diagnosis

The evaluation of the DIM diagnostic has been performed in real driving situation. The system has been tested with 17 subjects (16 men, 2 with transparent glasses, 1 woman). The vehicle was driven on motorway in real conditions; the total duration of the experiment was about 4 hours. A collection of about 80 real driving videos has been set. The output of the DIM has been compared with those of a video. Specific tools have been developed to facilitate the manual expertise of the videos. Ambiguous positions are not considered for the evaluation. A statistical analysis of the performance of the system is performed using specificity and sensitivity. The evaluation results are reported in Table 10 and Table 11.

	Manual expertise		
DIM		OFF	ON
	OFF	True Positive (HIT)	False Positive (False Alarm)
	ON	False Negative (MISS)	True Negative (PASS)

Table 10: Table of correlation for DIM diagnostic

Accuracy	2 levels (attentive, distracted)
Latency	800 ms for On-road to Off-road transition then 100 ms
	200 ms for Off-road to On-road transition then 100 ms
Specificity (average)	98.77
Sensitivity (average)	99.74

Table 11: Evaluation of the DIM Diagnostic

Validation of the first DSA modules in driving simulator studies

For a technical and empirical validation of the algorithms, the first version of the algorithms was implemented in the WIVW driving simulation. 2 separate studies were conducted: one for the validation of the drowsiness detection module, one for the validation of the distraction detection module. The results were used to further modify and improve the developed algorithms.

Validation of the first version drowsiness detection module

Experimental setup

The empirical tests were conducted with 12 test drivers in the WIVW driving simulator. They were all familiar with simulator driving. Mean age was 32 years (standard deviation 10 years), the oldest driver was 59 years old, the youngest 23, 7 were male, 5 female. They all had a normal non-corrected vision and did not require glasses or lenses for driving.

They were randomly assigned to 2 experimental groups: 6 drivers drove the test course in the Driver Assisted mode of the HAVEit system (DA group). This means that they were slightly assisted by a lane keeping assistance system which only intervened in critical situations when the driver tended to get off the road by providing a slight force on the steering wheel towards the opposite direction. The other 6 drivers drove in the Highly Automated mode of the HAVEit system (see Figure 109 left). On this level, lateral and longitudinal control is performed automatically by the system up to a speed of 130 km/h. The driver can drive hands-off. In case of a situation the system can not manage (e.g. if a lead vehicle is decelerating and the relative speed between ego and lead vehicle is too large) a take-over request is given. The driver then has to take-over the steering wheel and the pedals and a transition towards the Driver Assisted mode is executed. Lane changes have to be performed manually by the drivers.

The driving task consisted of driving on a 2 lane motorway with mainly straight and slightly curvy sections. The valid speed limit was set to 120 km/h over the whole track. The track contained free driving scenarios (without any lead vehicle), car follow scenarios with a lead vehicle driving around 110 km/h, traffic jams, forced lane changes at road works and heavy braking manoeuvres of the lead vehicle in random intervals (resulting in a take-over request when driving in Highly Automated mode). The drivers were instructed to always stay on the right lane even when the lead vehicle drives slower than allowed, stick to the valid speed limit and only execute a lane change when they are forced to at road works. Furthermore, driving in night time was simulated by a very realistic illumination of the surrounding environment and traffic (see Figure 109 right). The drivers started the drive after lunch at about 1.30 pm and after having performed another 2 hours lasting study with a distracting task. The duration of the drive was determined by the time it took the drivers really get drowsy or even sleepy. Due to the very realistic night-time environment, the time of day and the heavy load of the previous study for most of the drivers it took not longer than 2.5 hours. For the low minority of drivers who showed no tendency to really fall asleep the drive was stopped after 2.5 hours.

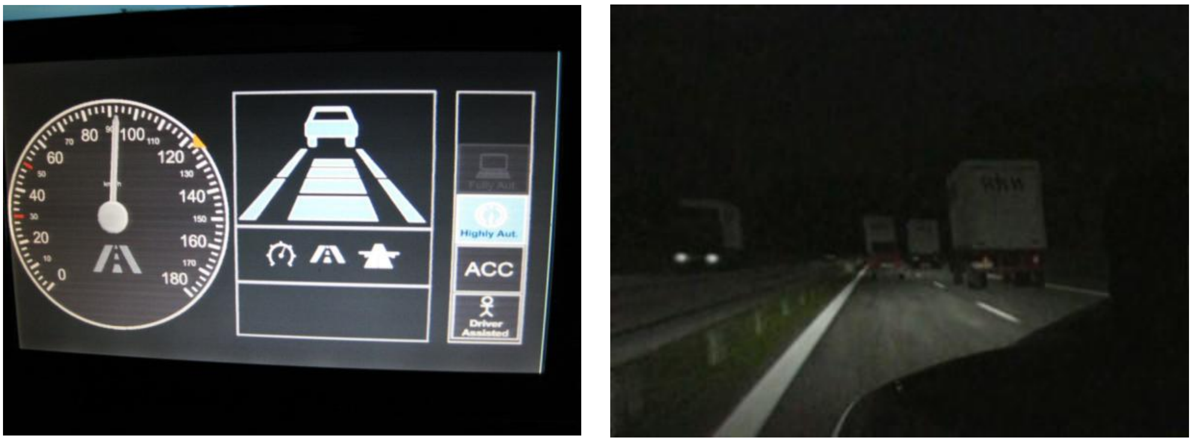


Figure 109: The HMI of the HAVEit system showing the state "Highly Automated mode" (left) and one of the night-time scenarios "traffic jam" (right) in the validation study.

The measurements for the validation of the DSA module were:

- The outputs of the direct driver monitoring (DMS system by CAF) - providing different drowsiness states: alert, slightly drowsy, drowsy, sleepy.
- The outputs of indirect driver monitoring (only for Driver Assisted group as for Highly Automated group no data are available): alert, drowsy, sleepy and unresponsive. Please note that the indirect monitoring provides no "slightly drowsy" output.
- Driver's subjective drowsiness rating every 20 minutes online by using the Karolinska Sleepiness Scale KSS. The Karolinska Sleepiness Scale is a 9 point rating scale from 1 = "extremely alert" up to 9 = "very sleepy with great effort to stay awake" (Akerstedt & Gilberg, 1990).
- Test leader's expert drowsiness rating every 20 minutes online by using the KSS.
- Observation of other drowsiness indicators by the test leader (e.g. yawning, scratching, head nodding, micro-sleep events)

Results

The first important question was if it is possible to induce drowsiness at all by driving in the driving simulator. As it gets obvious from rows 1 and 2 in Table 12 drowsiness induction was very successful in the study - nearly all drivers really got drowsy or even sleepy within a very short time of usually less than 2 hours. 10 of 12 drivers rated themselves with maximum values of 8 or 9 on the KSS (see row 1).

	Driver assisted						Highly automated					
	D1	D2	D3	D6	D9	D11	D4	D5	D7	D8	D10	D12
Max. drowsiness rating by driver	9	8	9	9	9	9	7	9	9	7	8	8
Max. drowsiness rating by expert	9	9	9	9	9	9	8	9	9	8	9	9
Drowsy events detected by direct monitoring?	yes	no	yes	no	yes	yes	yes	yes	yes	yes	yes	yes
Sleepy events detected by direct monitoring?	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Drowsy events detected by indirect monitoring?	yes	yes	yes	yes	yes	yes	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Sleepy events detected by indirect monitoring?	yes	yes	yes	yes	no	yes	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Drowsy/sleepy events detected by test leader?	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Table 12: Drowsiness detection quality provided by DSA module

Two drivers (marked in yellow) reached only a value of 7 according to their opinion which means that they perceived beginning drowsiness but without having problems to stay awake. In most cases this is in agreement with the expert judgment of the test leader (see row 2). Some drivers slightly underestimated their drowsiness level.

a) Detectability of drowsy and sleepy events by direct and indirect driver monitoring

The second question was whether the DSA algorithm could reliably detect the different stages of drowsiness. When looking at rows 3 to 6 in Table 12 it can be seen that drowsiness was very well detected by both measures in the DSA module. Rows 3 and 4 reveal that the camera reliably detected drowsy and sleepy events for all except 2 drivers: For driver D2 the camera had tracking problems over the whole drive. One explanation could be that the driver had very bright eye brows so that the detection of relevant facial features was impeded. Driver D6 had sleepy blinks without having drowsiness blinks before. This is unusual, however, but not impossible. Indirect driver monitoring was only available for the Driver Assisted condition (see row 5 and 6). Here it gets obvious that for all 6 drivers drowsy events could be reliably detected by the respective parameters. For 5 of the 6 drivers also sleepy events were detected. The last row shows that at least one of the measures should have indicated drowsy or sleepy events as this was observed by the test leader. So, if one of the measures did not fire this could be definitely defined as a missing. Please note that the table is useful for a first overall evaluation of the sensitivity of the measures.

Occurrence and plausibility of single outputs: DMS output

Table 13 shows the distribution of DMS drowsiness outputs across all drivers and driving time. As can be seen, most of the time the drivers were detected as “alert” (85.9%). In 9.6% of the time drivers were detected as “slightly drowsy” by the camera, in 2.8% of the time as “drowsy” and 1.7% as “sleepy”. The number of false alarms was quite low and mainly due to incorrect classification of display glances or due to the driver speaking (might get a greater problem in real driving). Some specific algorithms were already implemented in the meanwhile to cope with the display glances and to discard them. This is running generally quite well but it depends a lot on the correct implementation of the camera into the vehicle. Further improvements could be achieved by a connection between distraction and drowsiness detection algorithms.

Parameter DRIV_STATE_DROW_DMS	% of total time
0 („alert“)	85.9 %
1 (“slightly drowsy”)	9.6 % (9502 sec)
2 (“drowsy)	2.8 % (2805 sec)
3 (“sleepy”)	1.7 % (1737 sec)

Table 13: Distribution of DMS drowsiness states in validation study

The internally computed confidence level of the DMS output (see Table 14) is in 32.8% of total time below 0.3 (not reliable); in 8.6% of the time between 0.3 and 0.6 (acceptable) and in 58.6% of total time above 0.6 (good). Overall, the tracking quality can be defined as quite good.

Parameter DRIV_CL_SRC	% of total time
<0.3 (low)	32.8 %
0.3-0.6 (acceptable)	8.6 %
>0.6 (high)	58.6 %

Table 14: Distribution of DMS drowsiness confidence level in validation study

A low drowsiness confidence level can have different reasons: Either the face recognition is lost temporarily because the driver is changing his position on the seat or is occluding parts of the face with his hands etc. or permanently due to specific characteristics of the driver's face (e.g. very bright eye brows), or too less eye lid closure events occurred within the observed time interval.

The first reason might lead to a higher number of missings as especially if the driver starts getting drowsy, he tries to activate himself by moving on the seat or grasping in the face. The second reason is less problematic as this occurs very often when the driver is extremely alert. During the drives it could be observed that despite a temporarily low confidence level “drowsy” and “sleepy” events that are classified by the camera are still reliable.

b) Occurrence and plausibility of single outputs: Indirect monitoring (parameters from driving behaviour)

From Table 15 it can be seen how often the defined criteria for the classification of the various drowsiness states based on indirect monitoring were triggered in the validation study. A criterion is fulfilled if the respective threshold of the parameter is exceeded. The following aspects can be summarized:

Most often the “drowsy” state was triggered by the indirect parameters. Especially sensitive was the criterion CRT_AD_SWR_MEAN_RATE: this event occurred in all 6 drivers of the DA condition. The other criterion for the “drowsy” state, CRT_SDLP_RATE, occurred only with three drivers and always after the other threshold for AD_SWR_MEAN_RATE had been exceeded.

This result indicates that parameters from steering behaviour are more sensitive than parameters from the resulting lane keeping performance in order to detect the “drowsy” state.

The “sleepy” state (initiated by CRT_N_SW_V and CRT_N_LC) was classified less often.

Events that are classified on state 5 “unresponsive” (for example: hands-off driving in Driver Assisted mode or too late reaction after a standstill) were extremely seldom (except critical TTCs that occurred very often due to the included critical braking event). This result is in full accordance with the expectations, that such events happen only when the driver had really fallen asleep (this could be verified by the online observation during driving).

For some criteria (CRT_N_SW_V, CRT_AD_SWR_MEAN_RATE, CRT_N_LC and CRT_T_LC) a higher number of false alarms occurred due to an avoidance manoeuvre at the critical braking event. As they are not directly correlated with an increased drowsiness but instead with a critical interaction with another road user these events should be not included in the drowsiness classification. To identify these specific events the classification of a new situation “critical interaction with another road user” is required.

Some parameters exceeded the defined thresholds also in cases of extreme distraction. In the present study this was the case for one driver who had extreme problems with reactivating the Highly Automated mode and therefore had heavy steering and lane keeping problems. By a connection between distraction and drowsiness algorithms these events could be clearly attributed to distraction.

Criteria	% of total time	Number of events clearly attributable to drowsiness	notes
CRT_AD_SWR_MEAN_RATE	8.6	69	Occurred in all 6 drivers of DA-condition; false alarms due to avoidance manoeuvre in critical braking situation
CRT_SDLP_RATE	2.0	22	Occurred in 3 drivers of DA-condition, but always after CRT_AD_SWR_MEAN_RATE; 100% w.r.t. driver state
CRT_N_SW_V	5.5	6	False alarms occurred due to avoidance manoeuvre in critical braking situation or due to strong distraction
CRT_N_LC	1.4	7	Occurred in 3 drivers; FA due to avoidance manoeuvre in critical braking situation or due to strong distraction
CRT_T_LC	0,01	7	False alarms occurred due to avoidance manoeuvre in critical braking situation or due to strong distraction
CRT_TTC	1,4	34	Always in critical braking situation
CRT_HANDS_OFF	0,9	1	FA in traffic jam
CRT_HANDSOFF_AFTER_TOR	0,0	0	FA due to a loss of mode awareness

Table 15: Criteria for drowsiness-related abnormal driving behaviour used in indirect drowsiness monitoring

c) Analysis of timely correlations between direct and indirect monitoring

Another analysis was the observation of the timely correlation between the direct and indirect monitoring: When are the criteria triggered? Which criteria react earlier? Are there correlations between DSA outputs and the subjective rating of drowsiness?

Figure 110 shows an example of the time plot for Driver D11 after 120 minutes of driving. The driver had several "slightly drowsy" events detected by the camera before. At the beginning of the 6th 20-minute-loop the criterion CRT_AD_SWR_MEAN_RATE is shortly triggered the first time (classified as "drowsy"), followed by a "drowsy" event detected by the camera. 5 minutes later again CRT_AD_SWR_MEAN_RATE fires, followed by the first "sleepy" event detected by the camera two minutes later. While CRT_AD_SWR_MEAN_RATE stays activated until the end of the loop, several drowsy and again one sleepy event are detected by the DMS. This analysis was done for all drivers over the whole drive.

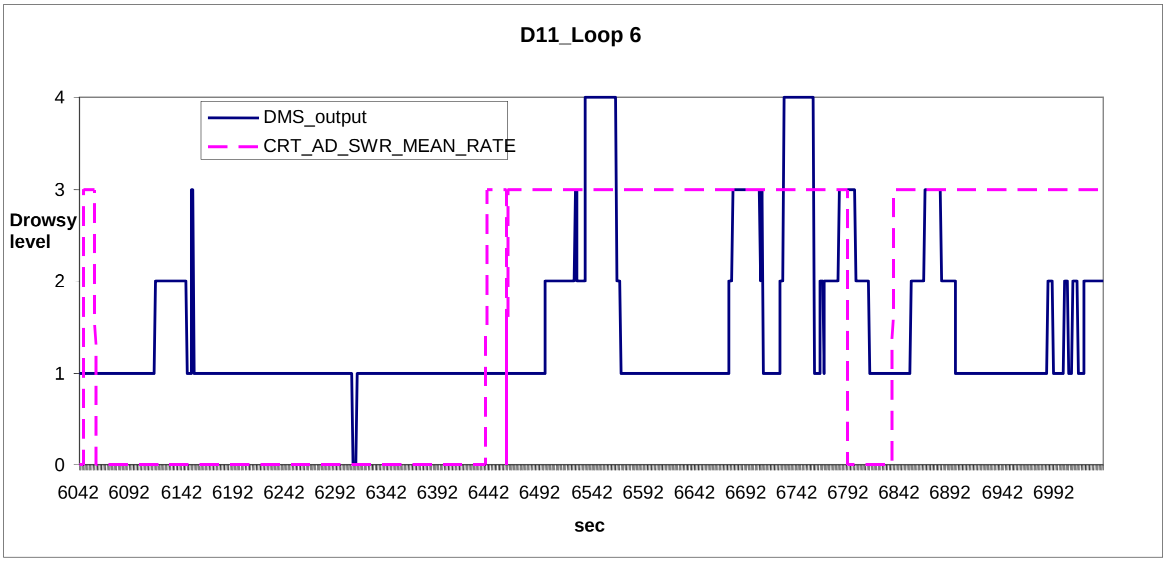


Figure 110: Example for time line of single outputs – Driver D11- loop 6

The outputs of the DSA are highly correlated with the subjective KSS drowsiness ratings by the drivers. Typically, first "slightly drowsy" events classified by the DMS occur in loops where the driver rated his drowsiness at level 7 of the KSS ("sleepy, but no effort to stay awake"). Drowsy and sleepy events either detected by direct or indirect monitoring usually do not occur before a KSS-level of 8 ("sleepy, some effort to stay awake") and 9 ("very sleepy, great effort to stay awake").

"Drowsy" events classified by indirect monitoring usually occur before "drowsy" events classified by direct monitoring, but after the occurrence of first "slightly drowsy" events classified by direct monitoring.

For direct monitoring it could be observed that "drowsy" events sometimes did not occur until first sleepy events already have been detected. This leads the driver's state sometimes directly jumping from "slightly drowsy" to "sleepy". This result corresponds to a choice that was made for the setup of the algorithms. There is a specific process for the detection of the sleepy state that can prevail on the normal strategy. Furthermore, the DMS system provides several modes (demo mode vs. robust mode) with different underlying algorithms for slightly drowsy and drowsy classification.

d) Sensitivity and specificity

For the computation of sensitivity and specificity of the measures the subjective drowsiness rating by the drivers (rated every 20 minutes) was used as criterion. Therefore, KSS-ratings up to 6 were classified as “awake”, ratings of 7 were classified as “slightly drowsy”, ratings of 8 as “drowsy” and ratings of 9 as “sleepy”. Then the concordance between observed drowsiness (KSS self-rating over 20 minutes) and predicted drowsiness (highest state classified by the DSA software within the 20 minute interval) was analyzed. This was done separately for DMS outputs (direct monitoring, see Table 16) and driving related parameters (indirect monitoring; see Table 17). For the computation of sensitivity and specificity, a dichotomisation in two categories was made: “awake”/”slightly drowsy” vs. “drowsy”/”sleepy”. Sensitivity is then calculated by the number of true positives (“drowsy”/”sleepy” drivers correctly identified as such) compared to all observed “drowsy”/”sleepy” events. Specificity is calculated by the number of true negatives (“alert”/”slightly drowsy” drivers correctly identified as such) compared to all observed “awake” / ”slightly drowsy” events. The total number of observations is lower for indirect outputs as they are only available when driving on the Driver Assisted level.

		observed (KSS-selfrating)				total
		awake	slightly d.	drowsy	sleepy	
Predicted (DSM-output)	awake	14	3	3	0	20
	slightly dr.	11	17	5	4	37
	drowsy	1	2	2	3	8
	sleepy	6	5	6	11	28
		32	27	16	18	93

Table 16: Drowsiness states as observed by the KSS-selfrating and predicted by the direct monitoring (DMS drowsiness output)

		observed (KSS-selfrating)				total
		awake	slightly d.	drowsy	sleepy	
predicted (ind.output)	awake	10	5	0	0	15
	slightly dr.	0	0	0	0	0
	drowsy	2	4	3	9	18
	sleepy	2	3	7	5	17
		14	12	10	14	50

Table 17: Drowsiness states as observed by the KSS-selfrating and predicted by the indirect monitoring- please note that the indirect monitoring does not provide a classification of the state “slightly drowsy”)

The analysis reveals a sensitivity of 64.7% and a specificity of 76.3% for the classification of drowsiness by direct monitoring. For indirect monitoring, sensitivity is 100% and specificity is 57.7%. What gets obvious is that direct monitoring tends to produce a higher missing rate but a lower false alarm rate. In contrast, indirect monitoring tends to produce a higher false alarm rate but a low missing rate, as it reliably detects all drowsy and sleepy events. Therefore, the two measures seem to complement each other.

<u>Validation of first version of the distraction detection module</u>

Experimental setup

The study was conducted with N=12 test drivers in the WIVW driving simulator (mean age: 32 years; SD=10 years, 7 male, 5 female). They were the same as in study 1). The test drivers were randomly assigned to 2 experimental groups: 6 drivers drove the test course in the Driver Assisted mode of the HAVEit system. The other 6 drivers drove in the Highly Automated mode of the HAVEit system.

The driving task consisted of driving in a 2 lane motorway of about 20 min length which was repeated 3 times in 3 experimental conditions and with varying sequence of the driving scenarios. The valid speed limit was set to 120 km/h over the whole track. It contained free driving scenarios without any lead vehicle, car follow scenarios with a lead vehicle driving around 110 km/h, traffic jams, forced lane changes at road works and one heavy braking manoeuvre of the lead vehicle (resulting in a take-over request when driving in Highly Automated mode). The drivers were instructed to always stay on the right lane even when the lead vehicle drives slower than allowed, stick to the valid speed limit and only execute lane changes when they are forced to at road works.

In addition, the drivers were instructed to perform a secondary task while driving. It was a hierarchical menu navigation task comparable to a modern in-vehicle information system (IVIS) that can be used for several functionalities (e.g. navigation system, vehicle data, entertainment functions and telephone) by one single display and one single controller.

Figure 111: Display of secondary task presentation in the middle console with joystick (left) and contents of the hierarchical menu (right).

The menu system was presented on a visual display at a lower position in the central console (approximately 34° to the right; 23° down, depending on the driver's seat position). To navigate within the menu, a commercially available joystick was used. The driver was instructed to navigate to a specific menu function (e.g. control average fuel resumption). The task was completely self-paced and interruptible. As soon as the driver confirmed the correct option, a new task could be started. Figure 111 shows an extract from the menu system and the positioning of the secondary task inside the vehicle. The drivers were instructed to prioritize the primary driving task and to perform the menu task only when the situation allowed it.

For a first test of suitable interventions in case of distraction a so called Attention Monitor was implemented. The interventions of the Attention Monitor are based on the calculation of a continuous distraction score that depends on the type of the distraction and the time the driver is engaged in the secondary task (for a detailed description of the algorithm see deliverable D32.1). As soon as a certain threshold is exceeded the Attention Monitor starts the escalation and gives respective feedback to the driver. For a more detailed description of the intervention strategy, see Deliverable D33.3.

In the study two variants of the Attention Monitor were implemented: AM 10: Attention Monitor with a threshold of distraction score = 10 (meaning after 5 sec uninterrupted menu navigation) and AM 20: Attention Monitor with a threshold of distraction score = 20 (meaning after 10 sec uninterrupted menu navigation). In order to evaluate the effects of the Attention Monitor a control condition without any interventions was introduced as a baseline. Each driver performs 3 drives of 20 minutes with each of the 3 AM variants in fully counterbalanced sequence.

Results

a) Direct monitoring (DMS output)

Identifying visual distraction caused by the secondary task used in the present study seemed to be somewhat problematic for the DMS system:

- As the inattentiveness diagnostic is based on driver's head/face positions >20° to the left or the right the system is not able to detect glances to the display when the face is not moved or only slightly moved towards the same direction - this varies individually for each driver. In the present study 3 of the drivers did not move their head towards the display at all. 4 others moved it only slightly with the result that detection rate was also very low. However, this seems to be problematic as the display of the secondary task was located on a position in the central console that is typically for an IVIS (in-vehicle information system). For later application in the vehicle it might be that especially the interaction with these devices will get undetected by the camera system.
- Head movements to the mirrors are detected quite well - however, especially in lane changes this is more an indicator of high attention instead of distraction. Therefore, the output of the inattentiveness diagnostic might be ignored in this special situation. However, one has to handle that case carefully because even if the driver is performing a manoeuvre he can for example spend too much time looking to the lateral rear view mirror and this can lead to a critical situation.
- The face recognition for the inattentiveness diagnostic seems to be very susceptible to driver's seat position and body movements. As soon as the driver starts to move on his seat tracking gets instable. One explanation is that face recognition detection and tracking is very sensitive to the position of the camera. Therefore, an optimum position of the camera must be found for each in-vehicle implementation. Experience has proven that such optimum position exists for all vehicles.
- In case of tracking problems the default value "distracted" is triggered: this leads to a high number and a long duration of false alarm events for some drivers - in extreme cases the driver is identified as distracted over nearly the whole drive - it is recommended to include a kind of plausibility check in the algorithm in order to define an additional state "undefined".

b) Indirect monitoring (parameters from driver's interaction with secondary tasks)

Driver's interactions with secondary tasks were restricted to the interaction with the HAVEit system (operation of the ACC lever) and the interaction with the IVIS (operations with the joystick).. All inputs were reliably detected and computed by the DSA software. In the validation study it was also analysed how driver's would accept interventions (mainly information and warnings to make the driver attentive again) that are triggered at thresholds of a global distraction score of 10 respectively 20.

Further results from the study showed that drivers expected the thresholds to be adapted to the current driving situation (e.g. higher threshold in situations with low demands as traffic jams, lower threshold in situations with high demands as sharp curves). Also the automation level might be considered in the classification of thresholds at least for reaching a higher acceptance by the drivers.

Improvements of the distraction detection module during the optimization phase

a) Direct monitoring

Basically, the DIM module provides general information about the head position: On road /off road[53]. The algorithm principle is based in a pose learning approach. Two categories of poses are distinguished: On-road poses (driver is attentive) and off-road poses (driver is inattentive or distracted). Poses are learnt off-line to train the pose detector. The learning is based on the acquisition of some specific face components like left and right eyes, mouth etc. so to cover a wide range of face morphologies, glasses, head poses and scenarios. This approach allows high robustness to environmental conditions, partial occlusion and glasses. Additionally, the real time tracking of the components provides supplementary information about the general direction of the head when off-road:

- Off Road + left
- Off Road + right
- Off Road + up
- Off Road + down

The assessment of the head orientation information is based on the tracking of the components trajectories previous to the off-road situation. If component movement amplitude is above a given threshold then rotation of the component is detected. The head rotation is achieved by majority vote. In order to guarantee a robust detection of the rotation 2 consecutive same head rotations are necessary to validate a head rotation or 1 head rotation followed by not-measurable image (lack of components). The following Figure 112 is presenting a typical situation of head rotation to the right side.

The DIM is also capable to avoid false alarms in case of camera occultation by detecting objects in front, hand on top of steering wheel, driver's arm etc. Three binary occultation outputs can be provided: left, central and right. Occultation is mainly characterized by a local increase of the image brightness. The corresponding occultation detection algorithms are measuring the image brightness parameters and specific Region of Interest (ROI). Then rule based detectors take occultation decision on the basis of the number and location of bright ROIs (see Figure 113).

[53] *On/Road information is characterized by the following scenario*
Head in direction of the road
Head is turned left or light less than 20°
Head is bended downward less than 20°
No limit upward as the driver can still easily look at the road
Off road information is characterized by the following scenario
Head is turned left or right more than 20°
Head is bended down more than 20°
Head is turned toward the radio/central console
Head is turned toward the central rear view mirror
Head is turned toward the right or left rear view mirror
Head is turned toward the passenger seat
Head is turned backward (e.g. looks at a children seated at the back)
Head bended downward

The evaluation of the capacity of the system to detect occultation has been performed on a real driving data base. An occultation considered there is either a left, right or central occultation. Table 18 below is presenting the results achieved on this data base. It demonstrates very good efficiency of these algorithms.

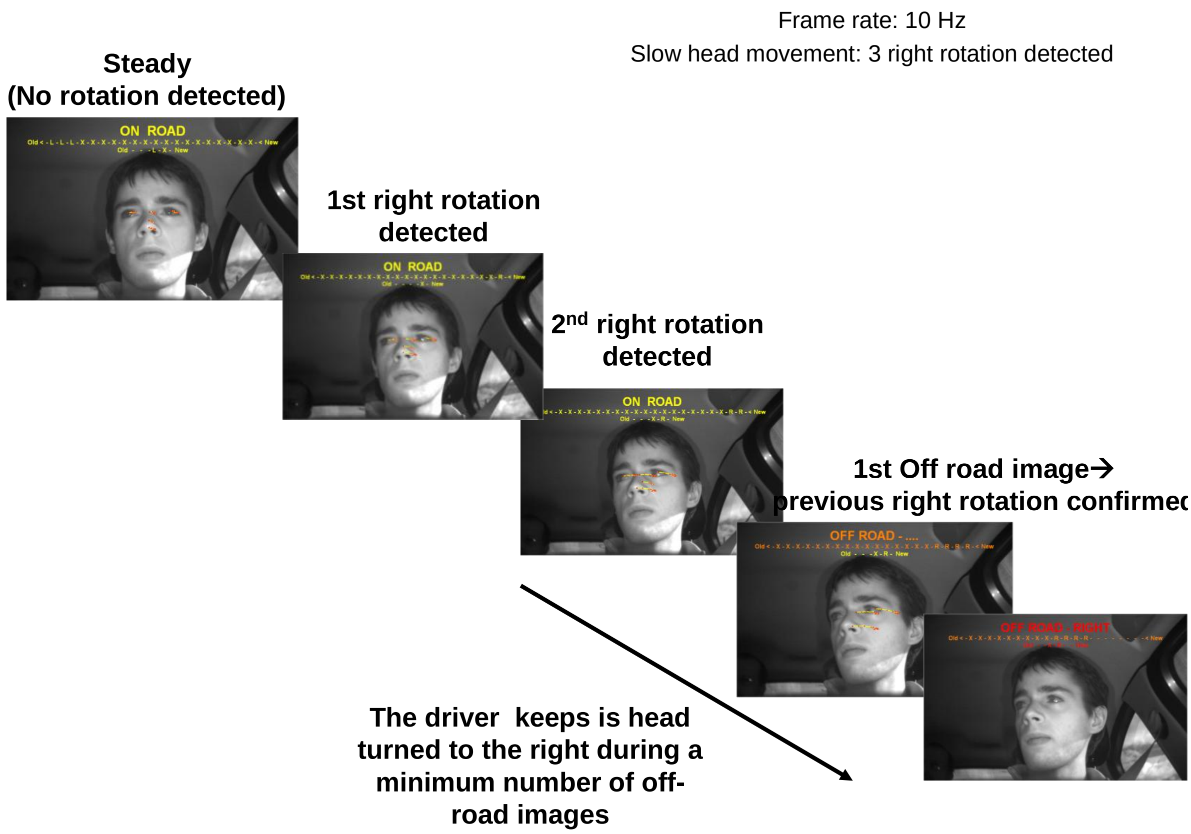


Figure 112: Typical sequence of head rotation detection and classification

Figure 113: Example of occultation

		Ground Truth	
		Positive	Negative
Occultation diagnostic	Positive	True positive: 6895	False positive: 7782
	Negative	False negative: 37	True negative: 103220
		Sensitivity: 99,47 %	Specificity: 92,99 %

Table 18: Evaluation of the performances of the DIM occultation algorithms

b) Indirect monitoring

Following modifications of the algorithms were implemented and demonstrated in the final version of the DSA modules in the HAVEit demonstrators:

- The computation of the continuous global distraction score was modified with reference to a slower increase of the values.
- The distinguished states and sub-states that were mainly designed for the drowsiness algorithms were also considered in the distraction algorithms (e.g. exclusion of inattentiveness diagnostic in lane changes).
- The classification thresholds of distracted / not distracted were adapted to the driving situation and the automation level.

c) Fusion of direct and indirect monitoring

The following proposals are made for the fusion of direct and indirect monitoring:

- Several analyses revealed that direct und indirect monitoring seem to complement each other - currently there are no indicators that one method is in general better than the other and should therefore be prioritized. However, there will be situations where the camera will provide more reliable outputs than the indirect monitoring and vice versa. This will heavily depend on the signal qualities on which the outputs can rely on. The signal qualities are already considered in the software architecture by including the confidence levels of the single outputs. Only if the confidence has been high enough within the last observed time window the measures are included in the algorithms. A further look will be made into more sophisticated fusion algorithms if they provide better results as the currently proposed disjunction.
- The proposed conjunction of direct and indirect measures for the „unresponsive“ state should be skipped. The camera will not be able to detect the driver for example, suddenly getting unconscious, and therefore not longer be able to react at all. Really critical events resulting from an unresponsive driver can be reliably classified by indirect parameters. However, it must get clear that this might be too late to make any useful interventions despite to execute a minimum risk manoeuvre. In the optimum case the DSA had detected a critical driver state before that “unresponsive” level.

Fusion of distraction and drowsiness detection

For a reduction of false alarms both in distraction and drowsiness detection the algorithms should be fused. In order to reduce false alarms in distraction detection, it should be thought of excluding glances to the mirrors detected by the inattentiveness diagnostic if the driver is currently performing a lane change manoeuvre.

If abnormal driving behaviour (e.g. worse lane keeping performance) detected by indirect drowsiness monitoring occurs simultaneously with the detection of distraction (either by driver inputs or by the camera) the driver should not be identified as “drowsy” or “sleepy” but as “distracted”.

In order to reduce false alarms in drowsiness detection, DMS drowsiness outputs should be ignored if the driver is detected as engaged with a secondary task by indirect distraction monitoring. This should make the decisions more robust. All external information that can be complementary to the monitoring is well suited for plausibility checks of the single DSA outputs.

Summary

From the analysis of the single outputs and the timely correlations between them it is to be concluded that the two measures direct and indirect monitoring could complement each other. There are no indicators that one method is in general better than the other and should therefore be prioritized. However, there are situations where the camera will provide more reliable outputs than the indirect monitoring and vice versa. The signal qualities are already taken into account in the software architecture by considering the confidence levels of the single outputs. Only if the confidence has been high enough within the last observed time window the measures are included in the algorithms.

The driver state assessment (including fusion of direct and indirect measurements) was implemented on the CSC during the course of the project. Complete modules with the integrated software were provided to partners Continental, Volvo, VW and DLR. These modules were successfully installed in the demonstrator vehicles. Both driver distraction and driver drowsiness are considered in the different highly automated vehicle applications that were presented within the frame of the HAVEit final event.

5.3 Challenge 3.3: Joint System design – interaction between driver and co-system

In other domains, there is already quite some experience with highly automated vehicles. In aviation for example, highly automated airplanes, with fly-by-wire systems, glass cockpits and automated flight management & control systems are used for decades with an excellent safety rate. From aviation it is also known that there can be side effects especially regarding the interplay between automation and the operator. “Mode confusion” and “out-of-the-loop” situations are just examples of a variety of effects that can arise with automation.

Not all the experiences of aviation can be transferred to the automotive domain: On the one hand, automotive automation might be simpler because the guidance and control task has 2 dimensions instead of 3. On the other hand, the critical time constants of driving are usually smaller, the environment less structured and therefore more challenging, compared to flying a wide-body airplane. The automation and human machine interface (HMI) has to be simpler because the operators, here drivers, are usually no professionals, with only limited chances for selection and additional simulator training. The transitions and HMI design of HAVEit systems will have to be much simpler compared to a modern flight deck, but to reach this simplicity, we have to develop and test those systems at least as systematic and carefully as in aviation in order to use the chances of highly automated systems without loosing the chances to the risks.

In HAVEit, risks and chances of highly automated systems are balanced by a iterative approach of prototyping and tests, starting with an initial specification, a clear definition of the use cases, especially regarding the automation transitions, by an instantiation of use cases into target scenarios, by designing the interaction together with experts and users, implementing these systems with rapid prototyping, and by testing concepts and prototypes in simulators and test vehicles. The following section describes the first round of this iterative approach of developing, testing and refining, with a first design already tested with external test persons in a motion based simulator.

The Joint System is the key element for all highly automated vehicle applications. In HAVEit, therefore strong emphasis was put on the suitable design and evaluation of the Joint System. Results of the different design steps are described in very detail in the public version of the HAVEit deliverables D33.1 (scenario modelling), D33.2 (preliminary concept on optimum task repartition in the Joint System driver / co-system), D33.3 (validation of the first designs by simulation) and D33.6 (final concept for optimum task repartition in the Joint System). To keep the size of the HAVEit final report at reasonable level, in this document, we only summarize the different design and validation steps and the intermediate results to conclude the key findings that led to the final design which was implemented in the HAVEit vehicle demonstrators and presented to the public during the HAVEit final event in June 2011.

5.3.1 Structuring the use space: use cases and scenarios

Design step one consisted in the structuring of the use space. Structuring the use space of a complex human-machine system is not a trivial task: How do we make sure that the system works as intended in all situations? The following section structures the use of the HAVEit systems with the help of three interrelated concepts: *Use space* as a collection of the most important use dimensions, *use case* as one specific combination of use dimensions that usually leads to a sequence of simple steps, and *scenario* as a combination of use cases specified to a level of detail that it can be tested in simulators and test vehicles.

In general, use cases describe “the interaction between a primary actor (the initiator of the interaction) and the system itself, represented as a sequence of simple steps. Actors are something or someone which exists outside the system under study, and that take part in a sequence of activities in a dialogue with the system to achieve some goal. They may be end users, other systems, or hardware devices. Each use case is a complete series of events, described from the point of view of the actor.” (Source: Wikipedia.com[54]). In the context of HAVEit, a typical use case would be “driving and transition from manual to highly automated level”.

In HAVEit deliverable D11.1 several use cases were defined, generic use cases for the Joint System and specific use cases for the demonstrators of the SP 5000 partners. Some of the use cases defined in deliverable D11.1 overlap, others are specific for one partner. The following overview assigns the different use cases defined in deliverable D11.1 to one common framework, the use space. The use space contains all relevant dimensions of use of which we assume that they might have an influence on the design.

[54] Wikipedia- the free encyclopedia. Available at: http://en.wikipedia.org/wiki/Use_case. Search for “use case”. [30.1.2008]

5.3.1.1 The HAVEit use space and its dimensions

In the mind map (Figure 114) the use space, scenarios and demonstrator applications of HAVEit are depicted.

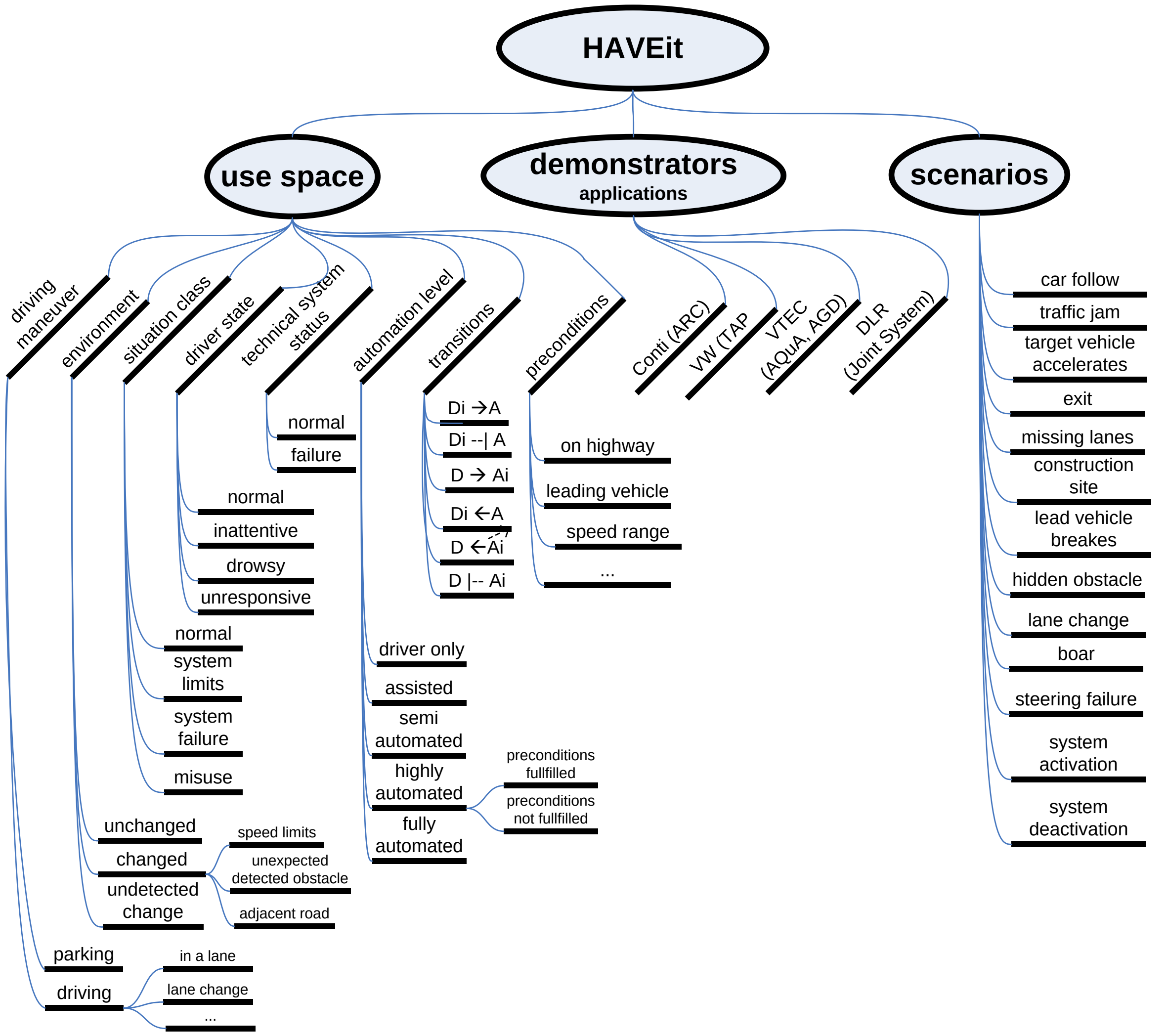


Figure 114: HAVEit mind map of use space, scenarios, and demonstrator applications

Use space

The use space contains several subdimensions that classify the use of the HAVEit system. The most obvious dimension of the use space is the driving manoeuvre itself, here classified as *parking* or *driving* and further specified as *driving in a lane* or *changing the lane.* Other sub-classifications of the driving manoeuvre could be added if necessary. Closely related to this dimension is the environment. The environment can be classified as *unchanged* or *changed*. A change of the environment can be caused e.g. by changing speed limits, unexpected detected obstacles, the appearance of an adjacent road or other events. The change can influence the automation behaviour and might lead to a transition request depending on the automation limits. Undetected changes are changes that are not detected by the automation itself and could lead to an automation break down without previous warning.

The third dimension is the classification of the situation class in *normal* situations, situations at *system limits*, *system failure* and *misuse*. According to the Response Code of Practice CoP[55] this classification of situations is essential for assessing and evaluating the controllability of the vehicle automation. It is important to keep in mind here, that the overall system in the context of HAVEit consists of the subsystems driver, automation and base vehicle, so system limits can be limits of the driver, the automation or the vehicle. Normal situations are situations in which all the subsystems of the driver-automation-vehicle system, the assistance or automation, the driver and the vehicle perform as specified. Situations at system limits are situations with which any of the subsystems can not cope with and therefore refuse activation or initiates a transition back to a lower automation level or to the driver. In situations in which a system failure occurs the automation suddenly breaks down for example due to a sensor failure. This means that there might be only little time for the driver to take back control. In the worst case the system failure is not detected by the automation so that the driver has to take over control without warning.

Closely related to the “situation class” are the dimensions “driver state” and “technical system status”. These two dimensions specify the condition of the driver and the automation. An example in the HAVEit context is when the driver is *unresponsive* to take-over requests, that might lead to a transition towards another automation level, e.g. to a minimum risk state.

Another dimension is the “automation level” in which the vehicle is driving. For HAVEit several levels of automation are possible in the design space. These are the levels Driver only, Driver Assisted, Semi Automated and Highly Automated and, in limited circumstances, Fully Automated (see also deliverable D11.1, for further background on assistance and automation spectrum or Flemisch et al. 2008[56] [4]).

In Figure 115, potential automation levels of the HAVEit design space are depicted in a spectrum or scale. The spectrum can be seen as a mapping of different degrees of driver involvement in the driving task. On the left border the spectrum is defined by the level Driver Only, in which the driver has full manual control over the vehicle and on the right border by the level Fully Automated, in which the automation. The level Driver Assisted can be divided in assistance by feedback and by support. Assistance based on feedback includes informing feedback from the system like acoustic, visual, or vibratory lane departure warnings and parking assistance, but not yet direct intervention of the automation with the primary driving task. Assistance by support can include aspects of assistance by feedback and is characterized by direct intervention during the primary driving tasks, if the current vehicle situation is considered to be too dangerous. Functions are lane keeping support, emergency braking and avoiding of objects, for example. *Semi Automated* describes the automation of about half of the driving task. One example is the Adaptive Cruise Control (ACC), where the complete longitudinal control is done by the automation (sublevel: “Longitudinal”). Other examples are some lane keeping systems that can almost do the complete lateral control of the vehicle (sublevel: “Lateral”). In the level Highly Automated there is an automation for longitudinal and lateral vehicle guidance but compared to fully automated the driver is still involved in the driving task most of the time. This level can be subdivided in hands on and hands off driving.

Additional States are the state *off* where the automation is completely switched off, the *minimum risk state* which the automation tries to reach in case the automation is no longer capable of performing a certain level of automation and the driver does not take over control and the state *failure* in which the automation works incorrectly or not at all.

[55] RESPONSE Consortium (2006). Code of Practice for the design and evaluation of ADAS. RESPONSE III: A PReVENT project

[56] Flemisch, F. O., Kelsch, J, Löper, C., Schieben. A. & Schindler, J. (2008). Automation spectrum, inner/outer compatibility and other potentially useful human factors concepts for assistance and automation. In D. de Waard, F. O. Flemisch, B. Lorenz, H. Oberheid and K. A. Brookhuis. Human Factors for Assistance and Automation. Maastricht: Shaker Publishing.

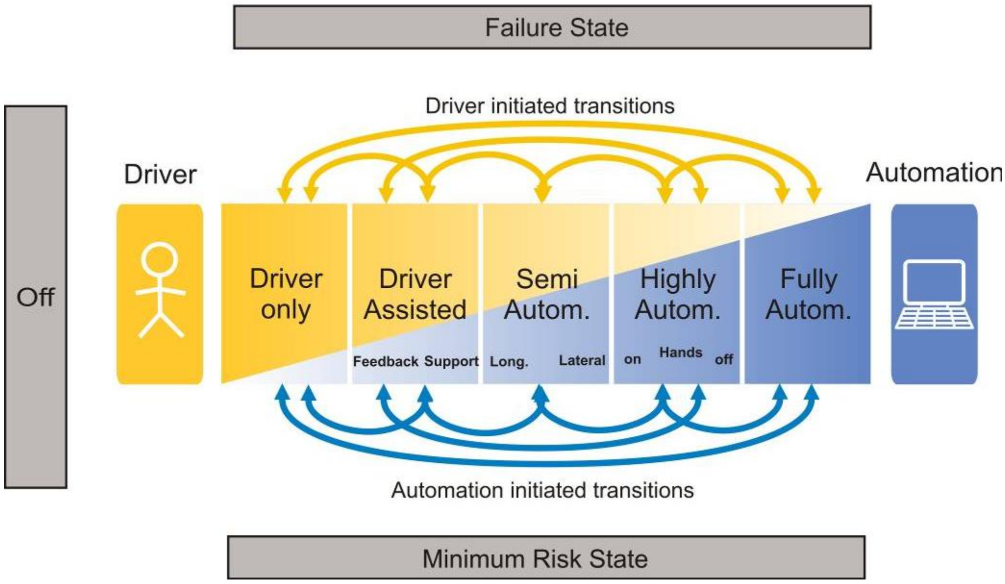


Figure 115: Levels of automation and all possible transitions in the HAVEit design space

The situation classes mentioned above and the automation levels are interrelated. Depending on the automation level the definition of normal situations and system limits can vary. For example, a vehicle driving in Semi Automated ACC mode (longitudinal control automated) can cope with specific changes in the environment that it can not cope with in Highly Automated mode. Loosing track of the lane markings is, e.g., not relevant for the ACC mode. For the Highly Automated mode loosing track of the lane markings is a situation at system limits and this leads to a transition of control back to the driver. System failures that mean a break down of the automation could occur within all driving situations and automation levels.

Within the dimension “automation level” all assistance and automation can be specified according to the preconditions for activation. These preconditions can be either fulfilled or not fulfilled.

Related to this is a further dimension of the use space: the “transitions” that can occur. The transitions are depended on the automation level and the situation class. As described in deliverable D11.1 transitions can be classified by the direction of transition (control towards automation/control towards driver) and initiation of the transition (by automation/by driver). In addition, the status of the transitions can be defined. Transitions can be either successful if the control is transferred successfully or transitions fail or are refused in case the driver or the automation does not take over control as expected. In a short notation the driver is symbolized by the letter “D”, the automation by the letter “A”. The index “i” indicates who initiated the transitions. The direction and the status of the transition is symbolized by an arrow (→). If the transition is refused or if it fails is indicated by a combination of a dashed arrow horizontal and one vertical line (- - >|). The notation Di → A stands for a successful transition from the driver towards the automation. The transition is initiated by the driver.

Table 19 shows an overview of all possible transitions and specifies in which situation classes the transitions can occur. A successful transition of control from the driver towards the automation that is initiated by the driver (Di → A) only occurs in normal driving situations. In these situations all preconditions are fulfilled so that the automation is activated successfully. A driver initiated transition from driver towards automation that is refused (Di - - >| A) occurs in situations at system limits. At system limits the preconditions for activation are not fulfilled so that the transition is refused by the automation.

If an automation initiated transition (D → Ai) should be allowed is a question of design. This transition means that the automation initiates a self-activation or reactivation. If such automation initiated transition needs the confirmation of the driver is not defined yet.

That the driver initiates a transition to take back control is possible in all situation classes (Di ← A). Driver initiated transitions in this direction are expected to be always successful because the automation is not allowed to refuse such a driver command.

In contrast, automation initiated transition from the automation towards the driver can be either successful (D ← Ai) or can be refused (D | < - - Ai). A successful transition means that the driver actively takes back control; a refused transition means that the driver does not react, for example due to drowsiness or inattention and that the automation has to go into the Minimum Risk State (MRS) at worst.

Short Notation	**D_i → A**	**D_i - -|→ A**	**D → A_i**	**D_i ← A**	**D ← A_i**	**D | ← - - A_i**
Direction of transition	Transition towards automation	Transition towards automation	Transition towards automation	Transition towards driver	Transition towards driver	Transition towards driver
Initiation	By driver	By driver	By automation	By driver	By automation	By automation
Status	Successful	Failed / Refused	Successful	Successful	Successful	Failed
Occurrence	Normal driving situation	Situation at automation limits	Normal driving situation	Normal driving situation Situation at automation limits Situation with automation failure	Situation at automation limits	Situation at automation limits Situation with automation failure

Table 19: General classification of transitions between driver and automation

Demonstrators

The dimension "demonstrators" in the use space stands for the different demonstrators that will be build-up and tested during the HAVEit project and that have some relevance for the Human-Machine-Interaction design process. These are the Continental system "Automated Assistance in Roadworks and Congestion" (ARC), the Volkswagen system "Temporary Autopilot" (TAP), the VTEC systems "Automated Queue Assistance" (AQuA) and "Active Green Driving" (AGD) and the "Joint System" demonstrator. Each demonstrator is described in detail in deliverable D11.1. For each demonstrator different use cases are applicable. Which use cases are relevant for each demonstrator depends on the automation levels and the driving situations that are covered by the system.

Scenarios

The scenarios are combinations of use cases specified to a level of detail that it can be tested in simulators and test vehicles. The scenarios listed in the mind map (Figure 114) are examples from the first human factors study that was conducted in the WIVW driving simulator.

Use cases

A fundamental challenge for the design and test of automation system is that many dimensions in use space (see Figure 114) can be combined with each other. If we would design individually for each of these combinations, this would result in an exploding complexity of use cases. We therefore decided to group the use cases with different cuts through the use space in a way that the use case catalogue is not yet complete, but covers the most important use dimensions and is still manageable. For a detailed description of each use case, the reader is kindly referred to deliverable D33.1, section 2.2. The following use cases were defined:

- (Normal) Driving + in a lane + activation possible
- Driving +in a lane +right overtaking
- Driving + lane change
- Driving + activation impossible
- Driving + deactivation necessary (due to detected environment change)
- Driving + Unexpected, detected obstacle in front
- Driving + Environment change undetected by co-system
- Driving + Env. change speed limit
- Driving + Env. change adjacent road
- Driving + Driver inattentive
- Driving + Driver drowsy
- Driving + Driver unresponsive (to transition request) + Transition to minimum risk state
- Driving+ Technical failure
- Misuse

5.3.1.2 From use cases to scenarios

In order to validate the HAVEit systems with reference to their technical functionality and controllability by the driver, the defined use cases have now to be transferred into testable scenarios in the driving simulator environment SILAB. A scenario can be described as a sequence of use cases (with corresponding driving situations) and the transitions between them. These can either be system-initiated transitions because of system limits or driver-initiated transitions (e.g. activation or deactivation of the system).

Typically, the automation level changes within one scenario. An example for a traffic scenario might be: The driver starts in the driving situation “driving in a lane with vehicle in front”- this represents a normal use case with the possibility of *highly automated* driving. Suddenly the target vehicle accelerates and leaves the sensor range – the driving situation changes to “driving in a lane without vehicle in front”. At this system limit a system-initiated transition to lower automation levels is required. After a certain time interval the target vehicle is reached again. The driving situation changes again to “driving in a lane with vehicle in front”. The driver can switch back into a higher automation level. Depending on the combination of several use cases and the transitions between them, various scenarios emerge (see Figure 116).

It is unfavourable to define a fixed set of scenarios which must be used for all investigations. Scenario modelling should be seen as continuous process over the whole period of the project including adaptation of existing scenarios or creation of new scenarios. Based on the specific study question different scenarios will be relevant and should be selected out of an available scenario set to create a suitable test course. A preliminary set of scenarios was implemented in SILAB which was used for the first baseline experiments on transitions. The focus was here on scenarios that require a system-initiated transition at system limits. These first example scenarios are described in detail in deliverable D33.1, section 2.3. To show the general method, exemplarily, two scenarios “car follow” and “missing lanes” are described here.

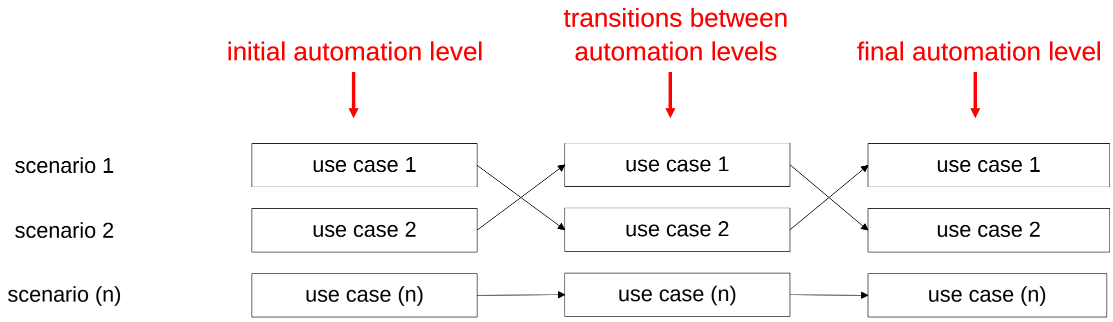


Figure 116: Transformation of use cases into testable scenarios

The scenario "car follow" represents a variant of the use case "(normal) driving + in a lane + activation possible", see Figure 117. It contains a lead vehicle within the sensor range driving within a predefined speed range below v_max of the system ("target vehicle"). There are few other vehicles overtaking on the left lane. In order to get a more realistic driving behaviour the lead vehicle slightly varies its speed within a range of 20 km/h. The scenario can be realized in different variants depending on the selected speed range of the lead vehicle. Assuming a system that requires the presence of a target vehicle for *highly automated* driving, this scenario represents a normal use case allowing the full range of automation levels from *Driver Only* to *Highly Automated*. No system-initiated transitions are required. The scenario can be freely varied in its duration. It can be used to assess the effects of various automation levels in free highway driving on driver's workload and system acceptance.

Scenario: Car follow

- Variant of use case "normal driving in the lane, activation possible"
- Specification: with target
- Car follow within various speed ranges
- Automation level: *Driver Only* to *Highly Automated*
- Activation / driver-initiated transitions possible
- No system-initiated transitions required

Figure 117: Scenario „car follow"

The scenario "missing lanes" (see Figure 118) is a combination of the use cases "(normal) driving in a lane" and "deactivation necessary - due to environmental change - bad lane markings". In a predefined section (e.g. 2000 meters) the lane markings are absent. Assuming a system that requires the detection of lane markings via sensors to drive in *Highly Automated* mode, the system will initiate a transition to lower levels of automation (either *Driver Only* or *Driver Assisted*). As soon as the lane markings are present again, the driver can change back into *Highly Automated* mode. The scenario can be used to assess driver's mode awareness as well as driver's reactions to transitions.

Scenario: Missing lanes

- Combination of use cases: “normal driving in a lane”, “deactivation necessary- due to environment change”
- Missing lane markings on the road producing sensor problems
- System-initiated transition to lower automation level required

Figure 118: Scenario “missing lanes”

5.3.2 Joint System design

5.3.2.1 Human-Machine-Interaction design process

In HAVEit an intense Human-Machine-Interaction (HMI) design process was started. All vehicles owners met several times under the lead of DLR to discuss and harmonize the selection and design of the automation levels, the transitions between these levels and the interfaces like displays and switching devices need for a highly automated vehicle. The challenges was to align the interaction design between the demonstrator vehicles in a way that one common HAVEit design was recognizable and the driver was able to easily switch between the vehicles. On the other hand the specifications needed to be so flexible that the company specific design could be shown. Some of the HMI meetings took place at DLR in the laboratory SMPLab (Figure 119). The SMPLab is equipped with the Theatre-Technique, a technique that allows driving through different scenarios with a vehicle automation that is played by a human (e.g Schieben et al. 2009[57], Flemisch et al. 2005[58]). The participants went through the HAVEit scenarios playing and designing what the automation should do in different situations and which transitions should be allowed.

The results of the theatre-sessions were depicted in form of (UML adapted) diagrams that show the human-machine interaction on different interfaces and the allocation of control between the driver and the automation. Figure 120 shows an example of such a diagram of the interaction.

Following the described procedure all interactions for the transitions mentioned in the use cases above were specified. All interaction diagrams for the HAVEit use cases can be found in D. 33.2. “Preliminary concepts on optimum task repartition for HAVEit systems”. The interaction design was tested in several studies in simulators and test vehicles to find out more about the appropriateness and acceptance of the chosen design.

[57] Schieben, A.; Heesen, M.; Schindler, J.; Kelsch, J; Flemisch, F.: The theater-system technique: Agile designing and testing of system behavior and interaction, applied to highly automated vehicles; Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI); Essen; 2009
[58] Flemisch, F.O.; Goodrich, K.H.; Conway, S.R.: At the crossroads of manually controlled and automated transport: The H-Metaphor and its first applications (progress update 2005); ITS’05; Hannover, 2005

Figure 119: HMI-Workshop at DLR using the Theatre-Technique for discussing the HMI design

For the transitions between the levels of automation specific studies were conducted because there were some open questions regarding the best transition design for driver initiated transitions (see D. 33.6 “Validation of concept of optimum task reparation” for a detailed report). Based on the study results and further discussions among all partners a final transition design was defined for all demonstrator vehicles. This design tries to be as consistent as possible for the transition design of different modes (here: Semi Automated and Highly Automated) and between the demonstrator vehicles. The design is summarized in the tables below (Table 20 - Table 24).

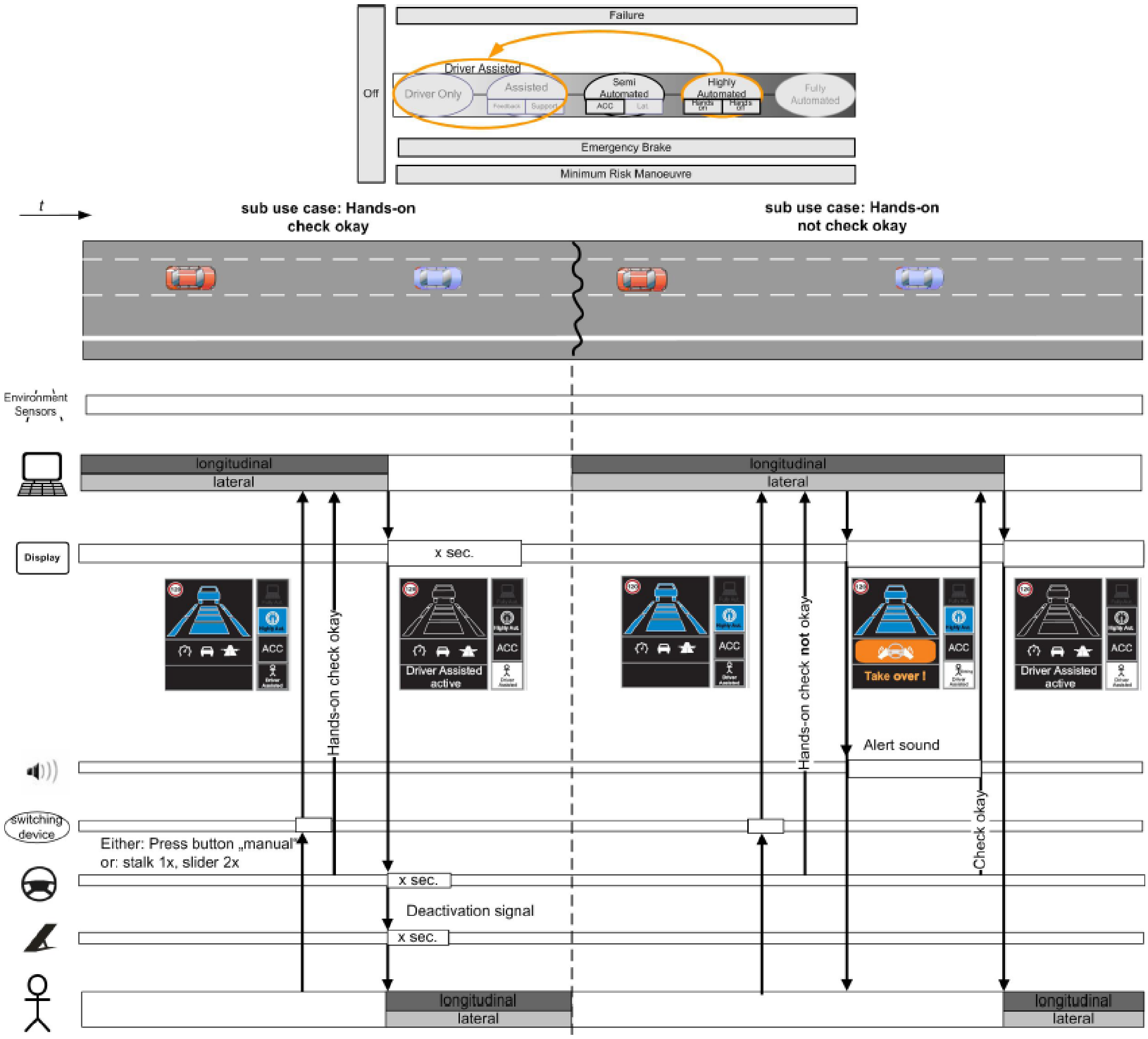


Figure 120: Driver initiated deactivation of the level *Highly Automated:* Sequence diagram of the interaction design in the Joint System demonstrator

Prototype	JSD	TAP	ARC	AQuA
SEMI AUTOMATED: Acceleration <u>up to</u> system speed limit or speed limit of road section				
				
Transition	• slight acceleration: increase of set speed • strong acceleration: transition DA_i←SA	• no transition, overriding of set speed	• no transition, overriding of set speed	• no transition, overriding of set speed
Interaction elements	• warning sound for transition • force threshold and vibration on accelerator pedal	---	---	---
SEMI AUTOMATED: Acceleration <u>beyond</u> system speed limit or above speed limit of road section				
				
Transition	• slight acceleration: change of set speed is blocked • strong acceleration: transition DA_i←SA	• no transition	• no transition	• no transition
Interaction elements	• warning sound for transition • force threshold and vibration on accelerator pedal	---	• force threshold and vibration on accelerator pedal • visual information about the speed limit	---

HA: Highly Automated, DA_i←SA: driver initiated transition from Semi Automated to Driver Assisted

Table 20: Transition design for the level Semi Automated: Driver accelerates up to and beyond the system speed limit

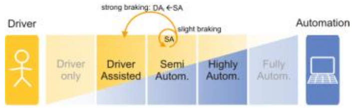
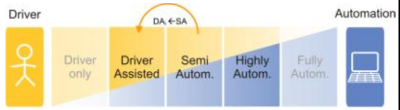
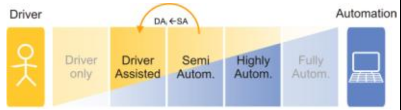
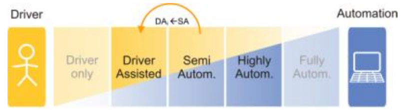
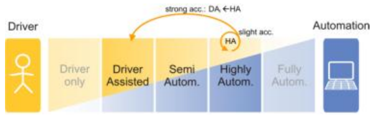
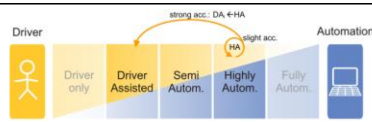
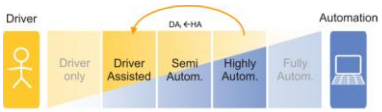
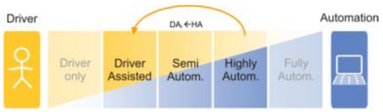
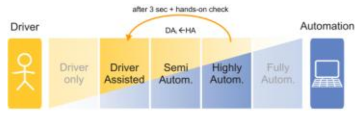
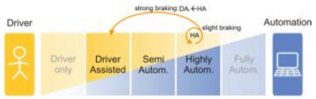
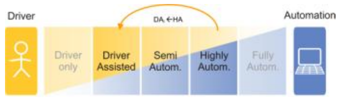
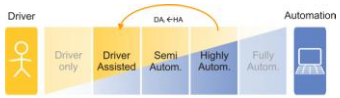
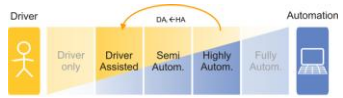
Prototype	**JSD**	**TAP**	**ARC**	**AQuA**
		SEMI AUTOMATED: Braking		
				
Transition	• slight braking: reduction of set speed • strong braking: transition $DA_i \leftarrow SA$	• transition: $DA_i \leftarrow SA$	• transition: $DA_i \leftarrow SA$	• transition: $DA_i \leftarrow SA$
Interaction elements	• transition sound	• no warning sound	• transition sound	• transition sound

Table 21: Transition design for the level Semi Automated: Driver brakes

Prototype	JSD	TAP	ARC	AQuA
HIGHLY AUTOMATED: Acceleration up to system speed limit or speed limit of road section				
				
Transition	• slight acceleration: increase of set speed • strong acceleration: transition $DA_i \leftarrow HA$	• no transition, overriding of set speed	• no transition, overriding of set speed	• no transition, overriding of set speed
Interaction elements	• warning sound • force threshold and vibration on accelerator pedal	---	---	---
HIGHLY AUTOMATED: Acceleration beyond system speed limit or speed limit of road section				
				
Transition	• slight acceleration: change of set speed is blocked • strong acceleration: transition $DA_i \leftarrow HA$	• slight acceleration over 130km/h is okay, faster driving will lead to a transition: $DA_i \leftarrow HA$	• threshold on the acc. pedal • if strong acceleration: transition: $DA_i \leftarrow HA$	• for x sec. override of set speed • after x sec: when hands-on check okay, transition: $DA_i \leftarrow HA$ • if hands-on check not okay: warning and then transition $DA_i \leftarrow HA$
Interaction elements	• warning sound for transition • force threshold and vibration on accelerator pedal	• warning sound for transition	• visual warning about the speed limit • sound for mode degradation if driver doesn't react.	• transition sound or • warning sound when hands-on check not okay

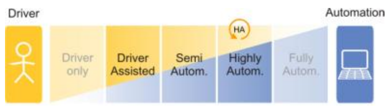
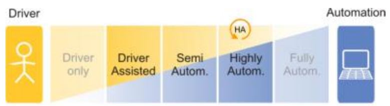
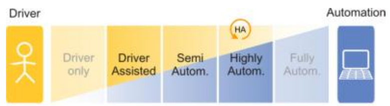
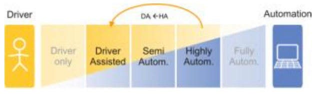
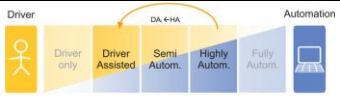
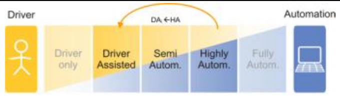
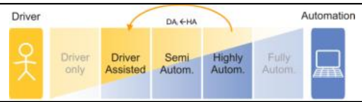
HA: Highly Automated, $DA_i \leftarrow SA$: driver initiated transition from Semi Automated to Driver Assisted

Table 22: Transition design for the level Highly Automated: Driver accelerates up to and beyond the system speed limit

HIGHLY AUTOMATED: Braking				
Prototype	**JSD**	**TAP**	**ARC**	**AQuA**
Braking				
Transition	• slight braking: reduction of set speed • strong braking: transition DA_i←HA	• transition: DA_i←HA	• transition: DA_i←HA	• transition: DA_i←HA
Interaction elements	• transition sound	--	• transition sound	• transition sound

HA: Highly Automated, DA_i←SA: driver initiated transition from Semi Automated to Driver Assisted

Table 23: Transition design for the level Highly Automated: Driver brakes

Prototype	**JSD**	**TAP**	**ARC**	**AQuA**
Lane Change with indicator				
Transition	• no transition: stays in HA	• no transition: stays in HA	• no transition: stays in HA	• no transition: stays in HA
Interaction elements	--	--	--	--
Lane Change without indicator				
Transition	• transition: DA_i←HA	• transition: DA_i←HA	• transition: DA_i←HA	• transition: DA_i←HA
Interaction elements	• transition sound	• transition sound	• transition sound	• transition sound

HA: Highly Automated, DA_i←SA: driver initiated transition from Semi Automated to Driver Assisted

Table 24: Highly Automated: reaction of the HAVEit demonstrators to lane change manoeuvres in level Highly Automated

Mode Selection and Arbitration Unit (MSU)

The previously described transition design and human machine interaction design (HMI) was implemented as software in the Mode Selection and Arbitration Unit (MSU). The MSU calculates if an automation level change is needed by the co-system or requested by the driver. Additionally, the MSU checks if the automation level which is requested by the driver is possible. If the MSU calculates a possible and requested or needed automation level change, a corresponding transition will be initialized. Additionally, the MSU manages the HMI (visual, acoustic, haptic) for these transitions in the Joint System demonstrator.

For the implementation of the MSU of the Joint System demonstrator into software the first step is represented by the specification of the required state machine. All available states and all transitions between these states have to be described there. Figure 121 shows a UML state diagram of the MSU main state machine which is responsible for managing the currentAutomationLevel and requestedAutomationLevel (see HAVEit deliverable D12.1).

Corresponding to the automation level and transition defined during the HMI discussions the UML state machine in Figure 121 shows the levels: *Driver Assisted*, *Semi Automated*, *Highly Automated*, *MRM*/*Emergency* and *Failure*. An additional state is *Off* which is not depicted directly as a box in the UML state machine diagram.

The available transitions are the arrows between these states. Orange arrows are transitions initialized by the driver. Blue arrows are transitions initialized by the co-system. The UML guards at the arrows describe the conditions for the transition.

For better comprehensibility an example of the work of the MSU is given: A driver is driving in the currentAutomationLevel *Driver Assisted*. He/she pushes the button for *Semi Automated*. The MSU checks inside the *Driver Assisted* level if the preconditions for the *Semi Automated* level are fulfilled. If yes, the MSU sets *Semi Automated as* requestedAutomationLevel and initializes the transition (see orange arrow with guard *DI[switch Sa]*) as currentTransition. Inside the currentTransition another state machine coordinates the HMI elements which inform the driver about this transition state. After the currentTransition is completed the MSU also sets the currentAutomationLevel to *Semi Automated.*

The transitions intermitted by the diamond (OR decision in UML) are transitions where the hands-on check for the steering wheel is conducted to ensure that the driver took over lateral control. If the driver took over control, the currentTransition is completed successfully and the currentAutomationLevel switches to the requestedAutomationLevel. If the driver is not responding to the take over request, the currentTransition changes to a higher HMI escalation scheme depending on the criticality of the situation or on the elapsed time. If the driver is still not responding at the end of the highest escalation scheme the requestedAutomationLevel and the currentAutomationLevel have to be reset to *MRM* state.

The transitions which begin with a small red lightning symbol are critical transitions which have to be immediately executed for every element inside the associated sub state (*Failure, MRS/Emergency* and *Normal Driving*). They will also immediately interrupt every currently executed transition.

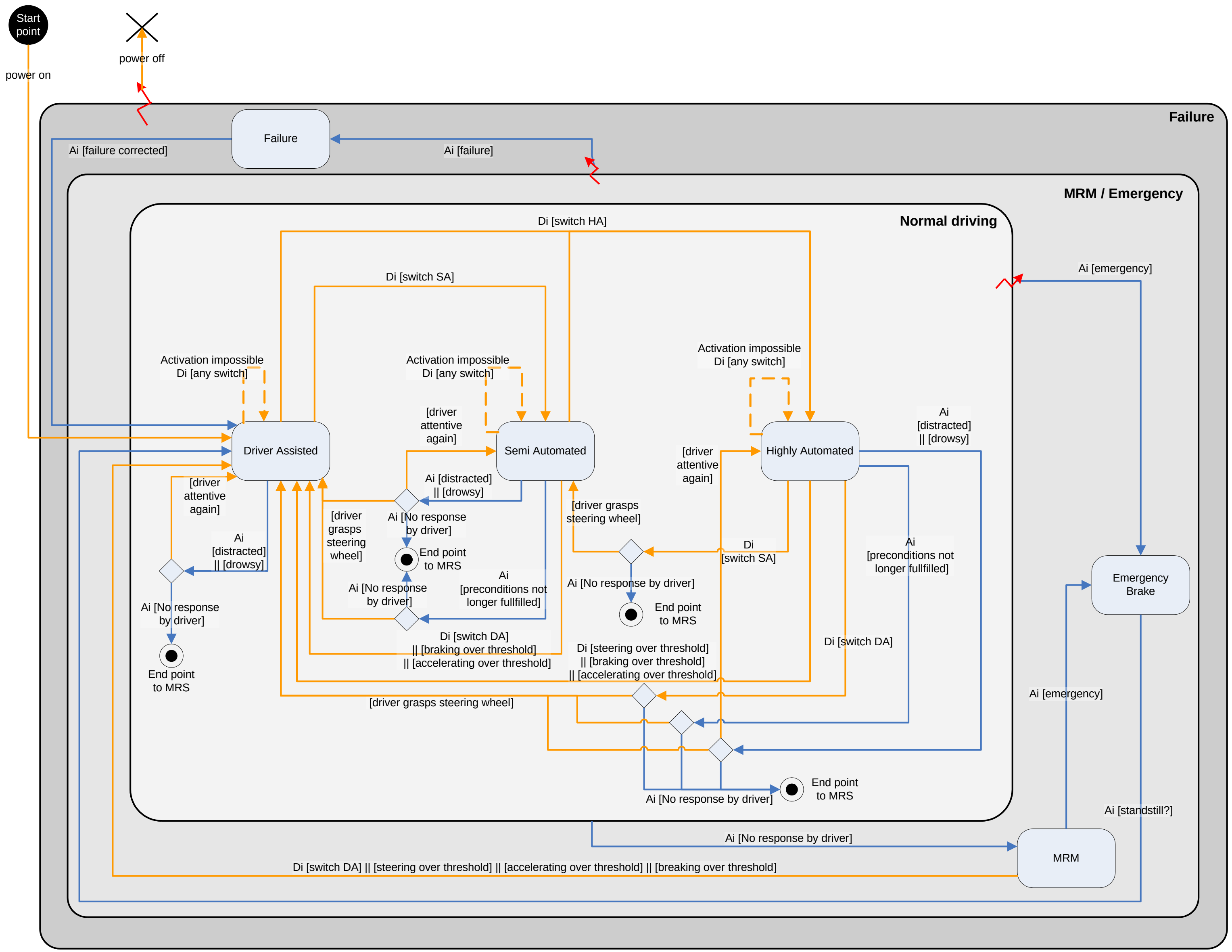


Figure 121: Main state diagram for the Joint System MSU (Mode Selection and Arbitration Unit)

5.3.2.2 Interface design

For HAVEit systems, it was decided to use the following primary driving interfaces: steering wheels, brake pedals, accelerator pedals. The primary driving interfaces for HAVEit systems were defined in the component tree (deliverable D11.1). The inceptors can be common or force-feedback inceptors. Force-feedback means that the inceptor can be used to display haptic information like forces, vibrations or ticks. The steering wheel can be either a force-feedback component that is mechanically connected or a force-feedback by-wire component with mechanical fallback. Additionally, there are interface components listed that are not relevant for steering, braking or accelerating, but directly relevant for the driving task. These components are the direction indicator, the parking brake and the gear lever.

The design options in the design space for communicating with the driver are manifold. To structure this variety an interaction matrix was built that covers the most important communication modalities (Figure 122). The modality of the interaction can be visual, acoustic or haptic. The coding of the interaction can be spatial, by colour or textual for the visual modality, spatial or verbal for the acoustic modality and continuous or discrete for the haptic modality. In addition, haptic information can be transferred by vibrations that are an extra class between continuous and discrete signals. All interaction patterns have a time component that can be varied to code different information.

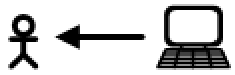
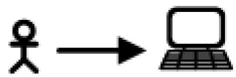
Modality	visual	acoustic	haptic
	*automation mode display*	*warning tone*	*vibration on steering wheel*
	*video of driver for DSA*	*not applied*	*steering torque*

Figure 122: Different interaction modalities for the communication between driver and co-system in the design space

For HAVEit systems it was decided to use visual, acoustic, and haptic communication modalities. The communication is bidirectional and can include information that the co-system gathers from the driver for example via haptic signals (e.g. torque recognition, hands-off detection) or visual (e.g. video monitoring of the driver for Driver State Assessment). It became clear during the specification phase that due to the significance of primary task automation in HAVEit the haptic channel will be an important communication channel for the primary driving task, in multi-modal combination with the visual and the acoustic channels (reference D11.1, section 4.3). Visual information is used continuously to inform the driver about the current automation level and all relevant information related to automated driving (automation level, warnings, take over requests, failure messages etc.). The acoustic channel is used carefully because different acoustic information is already fairly often used in today's cars and might be annoying for the driver.

The display design of the primary display for the HAVEit systems consists of the following information components Figure 123). There is an Automation Mode Display which indicates the current automation level and all available automation levels. Second, there is an Automation Monitor that contains information about the current automation status and its functionality. Third, there is an area for warnings, messages and preconditions called Message Field.

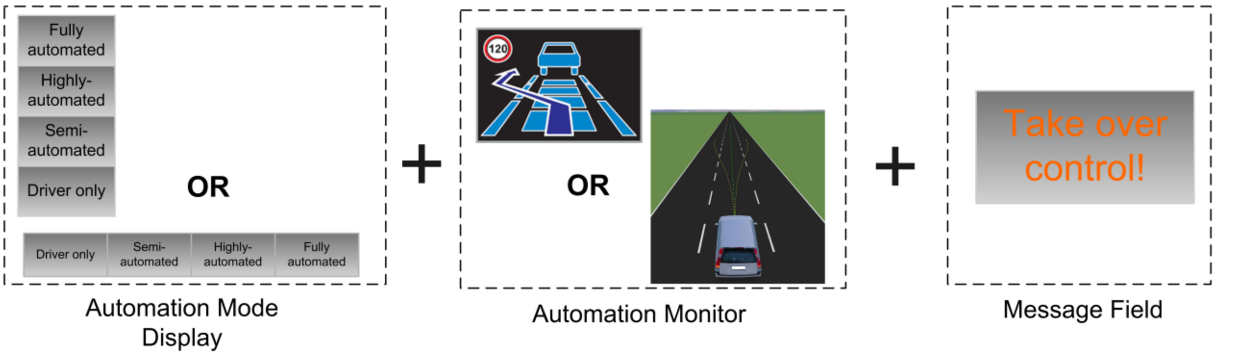


Figure 123: Predefined display components for HAVEit systems: The Automation Mode Display, the Automation Monitor and the Message Field

5.3.2.3 Interaction matrix for the Joint System demonstrator

In order to describe the key issues, the interaction matrix and the interface components are briefly described taking the Joint System demonstrator as an example. The Joint System demonstrator will use visual, acoustic and haptic communication. As depicted in Figure 124 the focus of the communication in direction from the co-system to the driver is seen in visual and haptic communication. A primary display based on the display elements described above informs the driver about the current level and automation status. The acoustic channel is only carefully used. For the haptic communication the Joint System demonstrator applies continuous force, vibrations for warnings and as alive-signal as well as discrete signals like tics on the force-feedback steering wheel and accelerator pedal.

One visual communication channel from the driver to the co-system is provided by the Driver Monitoring Camera. Speech commands are not considered; because of limited reliability of state-of-the-art speech recognition systems their use in safety relevant systems is not feasible. The forces of the driver on the wheel and pedals are used by the co-system to interpret the current driving manoeuvre and to detect the driver state and transition requests.

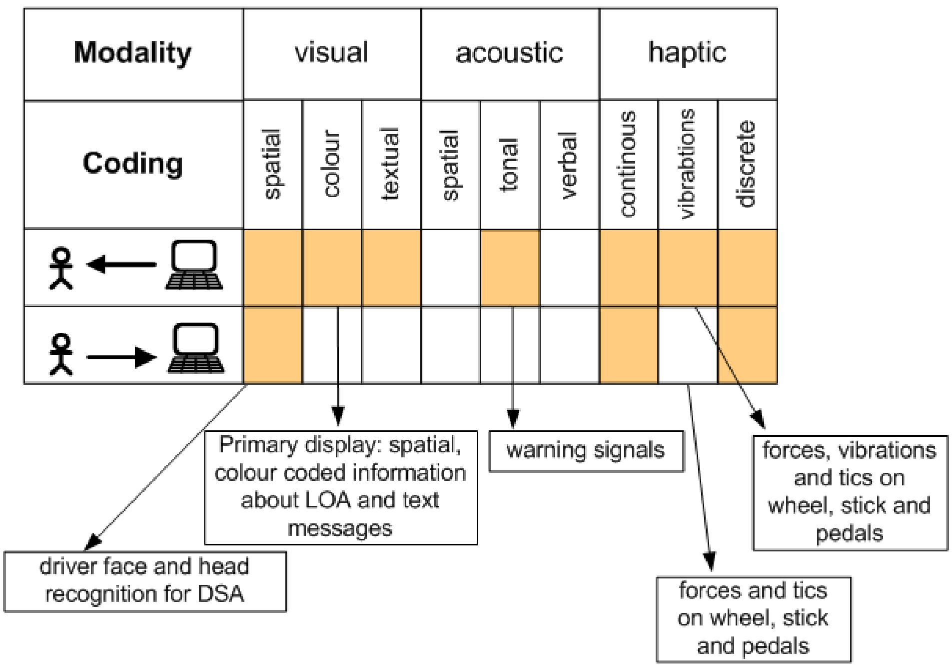


Figure 124: Interaction elements on different modalities for the Joint System demonstrator

Realisation of the different HMI channels in the demonstrator:

- *Haptic components:* force-feedback steering wheel, force-feedback accelerator pedal, (see primary driving interfaces) are used as primary driving interface and additionally for providing haptic feedback to the driver.
- *Acoustic components:* There is a stereo sound system.
- *Visual components:* There is a primary display, a secondary display, LED and flash-lights.

The display design for the primary display is outlined in Figure 125.

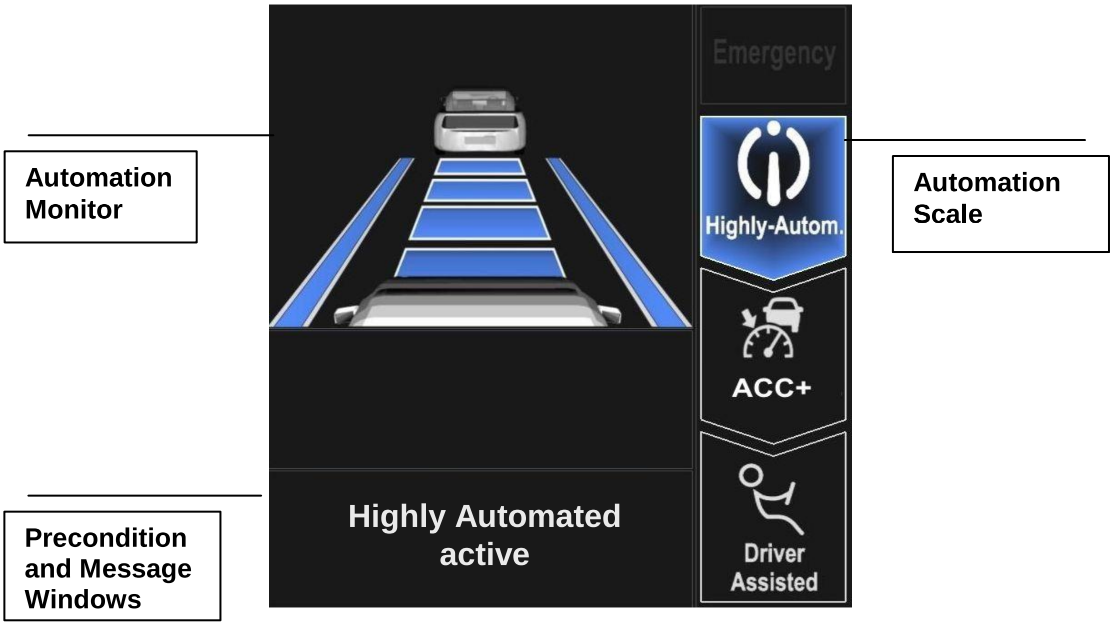


Figure 125: Display design for the primary display of the Joint System demonstrator including the Automation Monitor, the Automation Scale and the Precondition and Message Windows

The Automation Monitor is in the upper left part of the interface. Its purpose is to provide the driver with graphical information about the environment and the form of assistance or automation offered by the co-system such as lane keeping and/or adaptive cruise control. The current speed limit is displayed in the upper left of the Automation Monitor. It is under discussion to put the speed limit information on a secondary display or to integrate that information in the speedometer. In addition, lane departure warnings or blind spot warnings can be displayed on the Automation Monitor.

The Automation Monitor includes three main elements. Horizontal bars indicate the status of the *ACC* system. The vehicle icon stands for a detected target vehicle. The vertical bars indicate the status of the Lane Keeping System. The colour coding for this display elements was designed as follows:

- no bars/vehicle visible: no automation available,
- unfilled, white framed bars or vehicle: automation is available but not active or target vehicle in front detected but no is no automatic distance control,
- blue colour: automation is active and working.

The second display is the Automation Scale which is located on the right side of the interface. It follows the design scheme for the spectrum of automation levels. The levels of automation are shown as discrete levels and depicted on a vertical spectrum following the principle “up is more automation". The scale contains pictograms for the different levels of automation, and shows the driver the current automation level (highlighted with a corresponding colour). For *Driver Assisted* this colour is white, for *Semi Automated* this is light blue and for *Highly Automated* this is dark blue. All available automation levels are indicated with a white frame. Not available automation levels appear in dark grey.

A third component is dedicated for displaying warnings, messages and preconditions. The area is named Precondition and Message Windows. It is positioned beneath the Automation Monitor. Here, the information can be provided verbally as text and/or graphics. For this purpose, the display consists of two parts, a part for displaying pictograms and preconditions, and a part for displaying text messages. For warnings, the display can contain pictograms, for example a white or orange steering wheel indicating a take over request.

The generic design scheme for the display was used to design the different displays of the demonstrator vehicles. Figure 126 shows the final design of the displays after several iterative alignments during the HMI discussion.

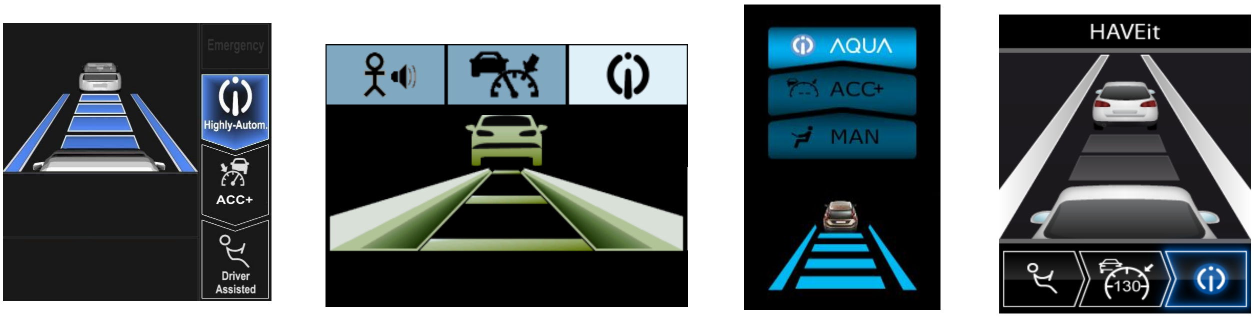


Figure 126: Final display design for the demonstrator vehicles (from left to right: Joint System, ARC from Continental, AQuA from VTEC and TAP from VW)

Switching devices for the Joint System

As the Joint System demonstrator is used for experimental comparison of different HMI-solutions, it was equipped with three different switching devices. First, there was a button solution with buttons for each automation level. The button of the current automation level was illuminated. Second, there was a stalk that is similar to the *ACC* stalk that can be found in current production cars. The functions of the stalk were extended to allow the switching to other levels than *Semi Automated*.

5.3.3 Joint System validation studies

During the course of the HAVEit project eight human factors studies were conducted by three partners. To use the time most efficiently, some of the studies were conducted in parallel. In the first study phase, general questions were addressed to validate the preliminary design of the Joint System (see Deliverable 33.3). Results of the findings by the four studies on the preliminary Joint System design were used to improve the Joint System design. With this improved design further studies were undertaken in the second study phase to address open design questions and to evaluate the final design (see Deliverable 33.6). Table 25 and Table 26 give an overview on the use cases and the main research questions addressed in the studies.

Study phase	Phase I				Phase II			
Use Case Class	Study VW "Automation levels"	Study DLR "Usability"	Study WIVW "Take-over request "	Study WIVW "Drowsiness & distraction"	WIVW "Distraction"	WIVW "Drowsiness"	VW " TAP transitions"	DLR "Comparison of prototype transitions"
Normal driving and activation of automation levels possible	x	x	x	x	x	x	x	x
Normal driving and activation of automation levels not possible	--	x	--	x	--	--	--	--
Driving and deactivation necessary (due to environment change)	x	x	x	--	x	--	--	--
Driving and driver drowsy	--	--	--	x	--	x	--	--
Driving and driver distracted	--	--	--	x	x	--	--	--
Driving, driver unresponsive and transition to MRM	--	x	--	--	--	--	--	--
Driving and detected obstacle	--	--	--	--	--	--	x	x
Driving and environment change undetected by co-system	--	--	x	--	--	--	--	--
Driving and speed limit change	--	--	--	--	--	--	--	x
Driving and lane change	--	x	x	--	--	--	x	x
Driving and right overtaking	--	--	--	--	--	--	x	--
Driving and adjacent road	--	--	--	--	--	--	--	--
Driving and technical failure	--	--	x	--	--	--	--	--
Misuse situations	--	x	--	--	--	--	--	--

Table 25: Overview of the HAVEit use cases that were addressed in the studies of phase I and II

Study phase	Phase I				Phase II			
Design Space: What shall the design be?	Study VW "Automation levels"	Study DLR "Usability"	Study WIVW "Take-over request "	Study WIVW "Drowsiness & distraction"	Study WIVW "Drowsiness"	Study WIVW "Distraction"	Study VW "TAP Transition"	DLR "Comparison of prototype transitions"
Which and how many automation levels?	x	--	--	--	--	--	--	--
How are the transitions?	x	x	--	--	--	--	x	x
What effects do previous experiences have on the drivers' expectations?	--	--	--	--	--	--	x	x
How good can the driver take over control?	x	x	x	--	x	x	x	x
How to bring the driver back in the loop?	--	--	--	x	x	x	--	--
How to keep the driver in the loop?	--	--	--	x	x	x	--	--
What configuration should the switching device have?	--	x	--	--	--	--	x	--

Table 26: Overview of the HAVEit research questions that were addressed in the studies of phase I and II

Studies in Phase 1: Testing the preliminary interaction design

To achieve a proper balance between width and depth and to address the most important Human Factors questions, four studies were conducted in the first phase of the project by three HAVEit partners.

- The first study of Volkswagen, conducted in the surrounding of HAVEit, tested especially the number and design of different automation levels and the transitions between these levels.
- The second study (DLR) explored the fundamental interaction schemes of Highly Automated driving in a usability assessment in which the preliminary design of the Joint System prototype (documented in D33.2) was extensively tested.
- The third study (WIVW) was specifically about the driver's reaction to take-over requests in case the Highly Automated driving can not longer cope with the driving situation.
- The fourth study (WIVW) addressed two specific use cases: *Driver drowsy* and *Driver distracted,* and presented the first evaluation results of the interaction design in those specific use cases.

The studies addressed different use cases and different design and evaluation questions to cover a wide range of issues in the first test phase (see Table 25 and Table 26). As the results of all studies are described in detail in HAVEit deliverable D33.3, here just a summary of the studies and their key findings are inlcuded.

Evaluation of transitions between four automation levels, integrating two different levels of highly automated driving

Main objective of the study

In this study the main focus was on user expectations as well as on user behaviour in transitional situations. These transitions occurred between four automation levels and were relevant from a technical and psychological point of view. The levels considered are shown in Figure 127.

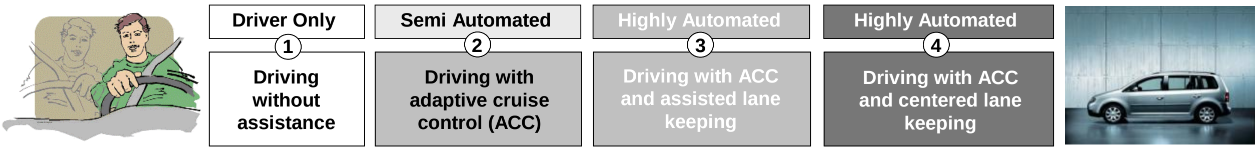


Figure 127: Four integrated automation levels that were tested in the study

The bandwidth of integrated driving functions encompasses the following stages: (1) Driving without assistance - (2) Driving with Adaptive Cruise Control - (3) Driving with combined lateral and longitudinal guidance, lane keeping only supported near the edges of the lane - (4) Driving with combined lateral and longitudinal guidance, lane keeping centred in the lane. For the individual modes, the following terminological simplifications were chosen in the study: (1) Driver Only (DO), (2) ACC, (3) Highly Automated Light (HA-L), (4) Highly Automated Strong (HA-S).

The area of automation considered was based on standard systems currently available on the market and on derivable potential integrations of functions. Furthermore, it was based on analyses of the need for support and on accident data (Staubach, 2009). In respect of the inserted transitions, not only previously considered transitions from the automation to the driver (e.g. Weinberger, 2001) were in the foreground but also transfers from the driver to the system. Moreover, as a consequence of the bandwidth of automation, the degree of control shifting between the automation and the driver varied. Direct deactivations and activations were thus supplemented with indirect operations via intermediate stages (e.g. ACC).

The aim of the study is to come up with recommendations for a user-centred design of bidirectional control transfer processes between selected system modes with defined function specifications. In the HAVEit use case catalogue, this would be the use case class “Normal driving, activation of automation levels possible”. Special attention is paid to the maintenance of mode awareness, the reduction of mode confusion and the identification of a degree of complexity that the driver can handle.

Method and participants

Only male employees of Volkswagen AG (n = 80) took part in the test. On average, they were 35.6 years old (sd = 6.2 years) and drove an annual distance of 18,887.50 km (sd = 9.123,62 km/year). Most of the participants had experience in the use of cruise control but not so much with ACC and hardly with any lane keeping functions. The 80 male subjects tested were given the instruction of using the presented automation stages during a relatively monotonous trip on a motorway. The background of the selected application context consisted on the one hand of studies demonstrating such a range of use of driver assistance systems. On the other hand, the driving environments on motorways are very structured and comparatively simple. From a technical point of view, that is why they can ensure a high level of system availability. As an analogy for future migration studies, the main focus was on testing a possible first area of use for driving functions with a higher degree of automation.

In the study 20 subjects were randomly assigned to each of the embedded stages: "DO", "ACC", "HA-L" or "HA-S". This level of automation was then always the starting point for triggered transitions to correspondingly higher or lower degrees of automation. Every test person was instructed to switch into these dedicated automation levels within the four integrated trips. The commanded transitions took place in approximately the final third section of the motorway route taken (Figure 128). The background to this was a question regarding effects of the implemented transitions after longer use of the respective driving functions. Due to the defined sequence of them, it can be seen that there were also trips without a transition (Figure 128, Trip 3). Transitions that were triggered by the driver or transitions that belonged to a take-over request initiated by the system therefore only occur three times in four trips (Figure 128). The trips among themselves lasted 15 minutes each.

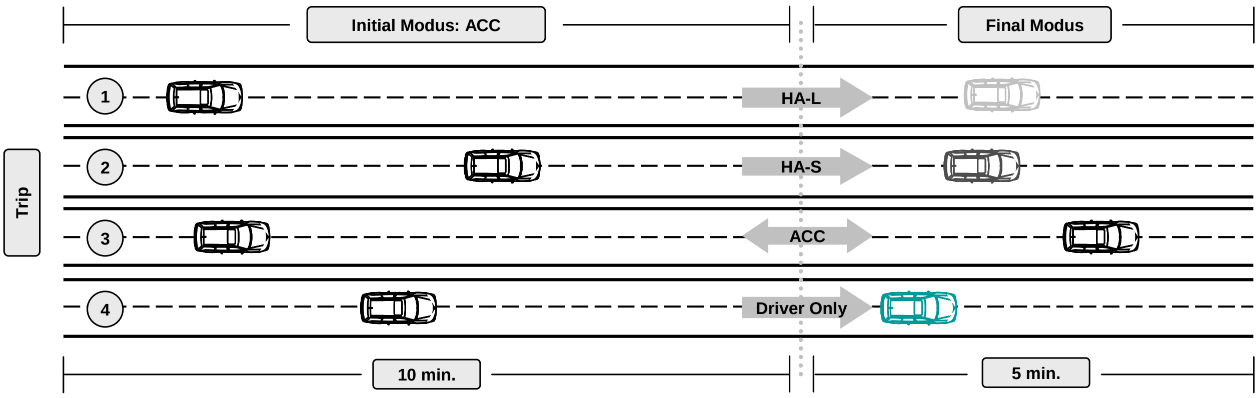


Figure 128: Sequence of four trips with the "ACC" starting mode as an example

A total of 240 transition situations were considered in the analyses. Apart from the sequence of transitions for each test group, the type of changeover was also randomly varied. This only concerned the "HA-L" and "HA-S" modes. Not only direct deactivations or activations but also indirect ones via intermediate stages (e.g. ACC) were possible. The transitions were triggered in relation to system relevant driving situations. For example, the "HA-S" mode could be activated when the lane markings were well visible and not covered. A display mounted on the right-hand side of the emulated combined instrument showed such changes of the system status to the driver in visual form. In addition, this was accompanied by an acoustic signal. The subjects had to activate driving functions offered by the system by pressing a button on the steering wheel. Referring to this, the participants received the instruction to use all modes and confirm all system offers with this one button. Also, they had to react accordingly when a take-over request in less automated modes occurred (HAVEit use case class "Driving and deactivation necessary (due to environment change)". A training period before each driving test provided them with help in handling the individual automation stages. In this phase the test persons also experienced situations for system activation and deactivation that were latent similar to the later scenarios during test drives.

In addition to a near-series implemented ACC system, a tolerance range was defined for the lane keeping variants. Within this range, the distance of a defined set-point from the ego vehicle was the main factor that determined whether corrective interventions in the steering wheel were carried out. In these cases the applied additional steering torque also varied in relation to the driving speed, which was a maximum of 110 km/h. For the "HA-S" mode, a lane-centred design of lane keeping was implemented and enabled the test subjects to take their hands off the steering wheel. Guidance with the "HA-L" system only took place in a supportive manner in which a counteracting steering force was initiated if the vehicle inadvertently came close to the edges of the lane.

Conclusions from this study

On the way from assistance to automation the results of this study show that the increase in complexity as well as the design of transitions needs to be considered. Mode confusion in respect of the “HA-L” and “HA-S” functions indicates that some drivers are mentally unable to clearly distinguish between these two modes. They overestimated the “HA-L” functionality and thought that it was the higher one of the “HA-S”. Due to the incorrect representation of function, the drivers removed their hands from the steering wheel. Moreover, after the perception that the system behaviour is not matching to the expectations of the driver, only a delayed search for information to verify the mental model took place.

Apart from such incorrect cognitive assumptions, it can also be observed that the system learning phase is characterised by rule-based and knowledge-based action. The knowledge that is to be formed and determined by degree of skill increases the amount of time needed for cognitive processes. With reference to this fact, we need simple systems with a small number of different automation stages that in addition complement each other. In order to actively promote mode awareness, it should be possible to differentiate the stages from each other. Not only by means of what is indicated to the driver but also by means of their specific control behaviour. Furthermore, in situations that are the same for the driver regarding the vehicle control (e.g. braking), the individual functions should also behave in a similar way (e.g. take-over request to “DO”). The steps for controlling the vehicle must also be designed similar for all levels of automation. A visual indication of the maximum available degree of automation, including the activation steps involved, could promote the build-up of mental systems representation.

Analogously to the knowledge gained regarding the stages investigated in this study, a function such as “HA-S” can be capable of direct activation and, vice versa, control can be handed back without intermediate stages. If the driver is still participating in the driving process as in the case of the “HA-L” mode, (de)activations via „ACC“ is undoubtedly conceivable. Important is that the transfer process from the automation system to the driver and vice versa is consistent within the automation spectrum. With higher levels of automation, the initiation of transitions needs to start earlier, so that the drivers have a chance to take over in time.

Usability assessment of the preliminary design for the HAVEit Joint System

Main objective of the study

The main objective of the study was to test the preliminary design for the HAVEit Joint System. For this, an intense usability assessment was conducted in a fixed-based driving simulator with non-professional users. The usability assessment focussed on finding out the most relevant usability problems that need to be improved in the design of the prototype. In addition, the study assessed what people expect from highly automated systems, how people use such highly automated vehicles and how they accept and like such systems. Three levels of automation, the level Driver Assisted, Semi Automated and Highly Automated and the transitions between these levels were tested for different use cases. Beside, two different switching devices were tested for switching between the different automation levels: a lever similar to the today’s ACC lever, and a button solution. The usability assessment resulted in first data about problems with the use of these two switching devices that builds the basis for improvement of these devices.

Method and participants

Eight participants (four male and four female) took part in the usability study. The participants were 25 to 62 years old. Their driving experience varied from 6 to 46 years of driving. Two participants used their car daily, three participants about three to five time a week, two used it one to two times a week and one participant used is less then once a week. The participants had different experience with driver assistance systems. The drivers were most familiar with the Route Guidance System, followed by the Park Distance Control. Half of the participants had experience with Cruise Control systems. Other systems that actively influence the driving tasks (like Lane Keeping Systems or Adaptive Cruise Control) were not or only little known by the participants.

During the study four of eight participants (two female and two male, aged 25-60 years) used the lever solution for switching between the automation levels, the other four (two female and two male, aged 24-62 years) used the buttons on the steering wheel.

The simulated test route in the DLR simulator was a German three-lane highway. The test route consisted of mainly straight sections with some slight curves. On the track there was a speed limit of 120km/h that was indicated by speed signs on the side of the road. There was little to medium traffic on the road.

The Joint System prototype to be tested consisted on three different automation levels that were selectable by the driver: Driver Assisted, Semi-Automated (ACC) and Highly Automated.

For the usability assessment the simulator was equipped with a force-feedback steering wheel and a force-feedback accelerator pedal. The force-feedback steering wheel and accelerator pedal allow applications of haptic signals, here forces and vibrations. The haptic signals were used to inform the driver about the control action of the automation and to warn him in critical situations.

A monitor (10,4 inch) was installed in front of the driver, replacing the traditional instrument cluster of a car. Here speedometer as well as three display elements defined for the HAVEit systems (see Figure 129) were displayed: the Automation Monitor, the Automation Mode Display and the Precondition and Message Windows.

Figure 129: Display element used for informing the driver about the current automation level

Two different kinds of switching devices were tested in this usability assessment: a lever similar to the ACC lever and a button solution fixed on the steering wheel.

There were different acoustic signals for informing and warning the driver. For information purposes there was a sound for a transition to higher levels of automation, a sound for transition to lower levels of automation and a sound for a refused transition used. For warnings there was a sound for the take over request that escalated in frequency and a sound for the Minimum Risk Manoeuvre (MRM).

Exploration procedure

The usability assessment had three main parts: The expectation assessment, the Naive Runs and the Trained Runs. The complete study took 3.5 to 4 hours per participant.

Part I, Expectation assessment: Before any contact with the prototype, the participants were asked about their expectation regarding highly automated vehicles and the reaction of highly automated vehicles in four different driving situations. Here, four of the use case classes were addressed: Normal driving and activation of automation level possible, Driving and lane change, Driving and detected obstacles and Technical failure. The expectation assessment was conducted in form of a semi-structured interview. The answers of the participants were recorded in written form and video-taped.

Part II, Naive Runs: After the expectation assessment, the participants were asked to test the software prototype of the highly automated vehicle in four so-called Naive Runs. These runs are called naive because the participants drove without any prior information about the system functions. In the Naive Runs the participants were asked to think aloud and vocalise everything that came into their mind about the interaction with the system.

Training: After the Naive Runs the participants conducted training. For this a written instruction similar to a vehicle manual was handed out (see Annex 3.3, D33.3). The participants were asked to read this manual carefully. Thereafter, the participants drove a training run that took ten minutes. In the first five minutes the participants were free to drive and test any function of the system as they liked. In the last five minutes the key interaction elements were trained by asking the participants to follow the instructions displayed on the screen. The interaction elements that were trained were the activation and deactivation of the automation levels via the switching device, strong braking and strong accelerating, the lane change manoeuvre, a situation in which the activation of Highly Automated is refused, the take-over request and the Minimum Risk Manoeuvre.

Part III, Trained Runs: After the training the participants drove the three above mentioned structured runs again. First they drove the run that was about normal driving and the activation and deactivation of the automation levels (Trained Run: Normal Driving and activation of automation levels possible). Then they drove the Trained Run: Transitions in specific situations and last the Trained Run: Driving and highly automated lane change.

After each run the participants were asked to fill in a questionnaire. After all three Trained Runs they were asked to give an overall evaluation of the prototype.

Overall evaluation of the highly automated vehicle and conclusions

At the end of the study the participants were asked about their overall evaluation of a highly automated vehicle (more precisely: the preliminary Joint System Design) with different automation levels similar to this that they had driven in the study (Table 27).

The participants rated this kind of vehicle as rather to very good and as quite to very useful. Five of eight participants felt that this vehicle is quite to very comfortable, two rated it as neutral. One participant who felt that there were too many automation levels rated it as rather uncomfortable.

When asked about the advantages and disadvantages of such a technique for highly automated driving the participants most often mentioned a more comfortable, relaxed driving and improved safety. The disadvantages that they mentioned were: an overreliance on the automation that could lead to critical situations in case of failure, a loss of feeling for the driving task and the driving environment, a reduced attention level, a misuse of the technique and higher effort than in normal car to handle the automation and different automation levels.

	very	quite	rather	neutral	rather	quite	very		mean (SD)
bad					1	3	4	**good**	2.25 (0.89)
useless						6	2	**useful**	2.13 (0.64)
uncomfortable			1	2		2	3	**comfortable**	1.5 (1.60)

Table 27: Distribution of the answers of the eight drivers regarding the questions: How do you evaluate such a highly automated vehicle that you tested?

All in all, when asked if they would you like to use such a highly automated vehicle the prototype got a positive rating (Table 28). Seven of eight drivers of this study said that they are quite or very willing to use this kind of vehicle. However, one participant who had a sceptical view towards automating the driving task said he would be rather not willing to use this kind of vehicle.

	very	quite	rather	neutral	rather	quite	very		mean (SD)
disliked			1			3	4	**liked**	2.13 (1.36)

Table 28: Distribution of the answers of the eight drivers regarding the question: How much would you like to drive such a highly automated vehicle?

Summing up, the usability assessment showed that besides some usability issues described below, all drivers could drive the prototype of the highly automated vehicle quite well. During the Naive Runs the drivers were already able to drive the vehicle without any prior instruction and evaluated the prototype quite well. After reading the manual and a short Training Run the performance improved for all tested use cases and some of the usability problems that were recorded in the Naive Runs vanished.

Regarding the use case class *Normal driving and activation of automation levels possible* the most obvious usability problem occurred with the lever. Here, drivers switched off the automation unintentionally when trying to activate the level Driver Assisted. This issue needs some improvements of the lever configuration, e.g. a solution might be that the OFF detent of the lever is blocked and the OFF button placed somewhere else on the dashboard. In addition, the configuration of the acceleration and brake pedal needs to be adjusted to allow an easier transition by the pedals.

Regarding the use case class *Take over request* the results showed that seven of eight drivers reacted correctly in the Naive Run by putting their hands back on the steering wheel when driving hands-off before. However, it took them relatively long to realize that an additional confirmation by switching device, accelerator, or brake pedal is needed to get the control back. While the scheme of interlocked transitions holds up in most cases, in this specific case the design might have been too conservative, or not sophisticated enough. A possible solution to be confident that the driver is back in the loop and takes over control might be to use the grip force the driver gives on the wheel or the camera data of the drivers head orientation. This might allow a more intuitive and easier handling of the take-over situations ensuring an adequate safety level for the transition of control back to the driver.

Regarding the use case class *Activation not possible* no serious problems were detected. All drivers were able to recognize that the requested transition to a higher automation level was not successful and could even predict this after the training. The precondition symbols were not self-explanatory but the drivers said that they gave good hints for unfulfilled preconditions after the training.

Regarding the use case class *Lane change* some improvements of the interaction design and the prototype seem to be needed. The drivers had problems to find out how the highly automated lane change should be triggered. Some of them searched for a “Yes” button to confirm the question on the display. Here, some modification of the display design could help to improve the intuitiveness. In addition, the haptic feedback on the wheel during the lane change and the force needed to initiate the lane change was not that smooth as desired. Thus, the drivers rated the lane change as not that comfortable and safe. However, the reported problems did not hinder the drivers to change lanes. Even though not all drivers really understand in the Naive Run that the lane change was conducted highly automated, the driver wishes were not blocked by the prototype, and they changed the lanes by using the indicator and steering as they are used from normal driving.,

Regarding the interface components the drivers gave an overall positive rating. The rating for the switching devices was good, even though real improvements need to be made regarding the OFF detent of the lever. The visual feedback and haptic feedback received positive ratings. Some changes could be made regarding the acoustic feedback. Here, the subjective rating differed between the drivers according to their personal liking.

During the drives some signs of mode confusion were also recorded. One driver took her hands away while driving Driver Assisted. Here, one solution would be to use the hands-on check which should detect this situation and remind the driver to keep his hands on the wheel. Two others reported a mismatch regarding the current automation level and the actions that they would expect.

Overall, the technique of highly automated driving (thus the preliminary Joint System design) was evaluated as a positive technique for allowing safe and relaxed driving. When asked if the drivers would like to drive such a vehicle seven of eight drivers would quite or very much like to use it.

Driver’s reactions to take-over requests when driving Highly Automated

Main objective of the study

Besides the usability studies on automation levels, transitions and HMI, an experiment on transitions was conducted in the motion base simulator at WIVW. Therefore, a set of scenarios which required transitions between levels of automation was implemented in the driving simulation. The goal was to define some basic data on how drivers normally react to such transitions in terms of reaction times (how fast do drivers react?) and quality of reactions (does the transition itself lead to any erroneous or critical driving behaviour?). Furthermore, the experiment should give first hints how drivers evaluate highly automated driving in general and how it affects their perceived workload. The design of the first generic interface draft could be also tested. In this study the set of automation levels was restricted to either Driver Only (“manual”) or Highly Automated driving.

Method

16 test drivers participated in the study. The mean age of participants was 32.8 years (min= 24 years; max=49 years; SD= 8.0 years) with 8 male and 8 female drivers. The test drivers were selected from a test driver panel of the WIVW which includes test persons who have completed at least 3 hours of practice in the driving simulation and have mostly participated in several other driving simulator studies before. This procedure ensures that all participants in the study were well trained in simulator driving.

For this experiment a simplified co-system was integrated in the WIVW simulation which allowed the driver to drive on a Highly Automated level. It consisted of a combination of lateral and longitudinal control of the vehicle, comparable to Adaptive Cruise Control system (ACC) plus a lane keeping system that is centred in lane. It allowed setting a target speed up to 130 km/h which was constantly kept by the system. Additionally, the system followed the lead vehicle with a predefined distance automatically. In addition, the system kept the vehicle at the centre of the lane so that no steering effort was required. This allowed the driver to drive hands-off. Precondition for activating the system in this study was the presence of a target vehicle detected within the sensor range.

If the system was not able to manage a specific situation (e.g. if it was not able to keep a safe distance to the lead vehicle), a take-over request (TOR) was given on which the driver had to react. After the driver had put his hands back to the steering wheel, the automation level was switched to Driver Only. The HMI used in this study was designed according to a first proposal described in Deliverable D33.1.

The test course took 50 minutes on a two-lane motorway with mainly straight sections and some broad curves. Most of the time, the driver had to follow a lead vehicle within various speed ranges (from 40 km/h minimum in the traffic jam scenario up to 120 km/h in car follow situations). The system could be run on Highly Automated level in these scenarios. These scenarios were representatives for the use case class *Normal driving and activation possible*. In between, additional scenarios were integrated in which the environment or traffic conditions changed in a way the system could not manage on Highly Automated level. Those scenarios belonged mainly to the use case class *driving and deactivation necessary*. They required either a system or a driver-initiated transition back to Driver Only. After the transitions the driver could either directly reactivate the system if all preconditions were fulfilled again. Otherwise a section of free driving without any lead vehicle followed (valid speed: 150 km/h) until the former vehicle was reached again or a new target vehicle appeared. The situations that required such a take-over were:

- The sensor had problems due to missing lane markings
- The target vehicle braked heavily
- The target vehicle left highway
- The target vehicle left sensor range by accelerating
- A lane change had to be executed manually
- A hidden obstacle was standing on the lane
- A boar crossed the road (use case: *driving and environment change undetected by co-system*)
- The system produced a steering failure (use case: *driving and technical failure*)

The participants were instructed to drive always on the right lane of the highway, to follow the lead vehicle and not to overtake. In order to test the effects of high automation levels and the required transitions at system limits, two driving conditions were compared: One group of eight drivers drove the test course on Highly Automated level. The other group of eight drivers completed the test course on the Driver Only level (manual driving condition).

After an explanation and practice of the functionalities and the use of the co-system the drivers performed the test track. The drivers of the Highly Automated group were instructed to use the system as intensive as possible and to only deactivate it when the situation required it. The drivers of the manual driving condition were simply instructed to drive along the highway on the right lane, to follow the lead vehicle in an adequate and constant distance and not to overtake. Both objective parameters as well as subjective parameters were collected from the drivers. The most important objective parameters analyzed were:

- Proportion of time drivers drove hands-off
- Lane and speed keeping performance in phases of manual driving
- Driver's reaction time to take-over requests (TOR, time between onset of TOR and hands-on, time between onset of TOR and pedal usage)
- Driving performance in take-over situations (occurrence of critical steering activity or critical distances to the target vehicle?)

Furthermore, a questionnaire was developed in order to assess the subjective evaluation of the system design, the interface and the acceptance of the system.

Conclusions

To sum up the results of this study (please see D33.3 for detailed description of results), drivers seemed to have no essential problems in the use case *Driving and deactivation necessary (due to environment change*). They reacted to required transitions of a Highly Automated driving back to Driver Only in most of the cases appropriately. Especially fast reactions were possible if the system limits were known and could be predicted.

More difficult was the occurrence of sudden events which were not detected and therefore not manageable by the system (example situation boar; belonging to use case class *Driving and environment change undetected by co-system*). In those situations, the drivers reacted later compared to manual driving. In addition, it seemed that the variation in driver's behaviour was reduced when driving Highly Automated. Compared to the manual condition, drivers more often showed only a reaction in longitudinal direction (braking manoeuvre) without also regarding the lateral control of the vehicle (e.g. additional avoidance manoeuvre). This might be also explainable by the higher time urgency with which the drivers have to react. However, no critical situations occurred due to the take-over itself. It could not be observed that any of the drivers was surprised by the take-over and snatched the steering wheel or comparable reactions. Also, driving performance after a transition back to Driver Only seemed not to be negatively influenced compared to fully manual driving.

In the conducted study in the driving simulation, a high system trust could be observed. After the system was activated, drivers took off their hands and feet very fast and drove hands-free nearly all the time.

The system with Highly Automated driving function was rated rather positive. It was evaluated as comfortable, easy to use and easy to understand. Feedback on current system state and system-initiated transitions were comprehensible. Drivers rated the acoustic feedback more necessary as the visual feedback (which was even not recognized by some drivers). Especially the display of the required preconditions for system activation was seen as very favourable. However, drivers complained that they still have to be attentive when driving with high automation levels due to the limits of the system.

The results of this experiment on transitions have some restrictions: only 2 automation levels and transitions between them were considered: Driver only vs. Highly Automated driving. This is a simple system design where no mode confusion occurred. More problems can be expected when the range of automation levels is increased to levels of Assisted and Semi-Automated driving in between Driver Only and Highly Automated. In further studies, it was tested if the driver still knows what he is responsible for when more complex transitions between automation levels are possible.

Evaluation of initial strategies for the use cases "driver drowsy" and driver distracted"

"Driver drowsy"

Objective of the studies

The main objective of the following studies conducted by WIVW was the validation of the DSA software component that was developed in order to assess driver's state, namely the driver's drowsiness level and distraction level. The component uses a combination of direct measures (outputs of a camera detecting the eyelid and head movements of the driver) and indirect measures (observation of driver's behaviour and performance in the interaction with the primary driving task or additional secondary tasks). A first version of this software is described in Deliverable D32.2. For a technical and empirical validation it was implemented in the WIVW driving simulation. Detailed results of these validation studies can be found in deliverable D32.2. Furthermore, based on the proposals made in deliverable D33.2 a first evaluation of intervention strategies for the use cases *Driving and driver drowsy* and *Driving and driver distracted* was conducted. These results are summarized in the following chapters. Please note that the presented results reflect an intermediate step in the design process. They should be used to identify the relevant factors that must be addressed for this special use cases in order to guarantee that the driver will accept such intervention strategies and that safety benefits will occur.

Method

The empirical tests were conducted with 12 test drivers in the WIVW driving simulator. They were randomly assigned to 2 experimental groups: 6 drivers drove the test course on the Driver Assisted level of the HAVEit system (DA group). This group was slightly assisted by a lane keeping departure warning system which only intervened in critical situations when the driver tended to get off the road by providing a slight force on the steering wheel towards the opposite direction. The other 6 drivers drove on the Highly Automated level of the HAVEit system (HA-group). On this level, lateral and longitudinal control was performed automatically by the system up to a speed of 130 km/h. The driver could drive hands-off. The presence of a target vehicle was not mandatory for driving Highly Automated in this system design. In case of a situation the system could not manage (e.g. too large relative speeds between two vehicles) a take-over request was given. The driver had then to take-over the steering wheel and the pedals and a transition towards the Driver Assisted level was executed.

The driving task consisted of driving on a 2-lane motorway with mainly straight and slightly curvy sections. The track contained free driving scenarios (without any lead vehicle), car following scenarios with a lead vehicle driving around 110 km/h, traffic jams, compelled lane changes at road works (requiring a transition to Driver Assisted) and heavy braking manoeuvres of the lead vehicle in random intervals (resulting in a take-over request when driving on Highly Automated level). The drivers were instructed to always stay on the right lane even when the lead vehicle drives more slowly than allowed, stick to the valid speed limit and only execute a lane change when they are forced to at road works. Furthermore, driving in night time was simulated by a very realistic illumination of the surrounding environment and traffic (see Figure 130). The drivers started the drive after lunch at about 1.30 pm and after having performed another 2 hours lasting study with a distracting task (reported in section **Fehler! Verweisquelle konnte nicht gefunden werden.**). The duration of the drive was determined by the time it took the drivers to get really drowsy or even sleepy. Due to the very realistic night-time environment, the time of day and the heavy load of the previous study for most of the drivers it took not longer than 2.5 hours. For the low minority of drivers who showed no tendency to really fall asleep the drive was stopped after 2.5 hours.

Figure 130: The HMI of the HAVEit system showing the state “Highly Automated level” (left) and one of the night-time scenarios “traffic jam” (right) in the validation study.

For a first test of suitable interventions in case of drowsiness an escalation strategy was implemented at the very end of the drive when most drivers were already drowsy or even sleepy. The question was if this strategy would be acceptable by the drivers and how it affects their alertness level. The first step of this strategy was to display a drowsiness information message which consisted of a blinking coffee cup and the text message “please take a break”. After 2 additional minutes the second step of the escalation strategy was triggered: A drowsiness warning was given which means that the information message was accompanied by an acoustic sound to alert the driver. In the third step after 2 more minutes the DA group received another drowsiness warning while the HA group received a take-over request which forced them to take over the driving task again and drive on in the Driver Assisted level till the drive was stopped. In the later application, of course, the escalation strategy will be not a fixed sequence without taking into account the driver’s reaction as it was the case in this study. Instead it will be based upon the outputs of the DSA module and will be stopped as soon as the driver stops for a break or is becoming more alert again. As the main focus of this validation study was on the observation of the natural drowsiness development and the detection quality by the software component, it was decided to not disturb this process but to explore the strategy at the very end of the drive.

Results and conclusion

Overall, the idea of an Attention Monitor that observes the driver’s drowsiness level and that gives him respective feedback was judged very positively and useful. However, it should be noted, that the system itself will be not able to make the driver more alert again. Key intention of the system is to persuade the driver to take a break and that it is in the responsibility of the driver to follow this recommendation. The present study did not test how a subsequent break would affect driver’s state. The HMI itself might be improved: While the acoustic warning was rated very positive, especially drivers of the Driver Assisted group argued that they have not noticed the mere visual information. An increase of the symbol size and the colour contrast might be helpful here.

What has to be further analysed is how the interventions are accepted if they are really triggered by the current driver state. An important decision is when to start the escalation. If it is started too early a low acceptance will result, especially as it can be expected that the number of false alarms will be higher on lower levels of drowsiness. If it is triggered too late, negative consequences for driving safety will be the result, especially when the driver is forced to take over the driving task in a state where he is no longer able to perform the driving task safely. These questions will be answered if it is clearly analyzed how good the detection rate by the Driver State Assessment component will be.

“Driver distracted”

Method

The study was conducted with the same 12 drivers of the first study in the WIVW driving simulator. Both studies took place at one day (study about distraction in the morning from 11 a.m. to 1 p.m. including training trials for practicing the use of the HAVEit system and the additional performance of the secondary task, study about drowsiness from 1.30 p.m. to approx. 4 p.m.) The test drivers were randomly assigned to 2 experimental groups: 6 drivers drove the test course on the Driver Assisted level of the HAVEit system (DA group). This means that they were slightly assisted by a lane departure warning system which only intervened in critical situations when the driver tended to get off the road by providing a slight force on the steering wheel towards the opposite direction. The other 6 drivers drove on the Highly Automated level of the HAVEit system. On this level, lateral and longitudinal control was performed automatically by the system up to a speed of 130 km/h. The driver could drive hands-off. In case of a situation the system could not manage (e.g. too large relative speeds between two vehicles) a take-over request was given. The driver then had to take-over the steering wheel and the pedals and a transition towards the Driver Assisted level was executed.

The driving task consisted of driving in a 2-lane motorway of about 20 min length which was repeated 3 times in 3 experimental conditions and with varying sequence of the driving scenarios. The valid speed limit was set to 120 km/h over the whole track. It contained free driving scenarios without any lead vehicle, car follow scenarios with a lead vehicle driving around 110 km/h, traffic jams, compelled lane changes at road works and one heavy braking manoeuvre of the lead vehicle (resulting in a take-over request when driving on Highly Automated level). The drivers were instructed to always stay on the lane even when the lead vehicle drives slower than the given limit, stick to the valid speed limit and only execute lane changes when they are forced to at road works.

In addition, the drivers were instructed to perform a secondary task while driving. It was a hierarchical menu navigation task comparable to a modern in-vehicle information system that can be used for several functionalities (e.g. navigation system, vehicle data, entertainment functions and telephone) by one single display and one single controller. The menu system was presented on a visual display at a lower position in the central console (ca. 34° to the right; ca. 23° down depending on the driver’s seat position). To navigate within the menu a commercially available joystick was used. The driver was instructed to navigate to a specific menu function (e.g. control average fuel consumption). The task was completely self-paced and interruptible. As soon as the driver confirmed the correct option, a new task could be started. Figure 131 shows an extract from the menu system and the positioning of the secondary task inside the vehicle. The drivers were instructed to prioritize the primary driving task and to perform the menu task only when the situation allowed it.

For a first test of suitable interventions in case of distraction a so called Attention Monitor was implemented. The interventions of the Attention Monitor are based on the calculation of a continuous distraction score that depends on the type of the distraction and the time the driver is engaged in the secondary task (for a detailed description of the algorithm see deliverable D32.1). As there were problems to reliably detect visual distraction by the camera (looking on/off the road) in the present study only the driver’s joystick movements were used as inputs for the computation of the distraction score. As soon as a certain threshold was exceeded the Attention Monitor started the escalation and gave respective feedback to the driver. If the driver exceeded a certain threshold of detected distraction, first, an attention request as mere information was given by the Attention Monitor. In the display there was a flashing orange icon of an open eye and the text message “Attention, please!” If the driver was detected as attentive again, the escalation stopped at this stage.


Figure 131: Display of secondary task presentation in the central console with joystick (left) and contents of the hierarchical menu (right).

If the driver did not react, the Attention Monitor started an attention warning. In the present study this was initiated when the driver was still or again above the defined threshold after 30 sec. In the display there was again a flashing orange icon of an open eye and the text message “Attention, please”. This was combined with a warning sound. If the driver reacted, the escalation stopped at this stage.

If the driver was driving on Highly Automated level and he/she did not react, the next escalation step was a take-over request for giving the control back to the driver. The criterion for reaching this level was that the driver was still or again above the defined threshold after 30 sec. The system then switched to the automation level Driver Assisted. In the current version of the software the escalation was stopped here. For later applications the next step if the driver does not react to the take over request and is still distracted would be a Minimum Risk Manoeuvre to bring the vehicle to a Minimum Risk State.

In the present study two variants of the Attention Monitor were implemented:

- AM 10: Attention Monitor with a threshold of distraction score = 10 (meaning after 5 sec uninterrupted menu navigation)
- AM 20: Attention Monitor with a threshold of distraction score = 20 (meaning after 10 sec uninterrupted menu navigation)

In order to evaluate the effects of the Attention Monitor a control condition without any interventions was introduced as a baseline. Each driver performed 3 drives of 20 minutes with each of the 3 AM variants in fully counter-balanced sequence. After each of the drives the drivers filled out an online questionnaire. After having finished all the loops the drivers answered a final questionnaire and were interviewed by the experimenter on specific aspects of the AM.

Please note that the here investigated version of the Attention Monitor was very conservative. At the moment this version was only based on the time the driver is engaged in a secondary task without having the information on the visual distraction that is involved. It currently also did not take into account the driving situation (e.g. whether the driver approaches a critical situation). Furthermore, both for Driver Assisted and Highly Automated driving the same thresholds for interventions were implemented. The following results reveal first hints which modifications will be necessary in order to achieve an acceptable solution.

Results and conclusions

From these preliminary results a number of suggestions for an improvement of the Attention Monitor and the underlying algorithms for distraction detection can be made:

- The algorithms inevitably need information about the visual distraction of the driver in order to define the distraction level
- The noticeability of the information message must be improved by increasing the size of the symbols and their contrast. As an alternative one can even think about skipping the information message and directly combine it with an acoustic sound. This makes sense as especially when the driver is visually distracted the visual channel might be overloaded.
- The thresholds for interventions need to be adapted to the automation level in order to reach an acceptance of the drivers when driving Highly Automated. Depending on the technical reliability one can think of a less conservative distraction level that is allowed.
- The thresholds for interventions should be carefully balanced out - if the thresholds are set too low this might not only lead to a decrease in acceptance but also to an additional load for the driver.
- The thresholds for interventions should take into account the current driving situation, e.g. the thresholds must be set higher in situations with low demands (low traffic density or low speed) and set lower in situations with higher demands. Another possibility would be that the Attention Monitor only intervenes if additionally a critical driving situation is detected by the system and the driver is not looking at the road in that specific moment. However, it has to be noted, that this requires that the situation can be detected in advance. If the critical situation already has occurred an additional warning might lead to even higher load. The better solution in such immediate situations is to directly intervene, e.g. by executing an emergency braking.

Studies in Phase 2: Testing the improved interaction design

Results of the findings by the four studies on the preliminary Joint System design presented above used to improve the Joint System design. With this improved design four further studies were undertaken testing the improvement and answering last questions on the transition design:

- The first study in this phase concerned the evaluation of the improved HMI design and the improved algorithms of the Attention Monitor for the use case “Driver drowsy in a simulator study at WIVW”.
- The second study was also conducted by WIVW testing the improved design for the use case “Driver distracted”.
- The third study concerned the evaluation of two different transition designs for the Temporary Auto-Pilot (TAP) and was undertaken by VW.
- The fourth study focused on the comparison of driver initiated transitions for four HAVEit transition schemes/prototypes. This study was done by DLR.

The studies, in particular details on the results achieved, are described in deliverable D33.6. To provide a complete view on the Joint System design process, in this document just a brief summary is provided. Among others, these studies led to the transitions summarized in Table 20 to Table 24. Finally, from these additional studies conclusions for the final version of the joint system were drawn (see subsection 5.3.4). The final design was then implemented in the HAVEit demonstrators and presented during the final event.

Evaluation of improved strategies for the HAVEit use cases "driver drowsy" and "driver distracted"

"Driver Drowsy"

Main objective of the study

The main objective of the following studies conducted by WIVW was the validation of the DSA software component that was developed in order to assess driver's state, namely the driver's drowsiness level and distraction level. The component uses a combination of direct measures (outputs of a camera detecting the eyelid and head movements of the driver) and indirect measures (observation of driver's behaviour and performance in the interaction with the primary driving task or additional secondary tasks). See also sections 5.2.5.2 and 5.2.5.3 above and D33.3 for details on the first study on the DSA component.

In deliverable D33.2 first proposals were made how the interaction strategy for the use cases "driving and driver drowsy" and "driving and driver distracted" could look like and tested in first studies (see D33.3 and studies reported above). During the course of the project a detailed interaction strategy was worked out and was implemented as a so called Attention Monitor (AM) system in the WIVW driving simulator. The following chapter describes the evaluation study for this Attention Monitor in case of drowsiness that was conducted in the driving simulator. The goal was to assess how it affects the driver's state and also how the improved interaction strategy and the used HMI are accepted by the drivers.

Method

The 12 drivers were randomly assigned to 2 experimental groups: 6 drivers drove the test course on the Driver Assisted level of the HAVEit system (DA-group). In this study, this means that they were slightly assisted by a lane departure warning system which only intervened in critical situations when the driver tended to get off the road by providing a slight force on the steering wheel towards the opposite direction. The other 6 drivers drove on the Highly Automated level of the HAVEit system (HA-group). On this level, lateral and longitudinal control was performed automatically by the system up to a speed of 130 km/h. The driver could drive hands-off. The presence of a target vehicle was not mandatory for driving highly automated in this system design. In case of a situation the system could not manage (e.g. too large relative speeds between two vehicles) a take-over request (TOR) was given. The driver had then to take-over the steering wheel and the pedals and a transition towards the Driver Assisted level was executed. The automation level Semi-Automated (driving with Adaptive Cruise Control) was also implemented in the HAVEit system but not separately investigated in this study.

The driving task consisted of driving on a 2-lane motorway with mainly straight and slightly curvy sections. The track contained free driving scenarios (without any lead vehicle), car following scenarios with a lead vehicle driving around 110 km/h, traffic jams, compelled lane changes at road works (requiring a transition to Driver Assisted when driving on Highly Automated level) and heavy braking manoeuvres of the lead vehicle in random intervals (resulting in a take-over request when driving on Highly Automated level). The drivers were instructed to always stay on the right lane even when the lead vehicle drives more slowly than allowed, stick to the valid speed limit and only execute a lane change when they are forced to at road works. Furthermore, driving at night time was simulated by a very realistic illumination of the surrounding environment and traffic. The drivers started the drive after lunch at about 1.30 pm and after having performed another 2 hours lasting study with a distracting task. The duration of the drive was determined by the time it took the drivers to get really drowsy or even sleepy. Due to the very realistic night-time environment, the time of day and the heavy load of the previous study for most of the drivers it took not longer than 2.5 hours. For the low minority of drivers who showed no tendency to really fall asleep the drive was stopped latest after 2.5 hours.

Interaction design of the Attention Monitor

An Attention Monitor (AM) system was implemented in the simulation which triggered different escalation steps and the associated HMI messages dependent on the drowsiness outputs of the DSA component. The general principles of this Attention Monitor for the use case “driver drowsy” can the described as the following:

The first escalation step and with it the information message is triggered if the driver is detected twice as “slightly drowsy” within a timeframe of 5 minutes- this should decrease the number of false alarms of the AM. If the driver is again detected as “slightly drowsy” within the next 15 minutes, the information message is repeated.

The second escalation step and with it the warning message is triggered if the driver is detected as “drowsy” or as “sleepy” the first time. The warning message is repeated if the driver stays “drowsy” (then every 15 minutes) or even gets “sleepy” (then immediately) and is driving in Driver Assisted mode.

The third escalation step consisting of a take-over request and a transition to Driver Assisted is triggered if the driver is driving in Highly Automated mode and is detected as still “drowsy” (then after 15 minutes) or even “sleepy” (then immediately) but only if he had received a warning on escalation step 2 before.

The fourth escalation step “Minimum Risk State” in case of a driver not reacting to the take-over request is not implemented in that version of the AM.

The time between messages of the system (either information or warning) is minimum 15 minutes. An exception is if the driver is detected as “sleepy”: This directly triggers an intervention of the AM (either an immediate warning when driving Assisted or an immediate take-over request when driving Highly Automated and already having received a warning before).

A dead time of 5 minutes after each message is considered where the driver is able to reach a lower level of drowsiness before the DSA output is observed again (e.g. to reach a better lane keeping performance).

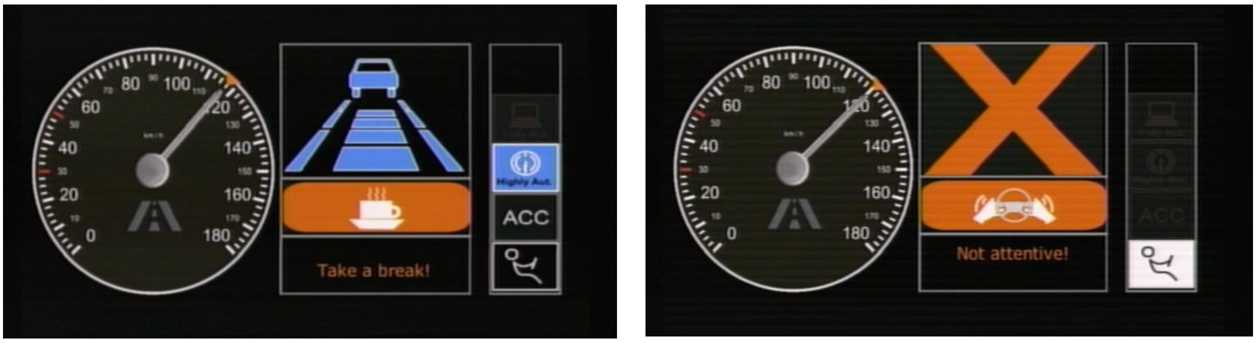


Figure 132: Drowsiness information and warning message (left) and take-over request for the HA group as final step of the escalation strategy (right)

Test procedure and measures

After having finished the calibration loop in the Driver Assisted mode, the drivers drove the subsequent test course according to their experimental condition either in the Driver Assisted mode of the HAVEit system or in the Highly Automated mode. Messages by the Attention Monitor were triggered online by the outputs of the DSA component. For the evaluation of the intervention strategy several subjective ratings were collected from the drivers, some of them online during the drive, others as a questionnaire after the drive.

Results and conclusions

To sum up the results, the driving simulator environment turned out to be a very good tool to study the development of drowsiness and possible countermeasures. It is able to successfully induce even high levels of drowsiness in a comparably short time. No differences in the average drowsiness development between the Driver Assisted and the Highly Automated group could be found. This result has already been revealed in the first validation study. However, the individual drowsiness levels seem to more strongly oscillate in the HA group maybe due to short-term recoveries in drowsiness during phases of assisted driving.

The DSA component quite successfully detected several stages of drowsiness. However, the false alarms produced by the DSA affected the evaluation of the AM interventions, especially those where the drivers received a warning or take-over request although they were awake. As false alarms can not be avoided, for heavy cases a solution has to be found in the later application (e.g. the possibility to switch off the Attention Monitor). Altogether, the AM produced only few false alarms in the present study. In 88% of the cases the interventions were correct. This also had an effect on the global evaluation of the Attention Monitor. It was perceived as very meaningful and useful, the HMI was very good to understand and the interventions seemed to have a positive effect on driver's perceived drowsiness level. Even the transition to Driver Assisted was evaluated as a meaningful and helpful tool for a drowsy driver and was not seen as domination by the system.

The adequateness of the single interventions turned out to be dependent from the initial drowsiness level at the moment of the intervention (with high ratings at the “drowsy” and the “sleepy” level and lower ratings at “slightly drowsy” level), the type of intervention (with higher ratings for warning messages and take-over requests (TOR) and lower ratings for information messages) and the time-on-task (with higher acceptance of interventions after 40 minutes driving). For the realization of the interaction strategy in the demonstrator vehicles based on these results, it can proposed to skip the information message on the “slightly drowsy” level or to directly combine this message with an acoustic warning. Furthermore, a certain driving duration should be considered that has to be elapsed before starting with AM interventions.

The short-term effects of the different AM interventions on the subjectively perceived drowsiness level are in general very low. On average, drivers felt less than one KSS-point improvement by the messages. This is an indicator that the messages themselves will not really help the drivers to get awake again. They should only serve for persuading the drivers to take a break at the next rest area. Also the take-over request and the subsequent 15 minutes drive on Driver Assisted (DA) level affects only partly the drowsiness perception by the drivers.

While some drivers seemed to benefit at least short-term from the take-over request or even longer from the DA drive, the others did not experience any changes in drowsiness or they perceived even more drowsiness. On average, the TOR and the subsequent DA drive seem to have only a slight and short-term effect on driver drowsiness that diminishes very fast over time. Also the observation of objective indicators for drowsiness occurring within the DA drive reveal that the TOR only affects a subgroup of drivers and also those only for short times. Therefore, it has to be assured that the driver interrupts his drive as soon as possible after a TOR and takes a break.

“Driver Distracted”

Main objective of the study

A further simulator study was done targeting the evaluation of the improved interaction strategy of the MSU (Mode Selection and Arbitration Unit) in case of distraction. The inputs for the interactions with the driver are the distraction outputs of the DSA component. It was investigated how effective the strategy of the Attention Monitor is and also how the improved HMI is accepted by the drivers. The results were used as recommendations for the final improvement of the MSU.

Method

The same 12 participants were included in this study as in the drowsiness study. The study about distraction took place in the morning from 11 a.m. to 1 p.m. including training trials for practicing the use of the HAVEit system and the additional performance of the secondary task, the study about drowsiness was conducted from 1.30 p.m. to approx. 4 p.m.

The driving task consisted of driving on a 2-lane motorway of about 20 min length which was repeated 3 times in 3 experimental conditions and with varying order of the driving scenarios. The valid speed limit was set to 120 km/h over the whole track. It contained free driving scenarios without any lead vehicle, car follow scenarios with a lead vehicle driving around 110 km/h, traffic jams, compelled lane changes at road works and one heavy braking manoeuvre of the lead vehicle (resulting in a take-over request when driving on Highly Automated level). The drivers were instructed to always stay on the right lane even when the lead vehicle drives slower than the given limit, stick to the valid speed limit and only execute lane changes when they are forced to at road works.

In addition, the drivers were instructed to perform a secondary task while driving. It was a hierarchical menu navigation task comparable to a modern in-vehicle information system that can be used for several functionalities (e.g. navigation system, vehicle data, entertainment functions and telephone) by one single display and one single controller. The menu system was presented on a visual display at a lower position in the central console (approx. 34° to the right; approx. 23° down depending on the driver's seat position).

To navigate within the menu a commercially available joystick was used. The driver was instructed to navigate to a specific menu function (e.g. "control average fuel consumption"). The task was completely self-paced and interruptible. As soon as the driver confirmed the correct option, a new task could be started. The drivers were instructed to prioritize the primary driving task and to perform the menu task only if the situation allowed it and when they were keen on doing it.

The interventions of the Attention Monitor were based on the DSA distraction output. This output is based on the calculation of a continuous distraction score that depends on the type of the distraction and the time the driver is engaged in the secondary task. The thresholds for what is being defined as unacceptable distraction is dependent from the current driving situation and also the automation level the driver is driving in.

For the present study the two inputs for the computation of the distraction score were the time the driver looks away from the road and the time the driver is making inputs on the joystick of the menu navigation task without any meaningful interruptions. This means that even if the driver looks to the road but is cognitively engaged in a secondary task the DSA counts this as distraction. However, the visual distraction is weighted higher (3 points per second) than the mere cognitive distraction (2 points per second). The continuous distraction score is calculated from the sum of both distraction inputs added over time. If the driver makes a meaningful interruption of the secondary task of more than 2 seconds, the distraction score is set to zero again.

The DSA distraction output is set to 1=distracted if a certain threshold of the continuous distraction score is exceeded. As an important result from the first DSA validation study (see Deliverable D32.2), this threshold was adapted to the current driving task demands and the automation level. If the driver is driving in a low demanding driving situation, e.g. a traffic jam, he expects to be allowed to interact more with a secondary task before being defined as unacceptably distracted. If the driver is driving in already high demanding situations where he might have to intervene very fast only very short or better even no secondary activities should be allowed. For a higher acceptance of AM interventions it was decided to allow more distracting activities when driving on Highly Automated level, at least in less demanding driving situations. However, as soon as the situation gets more demanding a direct intervention of the AM in the way of a warning message has to take place.

Interaction design of the Attention Monitor for distracted driving

The general principles of this Attention Monitor for the use case "driver distracted" can be described as follows:

The first escalation step and with it the information message is triggered if the driver exceeds the threshold for the distraction score the first time (DSA distraction output=1). A dead time of 10 sec after an AM intervention is considered to allow the driver to finish an already started secondary task without being warned again.

If the driver still remains distracted after this 10 sec dead time (meaning still or again above the threshold of the distraction score) the second escalation step and with it the warning message is triggered. If the driver is driving in Highly Automated mode and is still distracted after the warning message has already been triggered a take-over request is given and the driver has to perform a transition to Driver Assisted level.

If the distraction score stays below the threshold for longer than 1 min the escalation starts again on level 1 with an information message the next time the driver exceeds the threshold. If the threshold is exceeded in high demanding driving situations the escalation is directly started on escalation step 2 with a warning message.

The fourth escalation step "Minimum Risk State" in case of a driver not reacting to the take-over request is not implemented in that version of the AM.

The information message triggered on the first escalation step consisted of an orange icon of an opened eye and the text message "please be attentive" shown in the Automation Monitor display of the HAVEit HMI (see Figure 133 left). In addition, a row of red LED lights arranged on the dashboard below the windscreen flashed for some seconds to direct driver's attention back to the road (see Figure 133 right). Those LEDs were included as the drivers complained in the first study (see deliverable D33.3) that the mere information message was hard to notice while being engaged in a secondary task.

The warning message on the second escalation step included the same display and flashing lights as well as an acoustic sound (a "gong") to direct driver's attention. In case of the take-over request for the Highly Automated group on the third escalation step, a big orange cross and a symbol for hands-on request was displayed together with the text message "please take over-not attentive enough" and an acoustic sound (three times "gong").

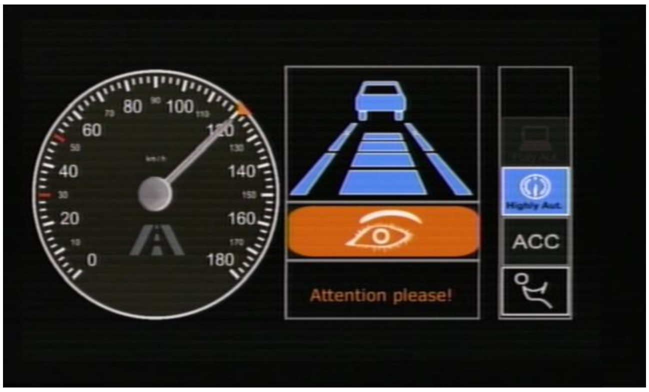


Figure 133: Info message in case of distraction (attention request; left) and flashing LEDs in the windscreen (right)

All 12 test drivers each had to drive three times through the 20 minutes course under three experimental conditions. In all three drives they were instructed to perform the secondary task as they liked. The three conditions were:

- Driving on Driver Assisted (DA) level of the HAVEit system with the Attention Monitor active: In this condition the drivers drove the test course on the Driver Assisted level. This means that they were slightly assisted by a lane departure warning system which only intervened in critical situations when the driver tended to get off the road by providing a slight force on the steering wheel towards the opposite direction. They were told that the Attention Monitor was active and intervened if the distraction level of the driver gets too high.
- Driving on Highly Automated (HA) level of the HAVEit system with the Attention Monitor active: In this condition the drivers drove the test course on the Highly Automated level. On this level, lateral and longitudinal control was performed automatically by the system up to a speed of 130 km/h. The driver could drive hands-off. In case of a situation the system could not manage (e.g. too large relative speeds between two vehicles) a take-over request was given. The driver then had to take-over the steering wheel and the pedals and a transition towards the Driver Assisted level was executed. They were told that the Attention Monitor was active and intervened if the distraction level of the driver gets too high.
- Driving on either Highly Automated level (6 drivers) or Driver Assisted level (6 drivers) of the HAVEit system with the Attention Monitor inactive. This condition served as a kind of baseline for the effects of the Attention Monitor itself on the engagement in the secondary task (will the driver do more or less tasks when AM is active compared to when it is inactive?)

After each of the 20 minutes drives the drivers were instructed to leave the highway on a parking area. Here they were requested to answer some questions about the previous drive online in the vehicle (e.g. “was the drive stressful or relaxed?” or “where the interventions of the Attention Monitor justified or unjustified?”). After all three drives drivers received another questionnaire to answer global questions on the Attention Monitor in case of distraction. In addition, they had the opportunity to give further comments in an open interview.

Results and conclusions

In general, one can summarize that the Attention Monitor in case of distraction was rated quite positive by the drivers. This is not self-evident, especially if one considers that it is a system that counteracts the driver’s intention to interact with a deliberately selected secondary task. If the Attention Monitor is explained as an assistance system that supports the driver and if the system is designed in an appropriate way a good acceptance can be reached. Drivers rated the Attention Monitor as a quite meaningful and useful system. Even a take-over request in case of repeated distraction is not perceived as domination by the system, but as quite meaningful.

The messages of the Attention Monitor were rated as quite comprehensible. The additional LED lights in the windscreen seem to be a very promising method to redirect driver’s attention to the driving task in case of distraction. In general, with regard to system understanding and acceptance the Attention Monitor and its HMI has to be considered as one subsystem within the later global HAVEit system. Therefore, it must be understandable for the driver which messages or transitions of automation levels are triggered by the AM and rely on driver’s state and which one are due to automation related limits.

Very interesting was the finding that drivers rated the interventions of the Attention Monitor more justified if they drive in Highly Automated mode. They expect more support here as they are aware of the higher risk of getting out of the loop. If they drive for themselves or only slightly assisted they are mostly able to use meaningful and deliberate strategies to interact with a secondary task. Those strategies seem to get lost when driving on higher automation levels. Under this aspect the current concept of allowing more interaction with secondary tasks when driving on higher automation level should be reconsidered. One could alternatively think of warning the driver only in critical situations when he is driving manually or slightly assisted but use the current solution of stepwise escalation for the Highly Automated mode to remember the driver of the increased risk of getting out of the loop.

Furthermore, it turned out that drivers accepted the AM interventions and reacted to it in a meaningful way. In critical situations for example when warnings or take-over requests are triggered they usually immediately interrupt the secondary task. In less demanding situations when the information message is given they sometimes allow themselves to finish an already started task or at least try to get in an optimal position to resume to it later very quickly. This strategy is completely acceptable and is already included in the current design of the Attention Monitor by the dead time of 10 seconds in non-critical situations after an AM intervention where the driver can shortly finish a task without being classified as distracted again.

The Attention Monitor as it was realized in the present study had positive effects only on the subjective perception of driving safety. In comparison to drives without AM where drivers already had a quite good lateral driving performance, no further improvements occurred with AM. However, the AM seems to not additionally distract the drivers as also no decrements in performance could be observed. To reach a clear positive effect especially in critical driving situations the warnings in case of distraction have to be triggered earlier to enable the driver to better react to the situation. This is however heavily dependent from the quality of situation detection.

Evaluation of two different transition designs for Temporary Auto-Pilot

Objective of this study

One aspect of this study was concerned with answering the question of what system responses the user intuitively expects on intervening in the vehicle control system with specific driver inputs during highly automated driving using the "Temporary Autopilot" (TAP) function. The main focus in this case was directed towards steering and braking manoeuvres within the vehicle's own lane as well as lane changes initiated by the driver. The user information received for this purpose was used for deriving the first transition-related design cues.

Furthermore, the study focused on the effect of system responses both in accordance with and not in accordance with expectations on the driver-vehicle-interaction. For this purpose, the users experienced two configurations of the TAP subsequently to being asked about their expectations. These differ in terms of the type and manner of the transitions triggered by the driving manoeuvres. With regard to the behavioural and expectation data obtained, it was therefore possible to supplement and adapt the already derived design cues. It was particularly interesting to identify whether the user wants consistent system behaviour across all situations under study, or if specific special cases can be identified.

These questions were supplemented by subjective evaluations of the system status display and the general system operation. The information obtained was used for refining the human machine interface (HMI).

In addition, this study integrated two specific driving situations in which the TAP showed a special behaviour. The first scenario concerned the situation of overtaking on the right in which the system independently adapted the speed in order to avoid infringing the rules of the road, and therefore did not overtake. In the second scenario, the subject matter was automatic reactivation of the TAP following partial or complete deactivation of the system due to the driver changing lane. In both scenarios, the point of interest was whether the automatic system behaviour is transparent for the user and whether he/she accepts this in the selected situations.

Method

In order to answer these questions, a road test was carried out in real traffic. The general conditions applicable for the investigation are explained in the following sections. These concern the test vehicle, the functional configurations considered as well as the test scenarios, the test road, the integrated HMI, the test sequence and the random sample used.

The investigation was carried out with a test vehicle from Volkswagen AG based on the Wizard of Oz (WOz) method (see Baum, 1900[59]). Similarly to this, an interface is used which is invisible to the user and responds to the user's inputs by means of a wizard as a deputy system (see Kraiss, 2004[60]). System components as well as complex, technical processes are therefore taken over and simulated by a human. The user himself does not know anything about this simulation in most cases. Therefore, the user expresses his wishes or requirements on the object which is "real" for him, and interacts with this in a relatively natural way.

In the case in question, a second driver seat together with pedals and multifunction steering wheel was set up in the back seat of a VW Touran (Figure 134). A partition wall with tinted, mirrored glass guaranteed that it was only possible to see from the rear forwards and not vice versa. The wizard concealed in this way sat in the middle of the rear of the vehicle and interacted from there with the actual driver by means of the available actuator systems (accelerator, brake, steering). In this regard, it was possible to reconstruct different degrees of automation by means of operating the particular driving functions, as well as to simulate specific configurations of lateral and longitudinal control. If the wizard takes over the accelerator and brake pedal function, for example, this means the test person can use the ACC function, and has to perform only the steering task.

Figure 134: Used test vehicle

This vehicle allowed simulating the range of functions as well as the previously existing system specification of the TAP (for information about this, see deliverables D53.2 and D53.3). In this regard, it should be mentioned that driving with the actual prototype (TAP) in real traffic conditions was not possible due to safety reasons. As a result, the test vehicle described was used as a methodological tool in order to conduct explorative prototyping, to evaluate specific functional configurations of the TAP as well as to derive fundamental configuration recommendations.

Because simulation of the TAP places exacting cognitive and motor-function requirements on the wizard, the test vehicle was controlled by a driver who had been trained on the VW Touran. His ability to reproduce the mode of function of specific driver assistance systems had already

[59] Baum, L. F. (1900). The wonderful Wizard of Oz. Chicago: G. M. Hill

[60] Kraiss, K. F. (2004). Anthropotechnik in der Fahrzeug- und Prozessführung. Vorlesungsskript der RWTH Aachen am Lehrstuhl für Technische Informatik

been validated in evaluation studies (see Schmidt et al., 2008[61]). In addition, the wizard received extensive training prior to the road test. This focused in particular on a functionally realistic simulation of system responses which went hand-in-hand with predefined transitions towards the driver, depending on the TAP configuration used. The decisive factor in this was driver inputs and driving manoeuvres instructed by the test leader which the test persons were supposed to undertake during the test. The following section accordingly explains the functional configurations which were considered in the corresponding test scenarios of this study.

Functional configuration

This road test focused on steering and braking manoeuvres initiated by the driver within the vehicle's own lane, as well as lane changes by the driver starting from highly automated driving with TAP. Similarly to this, the system responses varied in the majority of the integrated test scenarios in the following two ways: Either there was transition from Highly Automated to Driver Only (see Transition A, Figure 135) or to Semi Automated (see Transition B, Figure 135). In this study Highly Automated corresponded to the prototype TAP and Semi-Automated to the function Adaptive Cruise Control (ACC). On the level Driver Only no assistance was active. Referring to this, the study is not focussed on the automation level Driver Assisted.

The starting point for considering the two system configurations of the TAP was the question as to whether there are situations in which a transition to ACC is more appropriate due to reasons of safety or comfort than for the control of the vehicle to be taken over completely by the driver. The argument in this case was based on the expectations on the system behaviour expressed by the users, as well as subjective assessment of the experienced vehicle response in the particular test scenarios. The type of instructed driver inputs is distinguished from the driving manoeuvres to be performed.

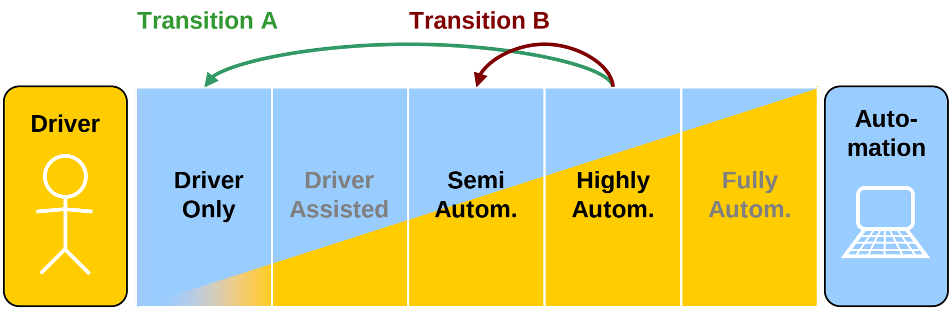


Figure 135: Integrated automation levels and transitions to the driver

Use case classes “Normal Driving + Lane Change”, “Normal Driving + Braking” and “Normal Driving + Steering” were assessed in this study (see D33.6 for details).

Different scenarios for these use cases were considered, e.g. for use case “Normal Driving + Lane Change”: Test persons were supposed to perform various kinds of lane change in these driving situations. The first and second cases were concerned with performing a "classic" overtaking manoeuvre. However, the difference concerned the situation that the test persons also accelerated during overtaking in the first situation. For this purpose, the test participants drove up to the accompanying vehicle which was always driving in front of the VW Touran. In the third scenario, a lane change was performed without the turn signals being activated.

[61] Schmidt, G., Kiss, M., Babbel, E. & Galla, A. (2008). The Wizard on Wheels: Rapid Prototyping and User Testing of Future Driver Assistance Using Wizard of Oz Technique in a Vehicle. Proceedings of the FISITA 2008 – The Future of Automobiles and Mobility, München

Because the reason of this driver intervention is non-transparent for the system, there was always a transition to Driver Only.

The road test took place on a section of approx. 70 km on the A39 motorway from Wolfsburg towards Salzgitter. The direction was reversed close to Braunschweig (at Helmstedt) for the vehicles to drive back to Wolfsburg. The speed limit for the majority of the route was 120 km/h. However, the maximum speed during the study was approx. 100 km/h because the test persons were concentrating relatively intensively on the driving manoeuvres. The entire journey undertaken by each test participant was 45 minutes on average.

As already mentioned, the VW Touran was accompanied by a second vehicle throughout the test. This drove in front in most cases, firstly in order to guide the test persons and show them which route to take. Secondly, the vehicle served as a means of protection when performing the selected driving manoeuvres which were characterised, for example, by driving closely behind the vehicle in front.

Before the test persons drove onto the A39 motorway, they initially filled in a questionnaire to provide general information of sociodemographic nature, including questions about their driving behaviour and experience with driver assistance systems. Following this, they were given a brief introduction to the test vehicle and the procedure in the investigation. Particular attention was paid to explaining the basic functions of the TAP in this case. However, a detailed explanation of the individual driving functions was not provided until afterwards. The test persons were initially asked to evaluate their first impression of the "Automation Mode Display". The focus in this case was directed towards transparency and comprehensibility of the individual automation levels. In addition, the test persons learned at the same time how and when they could activate a level in the road test.

Following this, the test participants drove onto the A39, and activated the TAP step-by-step. After this, they had the opportunity to get used to the highly automated system first of all, and to allow it to take over from them. Thereafter, the test persons described their overall impression of the graphical display on the instrument cluster. The main emphasis here was placed on assessing criteria such as look-and-feel, comprehensibility and operability.

Conclusion

In conclusion, it is possible to answer the research questions formulated initially by stating that the users already intuitively possess a very precise idea of how a function such as the TAP should behave. If the response differs from the expectation, however, the driver can very easily become confused. In that case, the users need more time to understand the changes of the system status. In order to avoid incompatibilities of this kind, the functional configuration should be built, firstly, on pre-existing system knowledge models. During the study, the test persons frequently mentioned the ACC system behaviour for this aspect. Secondly, the drivers made use of system responses they had already experienced, and mentally transferred these across to a driving situation that they regarded as similar. This means they created a model of the overall function that was consistent for them.

In order to integrate this in the development process, it makes sense to represent the mental model of the user in the form of status diagrams, and to transfer this to the target function. Accordingly, the results of this study make it possible to derive the transitions between the separate automation levels of the TAP as shown in Figure 136. At this juncture it should be noted that these exclusively relate to steering and braking manoeuvres within the vehicle's own lane as well as to lane changes that were initiated during a highly automated drive by the driver. In addition some of the derived transitions must be considered from a special perspective. Therefore, the required transition to Driver Only in case of strong intervention on steering wheel is justified because the users feel safer in this situation if they have control over the vehicle themselves and do not additionally have to deal with an unexpected system intervention.

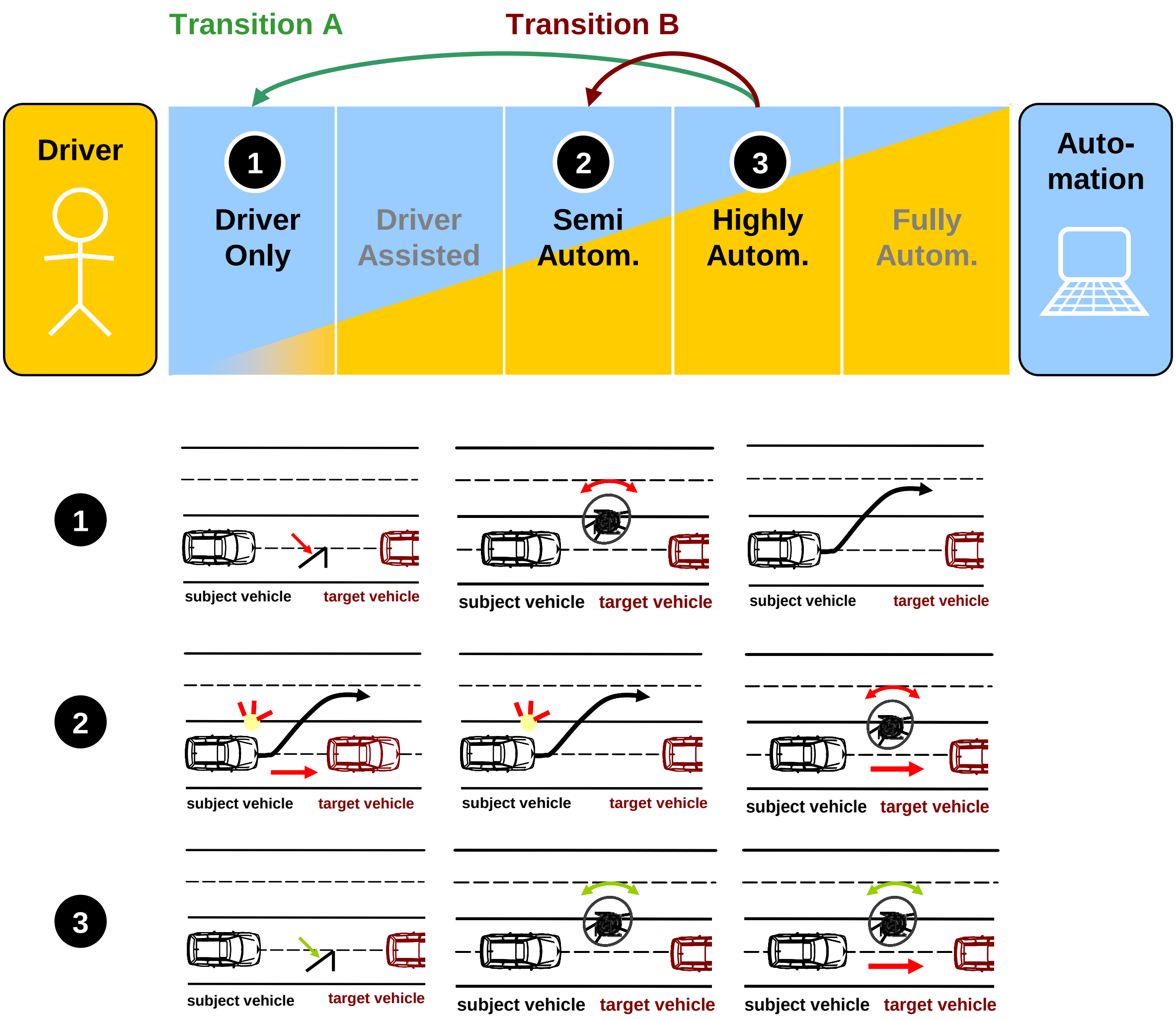


Figure 136: Derived transitions for the integrated automation levels of TAP

In the comparable case, involving the approach to a vehicle driving in front, the test persons wished for a transition to Semi-Automated (see Figure 136). The background to this is that they associate the ACC function with a type of collision protection. However, this rear-end collision protection can also be guaranteed by other systems in the vehicle. In contrast to this, the test persons expected no change to the system status in the same approach situation if they only exert slight steering moments (see case 3 in Figure 136). For this case, it should be also noted that they nevertheless wish to have an audible and/or visible warning if the distance from the vehicle in front is too short.

Furthermore, it was possible to establish for the two separately considered cases of automatic system interventions that these are transparent and desirable for the user if certain preconditions are met. Accordingly, automatic system reactivations following a lane change performed by the driver is rather recommendable after longer periods of use of the TAP. On the other hand, an automatic brake intervention to avoid overtaking on the right on a motorway at speeds below 80 km/h requires a clear visual explanation of the system response. This should contain information, best given in symbolic form, establishing a link or analogy with the concerning situation.

In addition, the used HMI was evaluated very positively by the test persons. In particular, the displaying symbols as well as the graduated structure of the overall function were regarded as transparent, simple and understandable. However, the pictogram of the driver in the automation level Driver Only should be supplemented by a picture of a steering wheel.

Taken as a whole, the TAP function was perceived very positively by the test persons, and regarded as sensible for the "motorway" application context. In this case, the test participants wanted to use the system for a speed range from 0 to 160 km/h. At the same time, they could imagine operating the system by means of the steering wheel or the steering column switches. For the first case, buttons or cross rocker switches are particularly suitable for deactivating and activating the TAP. The test persons would like to push the control element upwards or to the right in order to switch the system on.

<u>Comparison of driver initiated transitions for four HAVEit transition schemes/prototypes</u>

<u>Objective of this study</u>

The study explored different design options for driver initiated transitions between the HAVEit automation levels Driver Assisted (DA), Semi-Automated (SA) and Highly Automated (HA) (Figure 137). Transitions are defined as the changes between the different automation levels, e.g. from Highly Automated to Driver Assisted. While the previously described study of VW evaluated different transition variants for the TAP system this study compared the four different prototypes of WP 5100 (Automated Assistance in Roadworks and Congestion – ARC), 5200 (Automated Queue Assistance – AQuA), 5300 (Temporary Auto-Pilot – TAP) and SP3000 (Joint System Demonstrator – JSD) and explored if the overall transition design of the prototypes is comprehensible for the driver.

Main objective of the study was to find out if the behaviour of the prototypes fits to the drivers' expectation and if there are any differences in the evaluation of the prototypes behaviour so that one can give advice which prototype design of the transitions might be best. As highly automated driving is a relatively new topic there were a lot of discussions within the HAVEit consortium which transition might be best in reaction to drivers' input like braking, accelerating or steering.

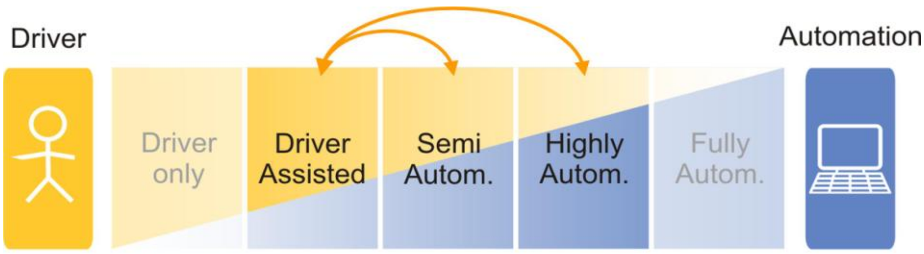


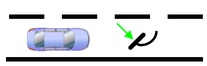
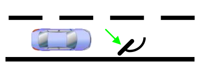
Figure 137: HAVEit Automation levels and transitions that were under research in the DLR study

The goal of this study was to provide a database for alignment of the prototypes and hints for design improvements. In the study of VW one result was that drivers based their expectations about the prototype behaviour on their previous experience acquired with other assistance systems like ACC or with the prototype itself in similar driving situations. Based on that result, the present study explored the effects of the experience with the levels Driver Assisted and Semi Automated of the HAVEit prototypes on the expectations regarding the transition in the level Highly Automated. To consider the different expectations by the users, 4 groups of eight drivers were built and each group tested the transition scheme of one of the 4 demonstrators (ARC, AQuA, TAP and JSD) in the DLR simulator.

Excerpt: Expectations regarding the transitions in Highly Automated before the test drive

For interpreting the results of the expectation assessment it is necessary to keep in mind that drivers in the group testing Prototype A experienced a different prototype reaction in Semi Automated than drivers testing Prototypes B, C or D. For Prototype B, C and D the transition design was the same. Multiple answers were possible and Table 29 presents the number of answers per category. Four of the eight drivers testing Prototype A mentioned that they expect that the prototype differentiates between slight and strong acceleration (4 of 8 answers in this category) as they experienced it in the level Semi Automated. Some drivers also expect that there is no transition and that the prototype stays in Highly Automated (3 of 8 answers in this category). Compared to this, none of the drivers testing Prototype B, C or D expected a differentiation between slight and strong acceleration. Instead, they expected most often that the prototype stays in the level Highly Automated (between 4 to 5 answers in this category) as they experienced it in the level Semi Automated. Transitions from Highly Automated to Semi Automated or Driver Assisted were mentioned next often (mentioned 1 to 3 times per category).

Table 29 (right columns) shows the results for acceleration beyond the prototype speed limit of 130km/h. Drivers of all groups mentioned most often that they expect that the prototype starts a transition from Highly Automated to Driver Assisted (5 to 6 answers in this category) as they experienced it in the level Semi Automated. They also mentioned a few times that there could be a transition to Semi Automated (1 to 3 answers in this category) or that the prototype stays in the level Highly Automated (1 to 2 answers in this category).

	acc. < 130km/h 				acc. > 130km/h 			
	A	**B**	**C**	**D**	**A**	**B**	**C**	**D**
stays in HA	3	**5**	4	**5**	2	1	--	2
stays in HA for slight acceleration; transition DA$_i$←HA for strong acceleration	**4**	--	--	--	--	1	--	--
stays in HA for short acceleration; transition DA$_i$←HA for acc. longer than 3 sec.	--	--	--	--	--	--	--	--
transition SA$_i$←HA	1	3	1	3	1	2	3	1
transition DA$_i$←HA	--	3	3	1	**6**	**6**	**5**	**6**
not sure what happens	--	1	--	--	--	--	--	--
Total number of answers	8	12	8	9	9	10	8	9

* Numbers marked in **bold** represents expectations that are consistent with the transition design in Semi Automated. Cells marked in grey indicate the actual prototype design that drivers experience in the level Highly Automated. Numbers indicate how often a category was mentioned.

Table 29: Expectations regarding the transition design in Highly Automated for accelerating up to (left columns) and beyond 130km/h (right columns) (multiple answers were possible).

Excerpt: Mental model and evaluation of a transition after the test drive

After the expectation assessment the drivers tested the prototype in the level Highly Automated in a test drive (“Second Free Drive Highly Automated”). Thereafter, the drivers were interviewed about their understanding of the transition design of the prototype they tested. Answers were classified as correct, if the driver reported a correct description of the prototype reaction, as incomplete if the driver reported some but not the complete prototype reaction correctly and as incorrect if he reported the wrong prototype reaction. Answers were categorized as “not valid” if the driver could not describe the prototype behaviour because he did not test how the prototype reacted to a specific driver input during the test drive.

In Figure 138 the correctness of understanding for the prototype reaction to accelerating up to and beyond 130 km/h is described. For Prototype A, B, and C 7 of 8 drivers gave a correct description of the prototype reaction to accelerating up to 130km/h. For Prototype D 4 of 8 drivers reported a correct understanding while the other four showed an incomplete understanding due to the time based transition of this prototype. These drivers did not understand that there is a transition back to Driver Assisted after a short time period of 3 seconds. For accelerating above the prototype speed limit of 130km/h 6 respectively 7 drivers per group reported the correct prototype reaction to their input. The high number of driver that report a correct understanding of the prototype transition design shows that drivers could build up a correct understanding even if the transition design of the prototype did not fit to their initial expectation (see for example Prototype C, Table 29: only 1 driver expected the actual prototype behaviour but 6 drivers report a correct understanding after the test drive).

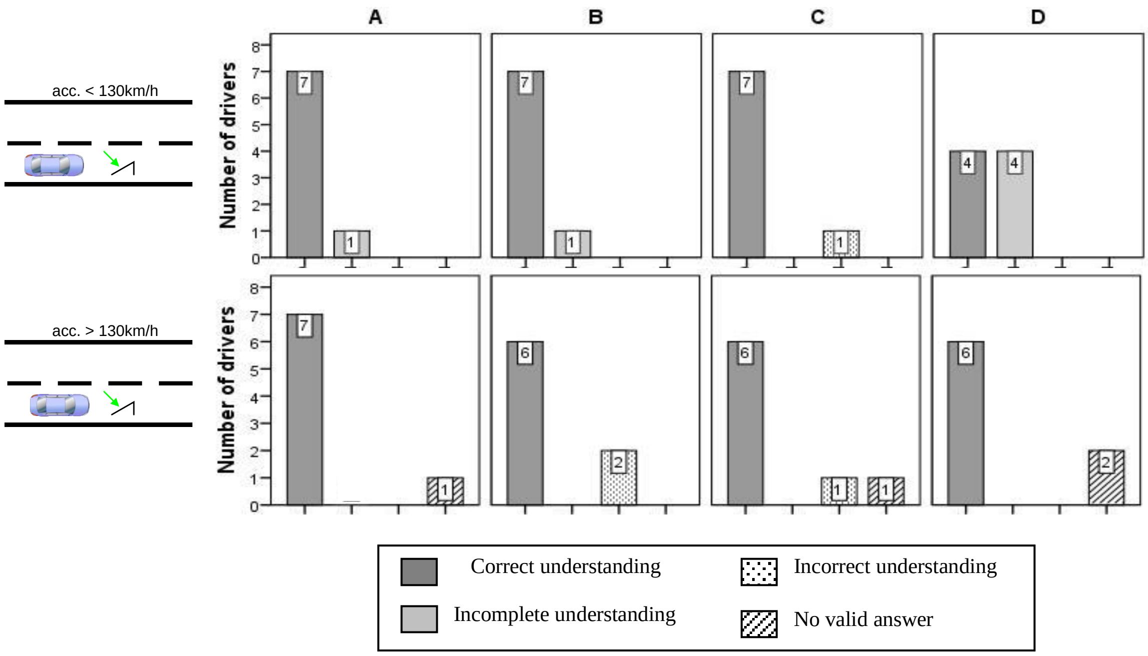


Figure 138: Quality of understanding of the transition design for each prototype for accelerating up to (above) and beyond 130km/h (below).

<u>Overall evaluation results</u>

Display evaluation

After the test drive the participants were asked to evaluate the display concept (see Figure 139). As all four prototypes worked with the same display the ratings of all 32 participants were analyzed in one analysis together. When asked which indicators the participants used to check in which automation level they were driving in, the display was most often mentioned. The display concept was rated as quite good (m=2.00, sd=1.08), as quite easy to understand (m=1.84, sd=1.17), as very necessary (m=2.53, sd=0.98) and as quite easy to read (m=2.31, sd=0.82).

When asked how well the drivers could comprehend on the display in which automation level they were in, the ratings ranged from rather bad (-2) to very good (+3). On average the drivers said that they understood rather well on the display in which automation level they were in (m=2.47, sd=1.08).

Highly automated vehicles evaluation

After the test drives the drivers were also asked to evaluate the highly automated vehicle with the three different automation levels that they drove during the test drives. An analysis of variance showed that there were no differences in the ratings for the four prototypes. That is why the ratings of all 32 participants are presented together. As shown in Figure 139 the ratings were overall positive and ranged on the average from rather (+1) to quite (+2) good. Regarding the question of the alertness level the drivers said that such a vehicle was on average quite sleep inducing (m=-0,59, sd=1,59).

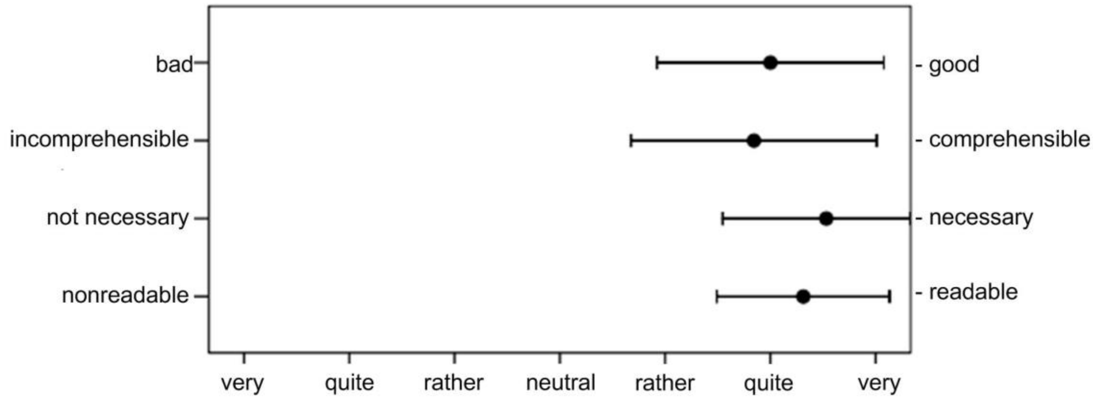


Figure 139: Assessment of the display concept for the highly automated vehicle (answers of 32 drivers; means +/- 1sd)

The drivers were also asked how much they would like to drive such a highly automated vehicle if such vehicles would be available in the future. Here, most of the participants were positively interested (Figure 140). Five participants gave a neutral to very negative rating. When asked about the reasons for their ratings they mentioned that they might get drowsy or distracted and saw this as a potential downside of highly automated driving.

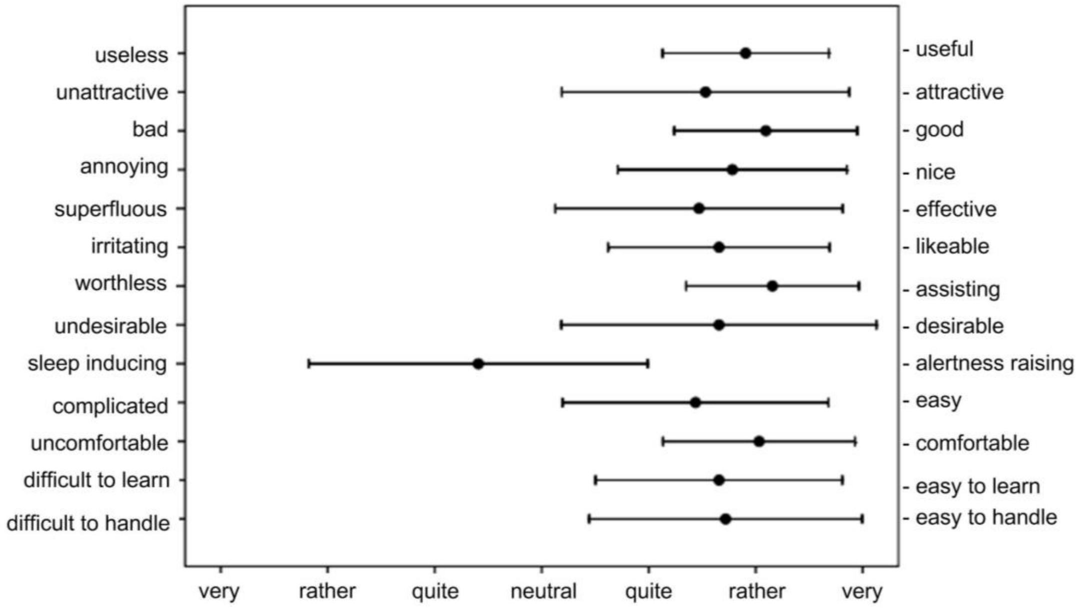


Figure 140: Evaluation of the highly automated vehicle (answers of 32 drivers; means +/- 1sd)

In addition, three drivers mentioned that they do not want to hand over control to vehicle automation. 27 participants rated that they would rather to very much like to drive such a vehicle. The drivers liked the increase in comfort and safety and mentioned that they think that such highly automated vehicles would allow relaxed driving especially on highways.

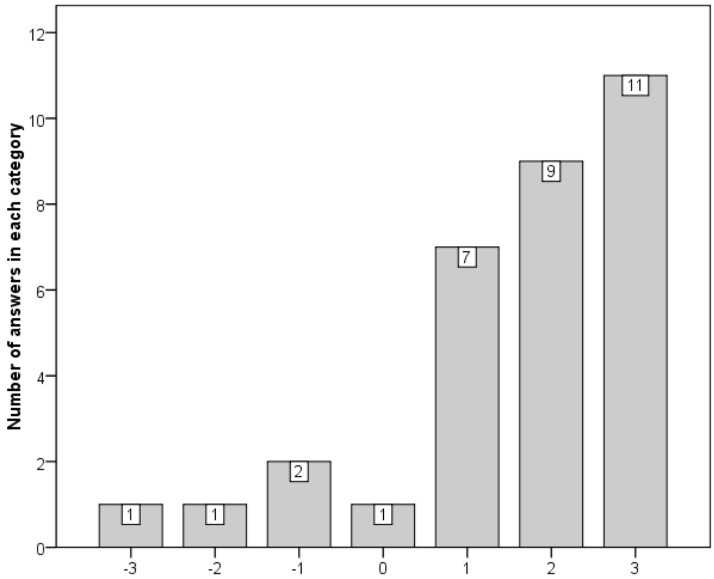


Figure 141: Evaluation of willingness to drive a highly automated vehicle (number of answers in each category for 32 drivers)

Conclusions on the four evaluated transition schemes

The results of the expectation assessment showed that the drivers' expectations seemed to be influenced by the prototype behaviour that they experienced in the level Semi-Automated during the training run. While drivers of group A expected a difference for slight and strong accelerating or braking this was not the case for drivers of group B, C and D. Drivers of these groups expected most often a prototype reaction that was compatible to the behaviour in the level Semi Automated, e.g. that the prototype would stay in Highly Automated when accelerating below the prototype speed range or that braking would lead to a transition from Highly Automated to Driver Assisted.

However, the participants were flexible in adapating to the prototype behaviour and had no problems to understand the transitions even in cases in which the transitions did not match their expectations. Most of the participants were able to build up a correct mental model of the prototypes behaviour for the different driving situations and could reflect this behaviour correctly in the interviews. No significant differences were found for most of the evaluation criteria that were assessed after the free test drive and the structured runs. Some improvements for the single prototypes can be deduced from the interview data.

During the drives some signs of mode confusion occurred. In general, mode confusion is a critical issue for the safety of highly automated driving especially when hands-off driving is allowed. An appropriate interaction design might help to avoid situations in which a control vacuum occurs and neither the driver nor the automation is in control of the vehicle.

One approach is to have an appropriate display design that explicitly shows the activated level of automation like it is done in the HAVEit automation scale and the colour changes in the display. This is supported by the result that the drivers took the display as most important indicator of the automation level and said that they understood rather well on the display in which level they were in. Another option is to apply a hands-on check for the transition to a lower level of automation or a hands-on warning for the levels Semi-Automated and Driver Assisted. The hands-on check should ensure that the drivers put their hands back on the steering wheel. Depending on the criticality of the driving situation the prototype might be tuned in a way that the transition only takes place when the result of the hands-on check is positive (like it is described in the interaction scheme "interlocked transition" in D33.3). The hands-on warning is a warning for the levels Semi-Automated and Driver Assisted and it alerts the driver to put his hands back on the steering wheel in case he forgot during the drive in which level he is in. Another solution is to have a specific sound design with sounds that are easy to distinguish for upward and downward transitions and warning.

One design option that needs to be carefully checked with regard to mode confusion is an automatic reactivation of the level Highly Automated, e.g. after a lane change manoeuvre. Such a design might be comfortable but might lead to mode confusion if the driver thinks by mistake that the automatic reactivation to Highly Automated already took place. This is especially critical for a design in which the automatic reactivation of the level Highly Automated happens only when a specific criterion, e.g. using the indicator for a lane change, is fulfilled. As long as the driver does not include this criterion in his mental model he might expect an automatic reactivation even though this would not happen because the criterion might not be fulfilled. Mode awareness might be supported by consistent transition design, including for example the above mentioned specific sounds for downward and upward transitions.

The evaluation of the one of the apparent HMI elements, the primary display, resulted in positive ratings. The display seems to fit well to the information requirements that drivers have when driving a highly automated vehicle.

When asked about their evaluation of the highly automated vehicle, drivers gave positive ratings regarding satisfaction and usefulness. They saw potential advantages in a relaxed, safer and more comfortable driving especially on highways. As downsides they mentioned that they might become more distracted or drowsy or do not want to give control away to an automation. When asked whether they would like to drive such a highly automated vehicle when it will be available in the future most of the participants were rather to very interested. This result shows that highly automated driving is appealing to people especially for driving long distances on highways.

5.3.4 Summary and conclusions for the final Joint System design

As highly automated driving is a new topic in the vehicle domain, at the beginning of HAVEit there was rather little knowledge of how to design the optimum task repartition between the automation and the driver in different driving situations. Therefore, the Human Factors work in HAVEit established an iterative approach of designing, testing and redesigning of the prototypes. In first studies (documented in section 5.3.3 above and in deliverable D33.3) we tested the initial design for the prototypes and addressed generic research and usability questions. In the deliverable D33.6 we documented studies that tested the HAVEit prototype designs in their refined versions and addressed some of the open and most relevant research questions. The main foci of these studies were the interaction design for the use cases regarding an inattentive driver, the refined HMI design and different transition variants that are under discussion for the HAVEit demonstrator vehicles.

Overall evaluation of highly automated driving

All studies showed positive results for the first contact of drivers with highly automated vehicles. In the studies the vehicle prototypes received positive acceptance and usefulness ratings. Participants were surprised about the technical feasibilities of automating a vehicle and enjoyed driving it. When asked about positive aspects of highly automated driving the participants said that highly automated driving provided good support and increased safety and comfort especially on highways. However, it seems to be also important to monitor potential downsides of highly automated driving closely. Some participants mentioned that they did not like to hand over control to the automation or that they were concerned to get too inattentive or drowsy while driving Highly Automated. In HAVEit cases of driver drowsiness and distraction are handled by the Attention Monitor.

Evaluation of the Attention Monitor

Especially for the cases that the driver gets drowsy or distracted during the drive, the HAVEit project already addressed and established a Driver State Assessment (DSA) on which basis warnings and transitions are triggered to ensure an optimum task repartition between the driver and the automation according to the driver state. In the studies presented in this section a refined interaction and interface design was tested successfully. The interaction strategies and the corresponding interface design were highly accepted - both in case of drowsiness and even in the more complex case of distraction. Drivers were aware of the higher risks of becoming inattentive, especially when driving in higher automation levels and they perceived an Attention Monitor as helpful.

Especially in the case of distraction, reaching a relatively high acceptance is a noteworthy achievement. Here, the refined interaction design and the underlying warning algorithms seem to be designed in such a way, that the driver accepted the system as a support system and not as an annoying warning system. In the current design, even system-initiated transitions to lower automation levels were not perceived as domination by the system. However, it needs to be clear that acceptance is only one factor in the evaluation of the HAVEit design. Another factor is safety: For the definition of interaction strategies the first priority lay on the avoidance of misuse of the HAVEit system. As HAVEit does not aim for fully automated driving but for highly automated driving where the driver still remains responsible for the driving task it has to be assured that he does not take himself out of the loop. Especially in case of drowsiness one cannot expect the driver to stay alert or become more alert again by the system intervention. However, the Driver State Assessment component and the accompanied interaction strategy of the co-system supports the driver to stay in the loop and persuades him to take a break or to redirect his attention back to the road what was successfully shown in the study results.

With regard to system understanding and acceptance the Attention Monitor and its HMI has to be considered as one subsystem within the HAVEit co-system. The studies revealed that the driver wants to understand which messages or transitions are triggered by the Attention Monitor and therefore depend on driver’s state and which messages are due to automation related limits. Drivers like to get a good mental model of the prototype behaviour and its transitions. That is why an adequate display design and e.g. extra messages as such used for the study raise the acceptance and understanding of the prototype reaction. The messages should give a short explanation of the transition reason to allow the driver to build up a correct mental model.

Evaluation of the transitions

Regarding the optimum task repartition transitions are not only triggered by the Driver State Assessment but also by driver input like steering or accelerating or by situational demands for example at system limits. Transitions between the automation levels are a critical design issue as the driver has to be aware which driving tasks he is responsible for. Because of this, generic interaction schemes regarding the definition of automation levels, the transitions, and the display concept were established at the beginning of the project (see deliverables D33.2 and D33.3). Different transition variants were carefully checked in the experiments to find out if the transitions fit to the expectation of the drivers, how the drivers react to automation level changes and if the drivers are able to build up a correct mental model of the transitions.

The results of both studies reported in subsection 5.3.4 showed that most drivers were able to adapt to the specific transitions and built up correct mental models about the prototype reactions even though they might have had different expectations regarding these reactions. The expectations the drivers had about the transitions were obviously based on their previous experience with other systems (e.g. cruise control or ACC) or lower automation levels in the HAVEit prototypes. The drivers seemed to have transferred their knowledge and expected compatible transitions for the level Highly Automated.

In addition, drivers seemed to build up generic schemes of the prototype behaviour while driving in the level Highly Automated and transferred this to other situations. An example is the expectation that a slight driver input, here slight braking and slight steering, would always lead to a similar reaction.

Summing up, it seems to be valuable if the prototype behaviour with regard to the transitions is compatible to other already existing systems and consistent in its behaviour itself. This goal of transferability of understanding from existing to new human machine systems has to be carefully balanced against the chances opened up by new concepts and systems. It is important to note here that the drivers' reaction to divergent system configurations, e.g. if drivers drive different cars, was not tested. There might be negative side effects regarding acceptance and safety if fundamental behaviour aspects are too different between cars. There is a clear need for a minimum amount of standardization for highly automated vehicle systems.

Evaluation of the display

One of the most apparent HMI elements of the HAVEit vehicles will be the display in the instrument cluster. Right from the beginning of the project the partners from the demonstrator vehicles started an alignment and harmonization process about the display concept and came up with three generic display elements. Those are the Automation Monitor, the Automation Scale and an area for text messages and warnings. The displays of all demonstrator vehicles include these elements and were further aligned during workshops and discussions. The Human Factors studies conducted in HAVEit showed that the generic display concept seems to provide the right level of information for the drivers. In all studies the specific demonstrator variants achieved positive ratings. In general, it seems to be valuable that the display concepts for highly automated vehicles are somehow standardized in its generic elements to allow the driver to easily change between cars of different manufacturers. Here, the HAVEit display elements seem to be one possibility in the wide design space for such generic display elements for highly automated vehicles.

Conclusion

Summing up, the Human Factors work in HAVEit provided important interaction design schemes for the new area of highly automated driving. The schemes were used as a basis for the design and implementation of the specific prototypes of the highly automated vehicle. During the course of the project these prototypes were intensively tested to answer several research questions regarding the interaction design and the effects of the prototypes on the drivers. The studies hinted aspects for improvement, but also showed that the interaction design is already adequate for most of the use cases, with positive ratings by the test drivers and experts.

Overall, it is to be concluded that a suitable Joint System driver / co-system has been successfully developed and HAVEit challenge 3.3 has been achieved. Consequently, all challenges of cluster 3 "Joint System" have been met. Thus, all horizontal issues in HAVEit have been solved.

6 Vertical Challenges: Safety Architecture Applications

In this section, results achieved within the frame of HAVEit's vertical challenges in cluster 4 are described in detail. To summarize:

- Challenge 4.1: Joint System Demonstrator (JSD)
- Challenge 4.2: Brake-by-Wire Truck (BbW)
- Challenge 4.3: Architecture Migration Demonstrator (AMD)

6.1 Challenge 4.1: Joint System Demonstrator

The previous section 5 described the HAVEit Joint System in detail. The basic principles of highly automated driving utilizing the Joint System were developed, refined and tested in simulators first, documented e.g. in deliverable D33.3. In addition to the simulator testing, the co-system was integrated and tested in a couple of integration phases into the DLR experimental vehicle FASCar – the so-called Joint System Demonstrator. As a benchmark accompanying the ongoing transfer of concepts and algorithms between horizontal and vertical subprojects, the integration and test of the Joint System demonstrator is documented in this section.

Focus is laid on the tests and results organized regarding the relevant HAVEit use cases. The highest priority was assigned to the automation level Highly Automated, because of the highest functional complexity of this automation level.

6.1.1 Configuration of the demonstrator vehicle

The demonstrator vehicle FASCar (Figure 142) served as test and development tool for the Joint System vehicle integration and demonstration. The vehicle properties allow demonstrating the full span of Joint System technology after finalising the commissioning phase. Its electronic architecture enables the control of longitudinal and lateral dynamics as well as enhanced HMI functionality like driver feedback methods (haptic, visual and acoustic feedback). The vehicle is also equipped with sensors for environment perception. The interface to the vehicle electronics provides electronic access to actuators for steering, brake and throttle which enables on-board Co-Pilot units to intervene into the current vehicle manoeuvre. Due to the demand of safety aspects the additional vehicle electronic devices are partly realized with redundant hardware systems and its software has been developed taking safety developing methods into account that were designed for avionic software systems.

The main sensors used in the FASCar to provide environmental information to the Joint System are a laser scanner system integrated in the demonstrators' front bumper and an in-vehicle camera connected to an image processing unit for lane marking detection. For development purposes this sensor system is complemented by high-precision DGPS data. An overview of the vehicle's sensor equipment is given in Table 30.

The FASCar has an open system architecture which provides an optimal development environment for rapid-prototyping of driver assistance systems and for tests and evaluations of close-to-production systems provided by the HAVEit partners. These close-to-production systems are integrated either as pure software code running on our hardware or even feature their own ECU, communicating via CAN-Bus or FlexRay. The validation of the assistance systems is conducted by means of specially adapted methods.

Figure 142: Demonstrator vehicle FASCar

Base vehicle	Volkswagen Passat B6 Extended interface to system electronics Drive-by-wire modifications Additional sensors
Environment perception	Lane detection camera Laser scanner object tracker
Additional sensors	Differential GPS unit I2V and V2V communication
Additional actuators	Active throttle pedal Driving wheel feedback actuator Active brake booster
System architecture	Redundant data management system Redundant ECUs based on avionic systems Fail silent ECUs FlexRay bus components

Table 30: Configuration of demonstrator vehicle FASCar

<u>Laser scanners</u>

The laser scanner system which is developed, adapted and provided by SICK in the HAVEit project consists of three single LUX sensors integrated in the demonstrators' front bumper. One sensor is installed in the centre of the vehicle looking straight ahead. Two further sensors are integrated in the left and right front corner of the demonstrator facing outwards at 30° to 40° (Figure 143).

The main purpose of the laser scanner system is the detection and tracking of other vehicles, pedestrians and stationary objects (guard rails) as basis for the data fusion which will calculate an overall environment perception taking the information of further sensors into account. The latest version of the laser scanner software also provides the opportunity of detecting lane markings. Together with the lane marking information of the front camera this additional feature might help to provide redundant and robust lane information.

Power supply, synchronization and data provision is performed by a central switch box. The LUX ECU processes raw data including time alignment, low level fusion with map data, object tracking and classification as well as relative positioning by providing relative distances to landmarks such as lane markings and crash barriers. Data acquired from the scanners are processed in a control unit supplied by SICK which provides a consistent object list aggregated from measurements of all installed units to other HAVEit applications. This resulting laser scanner output is provided as input to the HAVEit fusion system.

Figure 143: Laser scanner component and its final position mounted below the beams

Lane detection camera

The lane detection sensor in the FASCar is based on a detection system developed by Continental, a compact unit featuring the video camera and processing circuit in a single case. As a model improvement of the Passat B6, Volkswagen has recently introduced a lane keeping system based on this sensor type which is included in the Joint System demonstrator vehicle. Due to a modification of the firmware, this sensor is now providing raw data via the CAN bus and therefore no additional sensor has to be installed in the vehicle. The calibration of the sensor has been carried out by Volkswagen during the serial manufacturing process of the vehicle. As input the front camera system needs information about the ego vehicle state, e.g. subject vehicle speed and subject vehicle yaw rate. As an output the camera sensor provides information about the lane (e.g. lane width, lane curvature, lane marker width and type) and about the lateral position of the subject vehicle within the lane. The cycle time of this lane information is approximately 40ms.

This front camera system is mounted behind the windshield. Together with the rain and light sensor it is integrated in the bracket of the middle rear view mirror (Figure 144). The hardware configuration is used in Passat cars for series production.

GPS

The vehicle is equipped with a DGPS positioning unit in order to receive lane information from a digital map in order to extend the data fusion by lane matching and to support the fusion during commissioning activities. A Novatel SPAN CPT GPS system has been installed and calibrated providing all positioning information on CAN bus. The satellite receiver of this system is capable of RTK (real time kinematic) positioning operation, which means that very precise positions can be provided with low latency even when the vehicle is moving. This operation mode is only available if differential correction data is provided and satellite visibility is not obstructed. An empirical value often stated for the horizontal positioning accuracy of such systems is 2 cm with a latency of less than 50 ms, however due to the nature of the GPS system there is no guarantee of accuracy.

Figure 144: Location of the lane detection camera mounted above the rear mirror

The component has therefore been installed on a carrier plate in the trunk of the vehicle where it is firmly fixed to the vehicle body and carefully aligned with the vehicle's axis. While the SPAN CPT can automatically correct a deviation of its mounting position from the vehicle's centre of gravity, acceleration measurements can be improved by explicitly providing an offset vector taking the position of the centre of gravity of the vehicle into account. The external GPS antenna is mounted on the roof of the vehicle, represented in Figure 145.

Figure 145: GPS receiver antenna mounted on the roof of the vehicle

Complementary to the satellite receiver, the SPAN CPT system is equipped with an inertial measurement unit (IMU) consisting of three accelerometers and three gyrometers. The SPAN CPT system has a tight coupling between GPS and IMU data, thus allowing the GPS positioning algorithm to perform better under adverse satellite reception conditions. This concept goes one step beyond conventional GPS/IMU applications, in which typically position data from a standalone GPS receiver are combined with IMU measurements in a Kalman filter. It is generally very promising in the automotive domain, where bad satellite reception is much more common than in aviation or maritime applications.

However, in order to keep systems closer to series-production status, the HAVEit specification requires only standard consumer-grade GPS data to be available in the vehicle thus the final HAVEit systems will not depend on RTK data. Nevertheless, the availability of such information is particularly useful for testing and validation purposes as well as temporary bypassing sensor systems for e.g. controller development.

Communication hardware I2V, V2V

The communications hardware added to the FASCar demonstrator vehicle consists of a 4G Cube developed by partner INRIA (see challenge 2.2 in section 4.2 above) – a compact wireless router running SCOPE and OPENWRT router operating system – connected by Ethernet to the already existing vehicle onboard computer where the HAVEit Joint System Framework is running.

This communications system is a Vehicle Web Service Communication Framework (VWSCF) also called SCOPE that has automatic service discovery. This communication system was integrated into the HAVEit Joint System Framework that is running in the FASCar demonstrator vehicle and is suited for infrastructure-to-vehicle (I2V) communications as well as for inter-vehicular communications (V2V).

According to the HAVEit architecture the communication component is connected to the data fusion. There is no direct link between the on board computer and infrastructure. Communication is managed by 4G Cubes (provided by partner INRIA) and vehicle/infrastructure data is propagated to the dynamic mesh network.

The communication system is installed in order to extend and to improve the world perception in the vehicle by providing additional environmental information that the onboard sensors cannot provide due to their limited range.

The use of the wireless communication allows interactions between infrastructure and vehicle (I2V) as well as between vehicles (V2V). For each situation, I2V and V2V, a scenario will be demonstrated proving the concept and validating the proposed wireless communication system:

- I2V: a dynamic speed limit is sent by the infrastructure to the Joint System demonstrator vehicle that will adjust its current speed accordingly.
- V2V: a priority vehicle coming from behind (were there is a limited sensor coverage) will inform the ego vehicle of its presence that will remain in its current lane not being allowed to perform lane changes.

The communication architecture is modular and based on embedded Linux boxes (4G Cubes) that are used to provide automatic connectivity for X2V applications. These boxes can be used as simple routers, or like advanced communication units.

Driver monitoring system

The Driver Monitoring System (DMS) provided by Continental is a vision based system observing head and face of the driver in order to provide information about his/her state of degradation (see section 5.2 above for details). The system thereby provides support not only in the event of drowsiness, but also during actions which temporarily divert attention from the driving task, such as inserting a CD or operating the navigation equipment.

Figure 146 shows the integration into the Joint System demonstrator. The image processing algorithms of the DMS application are designed for the camera component to be mounted on the dashboard or in the instrument panel. The camera component position and orientation are optimized for monitoring the driver's face in the axis of the driver's eye ellipse midpoint.

Figure 146: Camera for direct driver state assessment

Vehicle data interface

The vehicle system architecture offers one private CAN port that enables bidirectional access to relevant vehicle data (see Table 31). With the ego-vehicle state provided by this CAN access and the additional information of the environment perception the joint system is able to generate a command vector that performs longitudinal and lateral control of the vehicle. This motion vector contains the demanded values for engine torque, brake pressure and steering angle and is extended by further values like gear-shift command, wipers or other HMI-commands.

Received data	
Environment perception data	Laser scanner data Lane detection camera data D-GPS data
Dynamic ego vehicle data	Accelerations Steering angle Yaw rate …
Inputs of the driver on the HMI	Buttons Pedal forces and position Steering wheel angle
Transmitted data	
Vehicle motion commands	Engine torque Brake pressure Steering angle
Other commands	Gear shift, wipers, horn

Table 31: Set of available, relevant data

While the data exchange of the sensor values and the set points for the longitudinal control uses the CAN communication the more safety and time critical steering system is connected to the steer-by-.wire system of the vehicle via FlexRay. An overview about the relevant CAN architecture is given in Figure 147. The vehicle behaves as slave to the Joint System. The demanded motion control is carried out automatically.

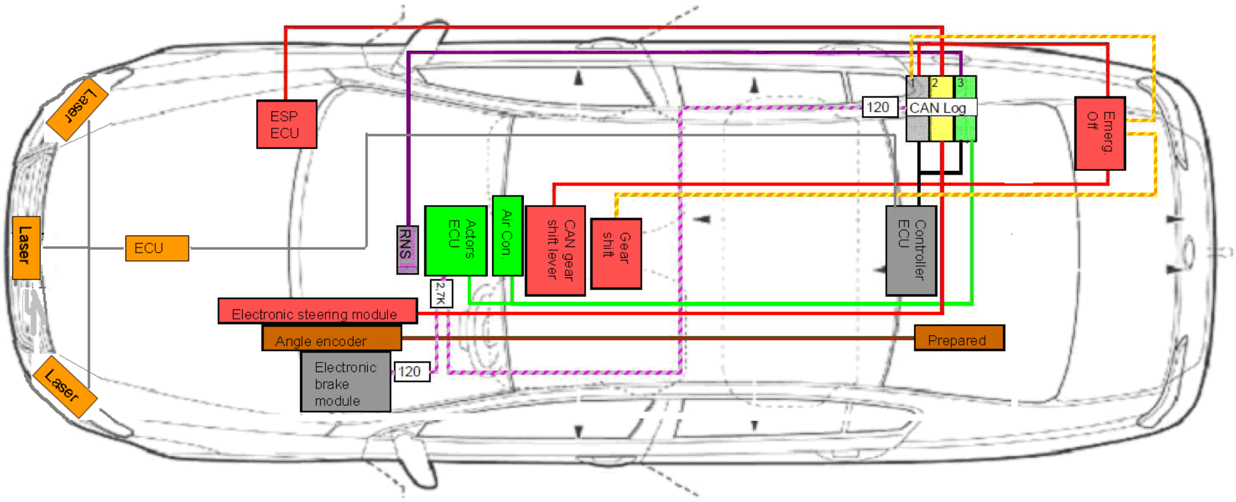


Figure 147: Basic system architecture of the FASCar

Drive-by-wire interface

The drive-by-wire interface of the vehicle allows a separate control unit (here the Joint System) to take control over the engine, the braking system and the steering system. The engine torque can be commanded by a torque vector that sets the current torque recommendation for the engine. For this purpose an electronic box has been implemented that emulates the voltage signal from the position sensors of the vehicle. In addition to that the gas pedal enables the Joint System to command separate force feedback signals, e.g. variable force points and vibration patterns are applied to the force feedback pedal.

The deceleration of the vehicle is commanded by a brake pressure signal which can apply up to 150 bar braking pressure on the serial hydraulic brake system. An active brake booster maintains both the mechanical connection of the braking pedal with the master brake cylinder as well as the pneumatic brake assistance function via its two vacuum chambers, thus ensuring that manual brake operation by the driver is possible at all times. Additionally, an electromagnetic valve is installed, enabling the device to actuate the main brake cylinder based on the electronic command. Typically, active brake boosters have an assigned ECU which accepts brake demands via a CAN interface and controls the hydraulic pressure in the braking system.

While the actuators for longitudinal control are series products, the steer-by-wire system is a prototype system that requires additional sensors. A specific steer-by-wire system has been developed that ensures steering intervention with a very high reliability. Its system architecture is presented and described in detail in section 4.3 above (challenge 2.3). The commanded steering values generated by the Joint System are transformed onto the FlexRay and XCC architecture (X-by-wire Control Computer, see section 4.1 above), which are working in a completely redundant mode, including hardware components and software processes.

When the system operates in steer-by-wire mode, the safety clutch is open, thus releasing the mechanical connection between the front axle and the steering wheel. Haptic feedback to the driver is generated via the driver feedback actuator while the steering task is performed by the servo steering actuator. In case of a failure in one of these actuators, the clutch can be engaged in order to restore the mechanical connection between the steering wheel and the front axle and therefore allow the driver to control the vehicle via a conventional steering system. Each of the two redundant XCC control units can command the engagement of the clutch. As an inherent safety feature, the clutch closes also automatically by means of a permanent magnet whenever the power supply is cut. This ensures that the system will enter a safe state (unassisted mechanical steering) even in case of total loss of electric power.

CSC-ECUs

The CSC electronic control unit described in section 4.1 above provides scalable microcontroller performance in order to implement different vehicle functions concerning chassis control, driver assistance systems as well as safety functions. Communication is provided optionally using CAN or/and optionally FlexRay. The Joint System demonstrator is working in its final configuration with three CSC ECUs included in a CAN network. On these CSC-ECUs three software components of the co-system as part of the Joint System will be executed:

- Driver State Assessment
- Mode Selection and Arbitration Unit
- Command and Haptic Feedback Generation

For a description of these modules, please refer to sections 5.2 and 5.3 above.

6.1.2 System validation by important use cases

At the beginning of the HAVEit project a use case catalogue was developed that covers all use cases that should be in the focus of this project. The use cases do not cover all possible situations of driving a highly automated vehicle but were selected to deal with the most important situations still manageable for the realization and testing. Scenarios are defined in HAVEit as a sequence of use cases, instantiated in a simulator or test ground. An overview of the HAVEit use case classes can be found for example in the deliverable D33.3. In a first step towards system validation in a real vehicle, some of the most important use case classes were selected to be tested. Having validated the Joint System in these selected use cases, the Joint System concept was transferred to the vertical vehicle applications (AMD, ARC, AQuA, TAP and AGD).

In each of the following sections, one of the selected use case classes is addressed and its validation in a test vehicle is discribed and discussed. Each section is structured analogously: First, it contains a brief description of the addressed use case class and its concrete application in the test run. Second, to illustrate each use case validation a short video sequence is shown and discussed. Further, diagrams are included to detail the most important aspects of each use case, such as the planned and driven path during a lane change or the measured distances to an object. Each section concludes with a short summary.

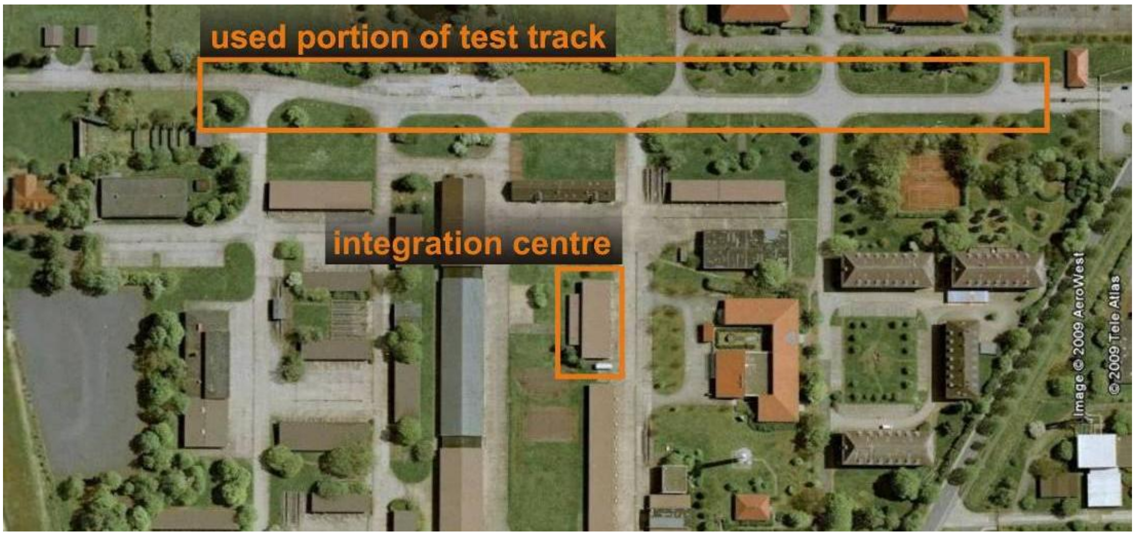
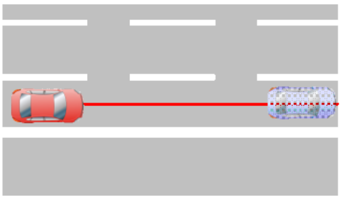


Figure 148: Test site in deserted military base near Braunschweig

The tests were performed on a test site in the deserted military base in the vicinity of Braunschweig, shown in Figure 148: The test track has some shortcomings, one of the most prominent being the middle lane marking which is a solid and not a dashed line. To overcome this drawback, the middle lane markings had to be artificially set to dashed in the software. In addition, there are only very small safety areas to the sides of the used portion of the test track. Thus, only very low velocities could be tested safely.

Use case "Driving and detected obstacle in Highly Automated"



The use case class Driving and Detected Obstacle (see deliverable D33.3) deals with a deceleration and braking situation triggered by a detected obstacle in front of the ego vehicle.

As the most challenging sub-case, driving in automation level Highly Automated was tested. In the test the ego vehicle is driving on the right lane of a two-lane road at a velocity of approximately 5.5 m/s. Then an obstacle appears on the same lane. The car automatically stops right behind the obstacle, while the driver is driving hands-off and feet-off.

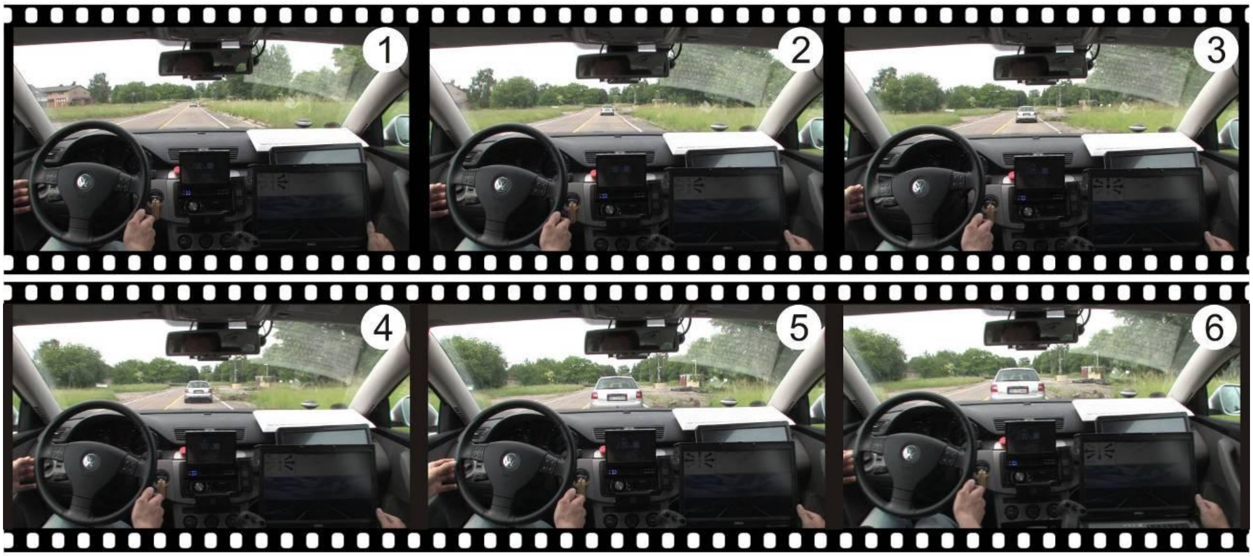


Figure 149: Driving and detected obstacle (HA): Video sequence

Frames 1-6 of the video sequence in Figure 149 show how the ego vehicle was driving in the automation level Highly Automated in the right lane while approaching an obstacle ahead. The ego vehicle maintained the desired velocity and compensated lateral deviations. When the obstacle came closer as in frame 4 to 5, it was necessary either to brake or to brake and change the lane to the left, which can be seen also in the computed trajectories and in the action grid in Figure 151. Since the driver did not initiate a lane change, the co-system braked and came to a stop after frame 6. The planned and driven velocity profiles are shown in Figure 150.

Figure 150 shows the performance of the longitudinal controller in terms of the velocity control. In this figure the reference velocity to be followed by the vehicle demonstrator, blue curve, is computed from the planned velocity profile by taking the velocity of the closest point of the planned trajectory to the vehicle at each longitudinal controller cycle. For this reason, the profile of the reference velocity is not continuous. The red line in Figure 150 signifies the actual vehicle velocity that is controlled by the longitudinal controller. We can see that the actual longitudinal controller has a significant time reaction but the whole system is able to stop the vehicle sufficiently behind the obstacle (10m of distance). However, we should take a deeper look at this situation to tune the longitudinal controller to enhance the accuracy of the controller while at the same time preserving the passengers' comfort.

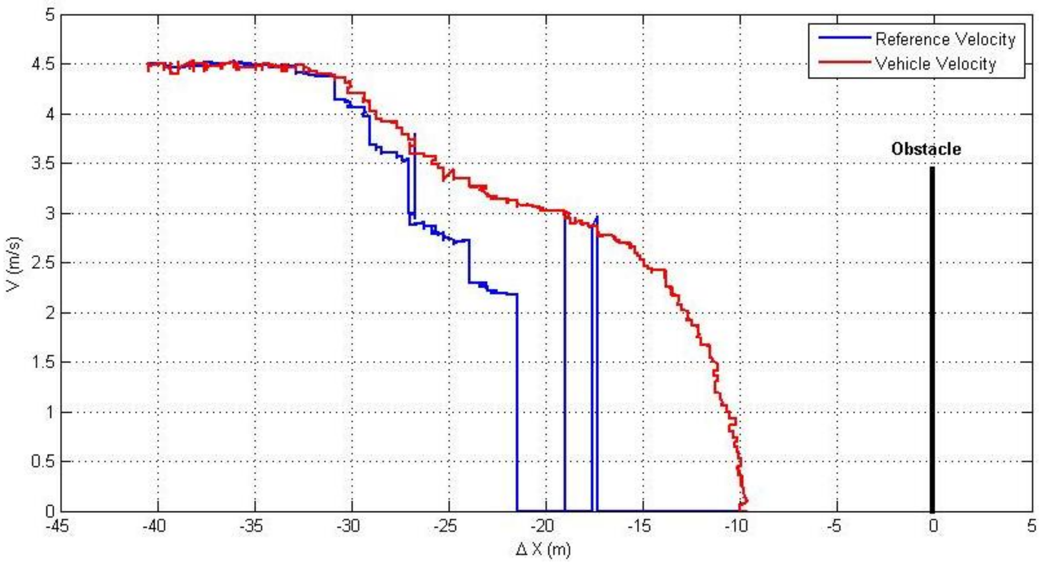


Figure 150: Driving and detected obstacle (HA): Planned and real velocity

Figure 151 illustrates the manoeuvre tree (grid on the upper left and right) and the planned trajectories; further, it provides a rough representation of the environment (bottom part). It shows that the standing lead vehicle has been detected by the laser scanners and the road has been correctly observed by the lane camera. The manoeuvre tree shows the currently performed manoeuvre "FollowVehicle". Three future manoeuvres are available with different valentials: "EmergencyBrake", "MinimumRisk", and "ChangeLaneLeft". The manoeuvre grid depicts the options to either brake or to change lane to the left and brake.

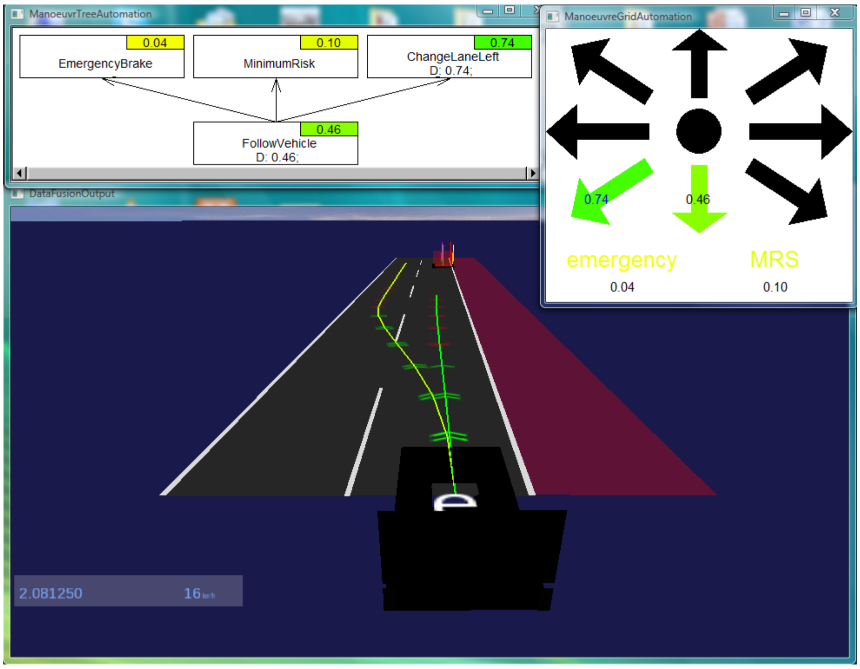
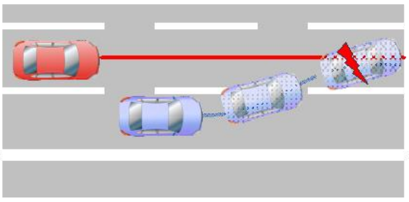


Figure 151: Driving and detected obstacle (HA): Planned trajectories, manoeuvre tree, and manoeuvre grid

For the selected manoeuvres, detailed trajectories are calculated. In case of a lane change, the corresponding trajectory indicates acceleration and some deceleration later on, whereas the trajectory for the "FollowVehicle" manoeuvre indicates to first accelerate and then brake stronger behind the obstacle. As observed in the video sequence in Figure 149, this trajectory is executed, finally coming to a stop behind the stopped lead vehicle.

The use case "Driving and detected obstacle" has been successfully validated in the FASCar. Switching between different automation levels was possible. In the automation level Highly Automated, the vehicle successfully drove in its lane and stopped behind a standing lead vehicle. Further tests are planned to demonstrate the same use case at higher velocities.

Use case "Driving and detected obstacle – emergency brake in Driver Assisted"



In contrast to the previous chapter, which deals with normal driving and an obstacle detected far enough ahead, this use case of the use case class "Driving and detected obstacle" deals with an emergency situation, which is triggered by an obstacle suddenly appearing in front of the ego vehicle.

In this section, the validation of the use case in automation level Driver Assisted is described, at which the reaction of the automation in automation level Driver Assisted was tested. The expected result was that the automation intercedes by an emergency braking manoeuvre only, if the driver does not react in time to the sudden obstacle. In the test scenario, it was sufficient for that purpose to replace the suddenly appearing dynamic obstacle by a virtual static obstacle at a fixed GPS position. The FASCar was driven on the right lane of a two-lane road in Driver Assisted at about 11 m/s. The virtual static obstacle was programmed to stand on the right lane and - for visualization purposes - additionally marked on the lane by traffic cones. When the driver did not react to resolve the situation, the automation level Emergency was activated and an emergency braking manoeuvre was performed to mitigate the collision.

As illustrated in Figure 152, the vehicle was driving in automation level Driver Assisted and approached a static obstacle (frame 2). As the distance to the obstacle decreased below a safe threshold, the HMI issued a warning to the driver (frame 3) and activated the automation level Emergency as can be seen in the display shown in frame 4. Thus, the vehicle performed an automatic emergency braking manoeuvre to avoid a collision (frame 5). Once the vehicle came to a stand still, the warning shut off and control was given back to the driver in automation level Driver Assisted (frame 6).

In the displayed video sequence, a virtual obstacle was programmed at a fixed GPS position, such that the traffic cones were located about 30cm after the rear bumper of the virtual obstacle. As can be seen by the small distance to the cones in frame 5, the final distance of the ego vehicle to the obstacle was approximately zero, such that there probably was already some contact at very low velocity.

Figure 153 shows the actual and the planned velocity profile of the vehicle demonstrator versus the distance separating the vehicle demonstrator and the obstacle. In this test, the driver controls the longitudinal velocity by himself with the assistance of the co-pilot. In the beginning, the system does not detect the obstacle and the planned velocity correlates with the driven speed. Once the obstacle is detected (at -60m distance), the co-pilot plans to slow-down (velocity profile blue) until it requests to stop the vehicle, but the driver ignores this suggestion (as can be seen by the gap between the planned velocity dropping to zero (at distance of -21m) and the initiation of the emergency brake). When the situation becomes critical, the emergency braking is triggered by the system and the vehicle is stopped.

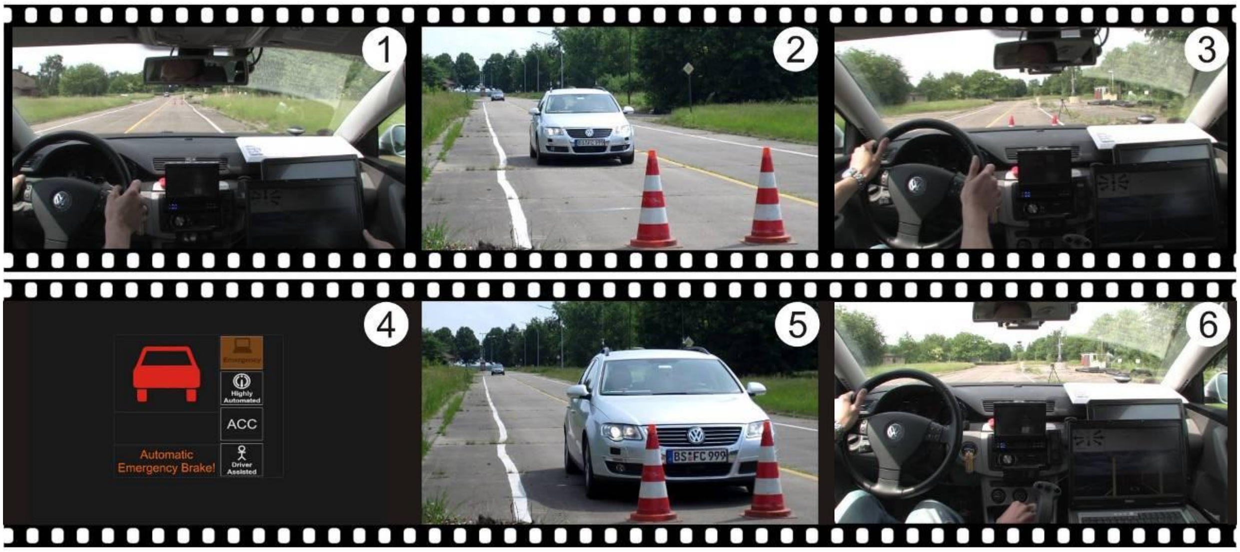


Figure 152: Driving and detected obstacle - emergency brake in DA: Video sequence

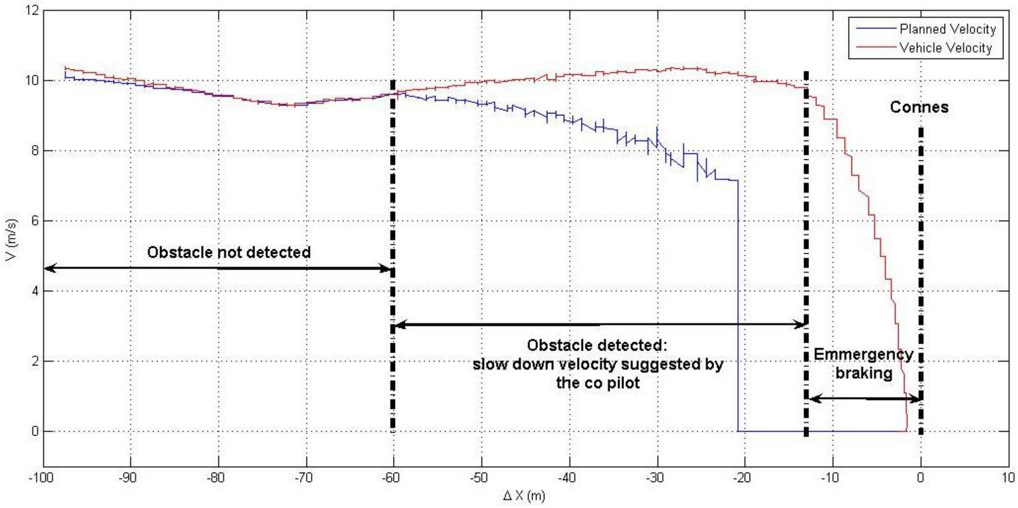
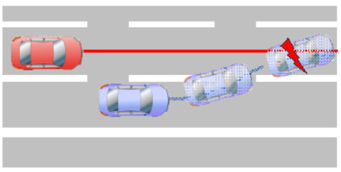


Figure 153: Driving and detected obstacle - emergency brake in DA: Planned and real velocity

The use case "Driving and detected obstacle - emergency brake in Driver Assisted" has been validated successfully. As shown, the vehicle performs an automatic emergency brake to mitigate rear-end collisions even in automation level Driver Assisted.

Use case "Driving and detected obstacle – emergency brake in Highly Automated"



As described before, this use case is derived from the use case class "Driving and detected obstacle" deals with an emergency situation triggered by a detected obstacle in front of the ego vehicle. The emergency interaction can be applied to all automation levels.

Regarding the validation of this use case class, in this section the reaction of the automation in automation level Highly Automated is tested. The FASCar drives autonomously on the left lane of a two-lane road in Highly Automated at about 2.5 m/s (for safety reasons). A team member suddenly holds an obstacle in front of the vehicle. In consequence, the automation level Emergency is activated and an emergency brake is performed to avoid the collision.

As illustrated in Figure 154, the ego vehicle is driving autonomously in automation level Highly Automated (frame 1, 2) in the left lane. On the left road side there is an obstacle (frame 3) which suddenly moves into the lane directly in front of the vehicle (frame 4, time 0.6s in Figure 155). As shown in frame 5, the obstacle is detected and the automation level Emergency is activated, because of the high criticality of the situation which could not be resolved by comfortable braking. Therefore, an automatic emergency brake is performed and the vehicle comes to a complete stop (frame 6).

Figure 155 shows the relative distance and velocity of the tracked obstacle that appears suddenly in front of the ego vehicle as determined by the data fusion component. It can be seen that the object is detected steadily while the ego vehicle decelerates (relative velocity is negative and finally becomes approximately zero).

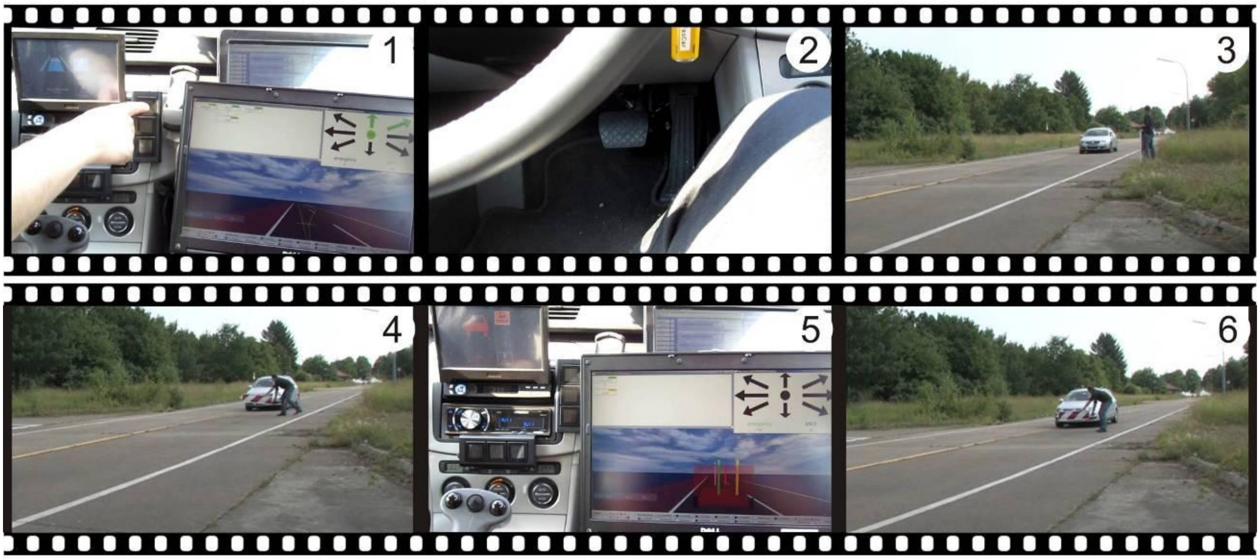


Figure 154: Driving and detected obstacle - emergency brake in HA: Video sequence

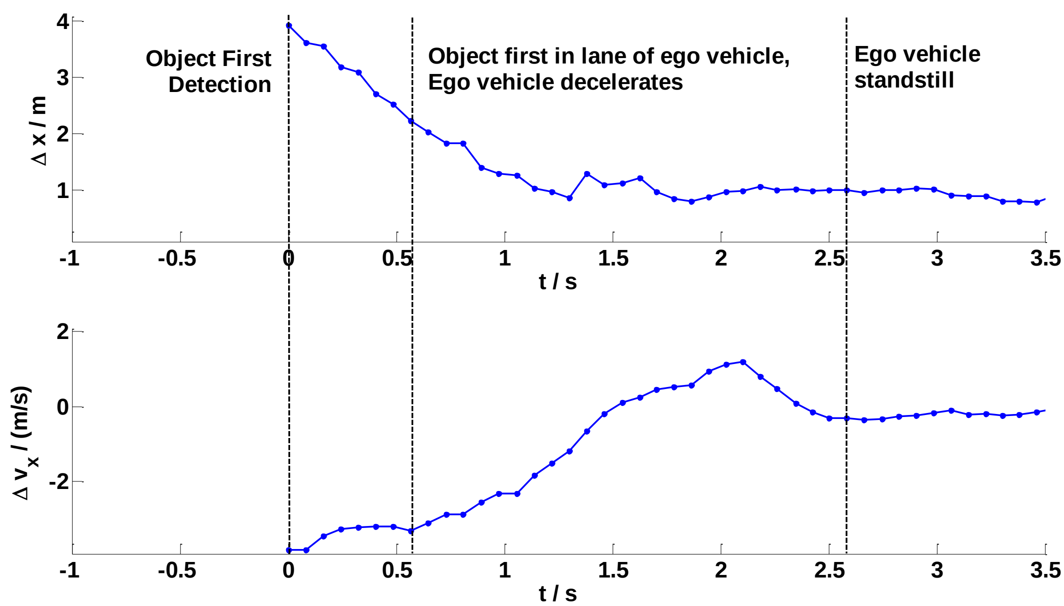
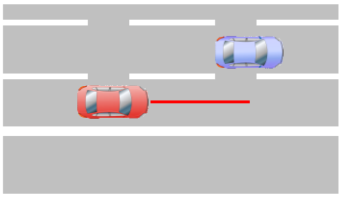


Figure 155: Driving and detected obstacle - emergency brake in HA: Relative distance and velocity of obstacle as determined by data fusion component

When an unexpected obstacle (a pedestrian for example) appears it usually comes from outside the road. The data fusion tracking component tracks the object even if this is outside of the road, but it doesn't report it to the other components. As soon as the object moves in the road it is included in the output of the data fusion, so the Co-Pilot and control components can take proper action. As shown in Figure 155, immediately after the object has entered the lane, the automation level Emergency is activated and an automatic braking is performed, which can be seen by the fact that the relative longitudinal velocity is decaying to zero, avoiding the collision with the obstacle.

The use case "Driving and unexpected, detected obstacle in front (HA)" has been validated successfully. The vehicle stopped automatically in front of a suddenly appearing obstacle to avoid the collision.

Use case "Driving in a lane avoiding right overtaking"



The use case class "Driving in a lane avoiding right overtaking" covers a specific driving situation, where the ego vehicle is driving on the middle or right lane and a slower vehicle is on the left. The reaction of the ego vehicle depends on the technical capabilities and can reach from no reaction, to some kind of information for the driver up to an automatic avoidance of such situations.

For the demonstrator tested here, the expected behaviour is that the automation does not overtake the other vehicle.

The use case was tested in the FASCar on a two-lane road, where the FASCar drove in automation level Highly Automated. It followed the right lane at initially approximately 5.5 m/s. A slower vehicle ahead on the left lane drove first at about 2.7 m/s, then accelerated up to 7 m/s and finally stopped on the left lane. It should not be passed on the right at any time (despite the low velocities, which only were due to restrictions of the test track).

As shown in the video sequence in Figure 156, the demonstrator vehicle is driving in Highly Automated mode on the right lane. As displayed in frame 2, there is a slower vehicle ahead in the left lane. The automation recognizes the situation and - in order to avoid right overtaking - offers either to change the lane to the left or to slow down, which is executed (frame 3). Once the demonstrator vehicle is driving at the same speed as the vehicle ahead, the planned trajectories offer to follow the lane or to change to the left lane (frame 4). As the vehicle on the left lane accelerates slightly, the ego vehicle matches this speed up to 5.5 m/s on the right lane. Finally, the vehicle on the left lane brakes to a standstill and the ego vehicle also to avoid right overtaking (frames 5, 6).

Figure 157 shows the situation where the ego vehicle currently performs the manoeuvre "FollowLane" and the vehicle ahead on the left lane has already been detected by the laser scanners. Optionally available manoeuvres are "EmergencyBrake", "minimumRisk" and "ChangeLaneLeft". The computed trajectories indicate that in case a lane change was selected, the vehicle should first accelerate briefly and then decelerate to follow the other vehicle in the left lane. In case the current manoeuvre "FollowLane" is pursued further, the corresponding trajectory indicates that immediate braking is necessary in order to avoid right overtaking

The use case "Driving in a lane avoiding right overtaking" has been demonstrated successfully. The manoeuvre planning creates appropriate manoeuvres and trajectories to avoid right overtaking and in Highly Automated the vehicle follows the planned trajectories well. Some adaptations are still necessary in order allow right overtaking at very low speeds according to the traffic rules, such that the road is blocked by a standing or very slowly moving vehicle on the left lane.

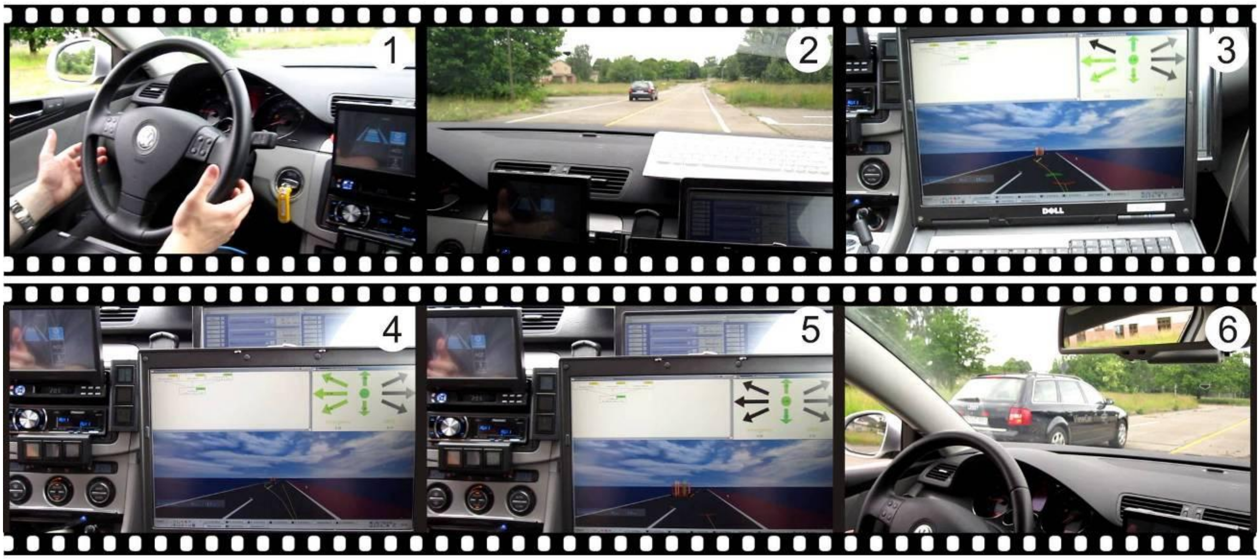


Figure 156: Driving in a lane avoiding right overtaking: Video sequence

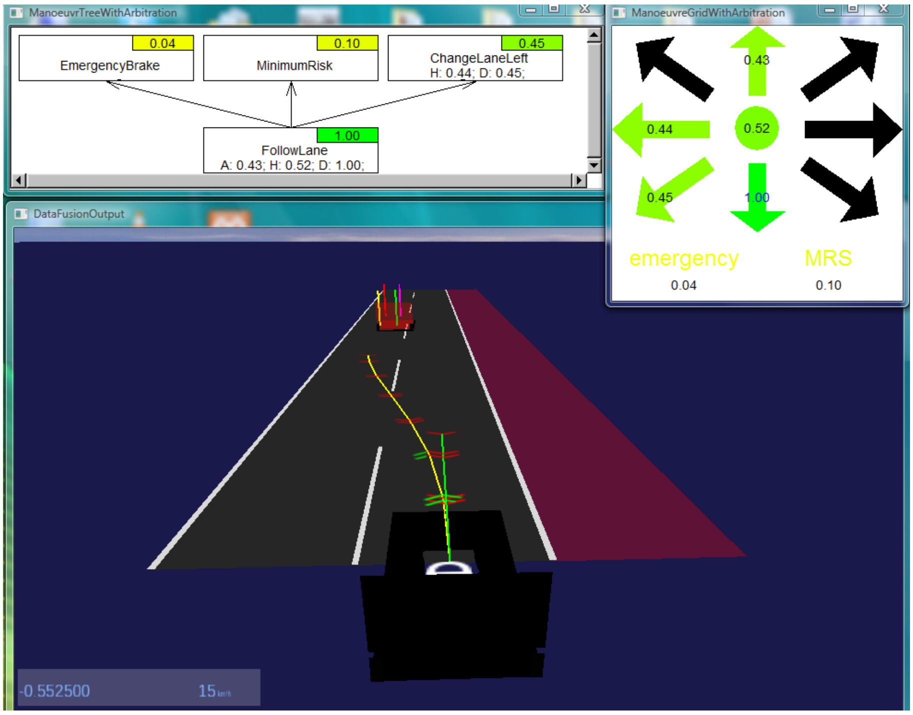
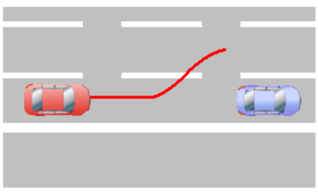


Figure 157: Driving in a lane avoiding right overtaking: Planned trajectories, manoeuvre tree and grid

Use case "Driving and lane change"



The use case class "Driving and lane change" covers a lane change manoeuvre in different automation levels. In lower automation levels the driver is supported e.g. by a blind-spot warning, in highly automated driving there is the possibility of a manual lane change or a highly automated lane change that is either initiated by the driver or done automatically.

With regards to the validation of this use case in the FASCar, a driver initiated lane change in automation level Highly Automated is tested. A standing lead vehicle on the right lane of a two-lane road was passed by a lane change to the left lane. The FASCar was driving autonomously in Highly Automated mode at a speed of about 10 m/s.

The video sequence in Figure 158 shows how the demonstrator vehicle approaches a standing lead vehicle in the right lane. The demonstrator vehicle drives in automation level Highly Automated. As the distance decreases, a lane change is suggested in the display shown in frame 2, which the driver activates by pushing the indicator button to the left, (frame 3). Immediately, the vehicle initiates a lane change (see frames 3 and 4) and then passes the standing vehicle (frames 5, 6) on the left lane.

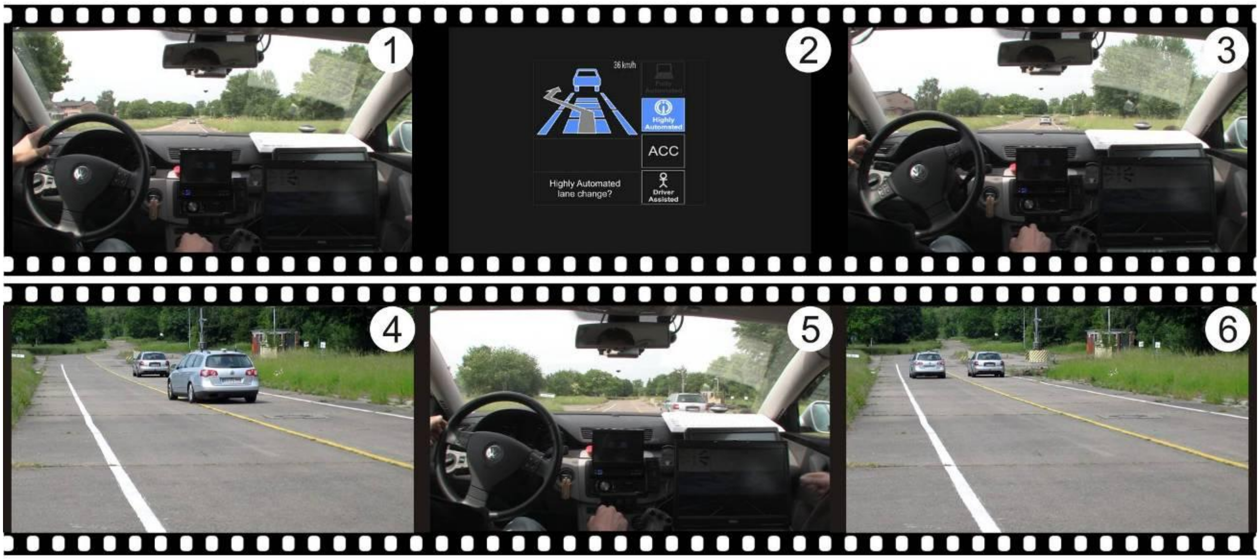


Figure 158: Driving and lane change: Video sequence

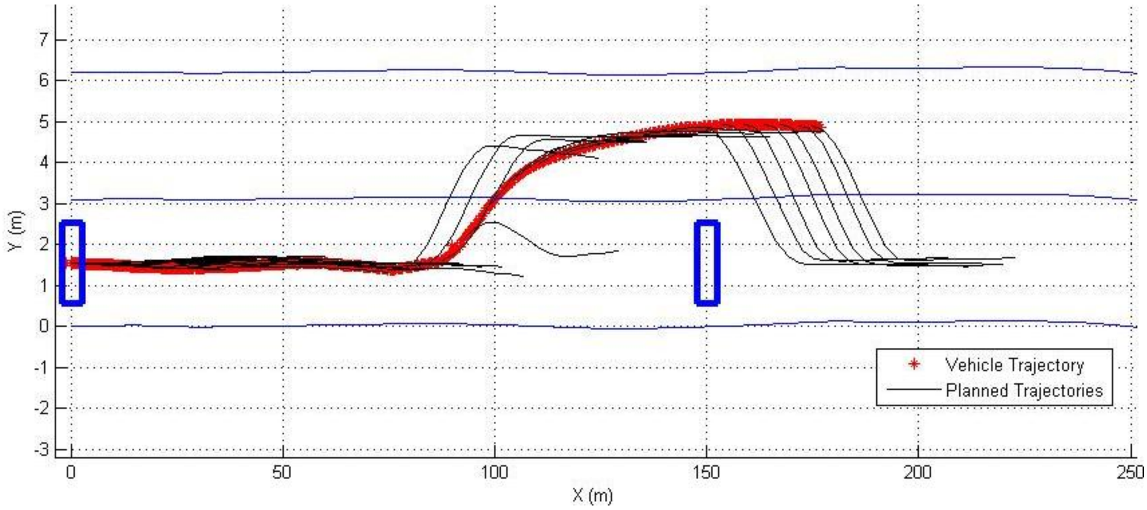


Figure 159: Driving and lane change: Planned and driven trajectory

Figure 159 shows the behaviour of the vehicle with respect to the planned trajectories. The two boxes show the initial position of vehicle demonstrator and the obstacle placed in the right lane, ahead of the initial position of the demonstrator. Note that the axes are scaled differently to allow observation of more details. At the beginning of this plot, the planned trajectories keep the vehicle in the middle of the lane. When the system detects the obstacle, the planned trajectories are deformed in order to bring the vehicle to the adjacent lane to the left to avoid the collision to the obstacle. When the obstacle is passed, the planned trajectories try to bring the vehicle again to the right lane (see at the end of the plot) but this lane change again requires confirmation of the driver (as the driver did not react, the vehicle did not change to the right lane again). The use case “Driving and lane change” has been validated successfully.

Use case “Driving and activation not possible”

This use case class is about situations in which the driver wants to activate a specific automation level but this transition is refused due to unfulfilled preconditions, e.g. driving too fast or not on a highway. Examples that are covered are the refused transition from Driver Assisted to Semi-Automated or to Highly Automated as well as the refused transition from Semi-Automated to Highly Automated.

In order to validate this use case class in the FASCar, the precondition of maximum speed for automation level Highly Automated was manually decreased in order to make this automation level unavailable during the test. In a later product, the maximum speed allowed for level Highly Automated of course might be higher. The driver tried to switch from Driver Assisted to Highly Automated, which was refused. Then the driver took over by selecting a lower automation level again. Meanwhile, the driver was following the right lane of a two-lane road at approximately 6.0 m/s.

Figure 160 illustrates the HMI display when the activation of the automation level Highly Automated is impossible. First, the demonstrator vehicle is in Driver Assisted (frame 1). Subsequently, the driver switches to automation level Semi Automated (frame 2). When trying to activate Highly Automated (frame 3) the display indicates the refused transition by a large orange “X” and symbols to communicate the driver that he must remain in control. After this initial warning, the symbols below the “X” are replaced by additional information regarding the reason for the refusal (frame 4) – in this case that the speed is too high. In frames 5 and 6 it can be seen how the driver switches back down to Semi Automated and finally Driver Assisted.

The use case “Driving and activation of automation level impossible” has been validated successfully for the case that the speed was too high to activate Highly Automated. The speed limit for this refusal was reduced for this purpose. All visual HMI components worked properly.

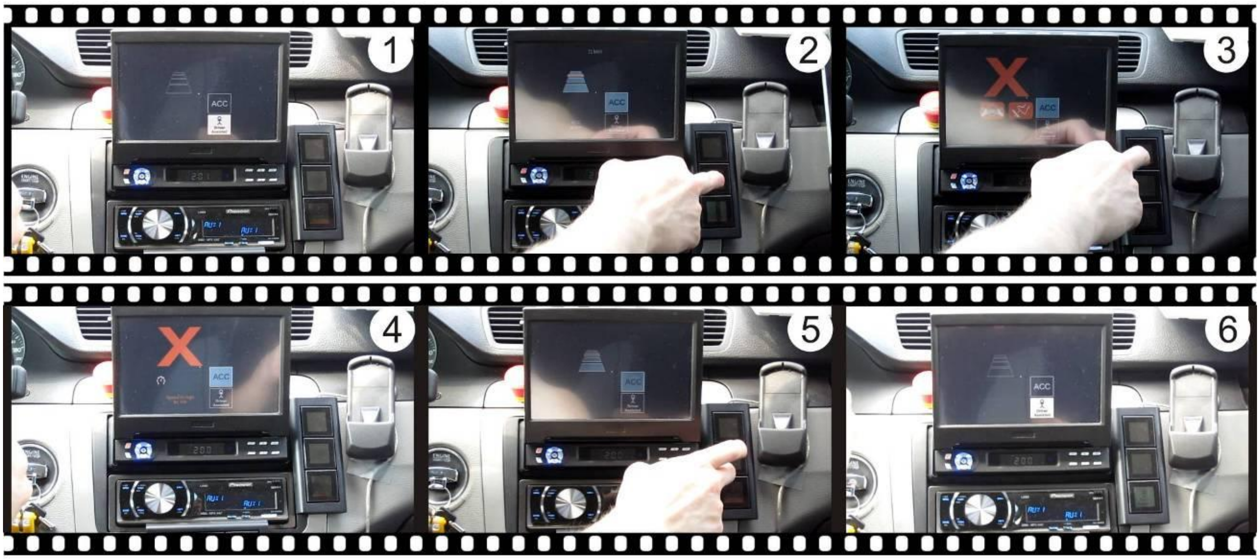
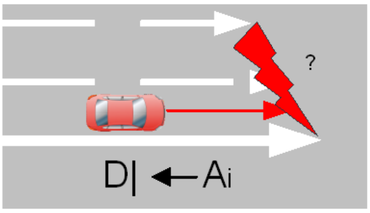


Figure 160: Driving and activation not possible: Video sequence

Use case "Driving, driver unresponsive and transition to MRM"



This use case class is about a situation in which the driver does not take over control as requested by the system. After a warning period the co-system starts a manoeuvre that is the manoeuvre with the minimum risk in that specific situation to avoid serious damage.

To validate this use case class in the FASCar, the vehicle was following the left lane of a two-lane road in automation level Highly Automated at approximately 6.0 m/s. This automation level was then manually made unavailable, which triggered an escalating take over request to the driver and ultimately led to a transition to the MRM (minimum risk manoeuvre). During the MRM the FASCar changed to the right lane, slowed down and then stopped completely.

Figure 161 shows the necessary deactivation of the automation level Highly Automated. First the demonstrator vehicle is driving in Highly Automated (frame 1). Then this level is no longer available and a take over request is issued, as shown on the display in frame 2 and 3. The take over request is escalating by blinking with increasing frequency. Since the driver does not react on the take over request, the minimum risk manoeuvre (MRM) is triggered, as shown in frames 4 and 5, and the field for Driver Assisted is still blinking to indicate the automation level requested by the take over request. The MRM is executed and the vehicle changes to the right lane, decelerates and finally comes to a complete stop. At this point, the driver takes over which terminates the MRM and the automation level Driver Assisted is activated (frame 6).

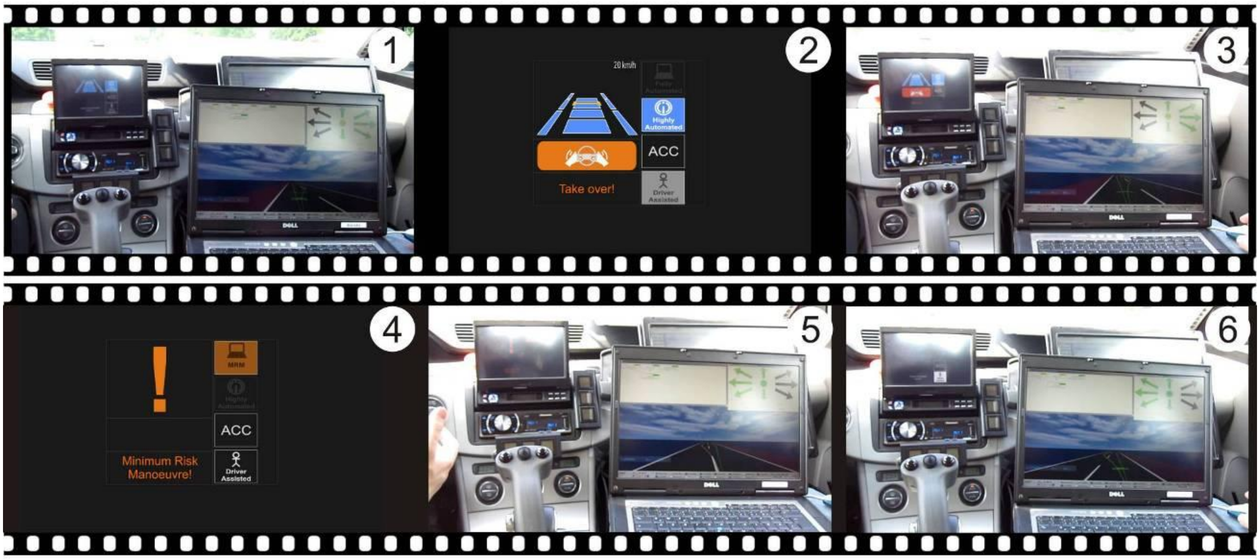


Figure 161: Driving, driver unresponsive and transition to MRM: Video sequence

Figure 162 shows the planned (thin black lines) and actually driven trajectories (red stars) for this use case. The actual positions plotted in the figure stem from the recorded GPS values. At some points, the GPS data included jumps and other irregularities. These errors, however, only show up in the above plot and have no influence on the actual performance of the demonstrator vehicle, since the GPS measurements are not used by the system and only exploited for the visualisation of the use case in this diagram. At the beginning of Figure 162, lane keeping is requested by the system and the co-pilot plans the adequate trajectories for this manoeuvre. The vehicle follows these plans and follows the left lane. After some time the MRM manoeuvre is triggered because the driver fails to comply with the take over request. The system commands the vehicle to change a lane to the right and then to decelerate to standstill.

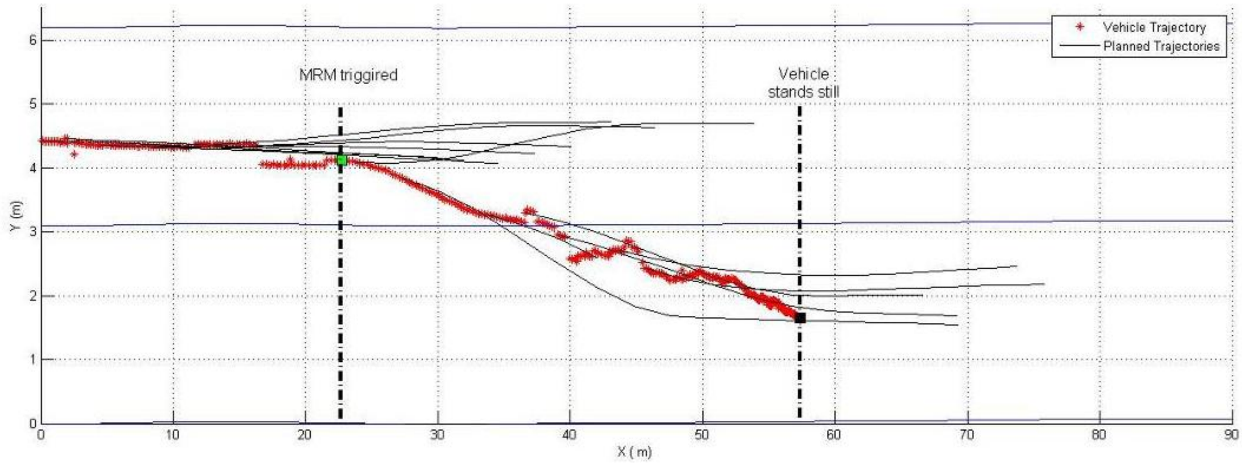


Figure 162: Driving, driver unresponsive and transition to MRM: Planned and driven trajectories

In Figure 163 we see the profiles of the reference velocity and the real vehicle velocity. In a normal driving scenario, the reference is planned to reach its desired value and the vehicle tracks this velocity profile very closely. After triggering the MRM manoeuvre, the reference velocity is planned in a way to bring the vehicle to standstill. We observe that in the deceleration phase, the vehicle only follows the reference profile with a certain delay. This is because we are using the same controller to control the acceleration and deceleration dynamics, although they are different because of different actuators used to affect these dynamics (engine to accelerate and brakes to decelerate).

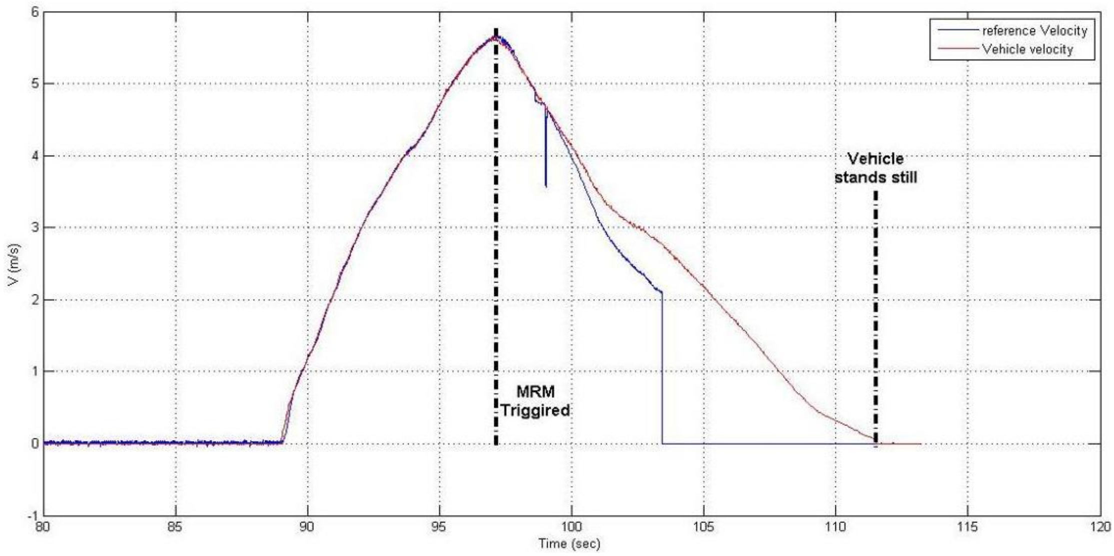


Figure 163: Driving, driver unresponsive and transition to MRM: Planned and real velocity

The use case “Driving and deactivation necessary (MRM)” has been validated successfully. The availability of the automation levels is determined automatically upon the fulfilment of the various preconditions.

6.1.3 Summary and conclusion

By means of several integration workshops held at DLR in Braunschweig, all Joint System components were integrated. Since the workshop in June 2010 all compentants are successfully working together in the test vehicle. During the validation the most relevant HAVEit use cases were tested in a real-world environment. The highest priority was assigned to the automation level Highly Automated because of the highest functional complexity of this automation level. Further, the same algorithms, functions and sensors are used as well in the other normal automation levels Driver Assisted and Semi Automated, even though in these levels the support of the driver is reduced compared to Highly Automated.

The measurement results presented above in the respective use cases show that – despite several improvements that were implemented during the final project phase - the Joint System is able to handle each of the use cases quite well.

- The environment sensors (lane camera and laser scanners) are available and deliver data which allow driving the relevant use cases.
- The Data Fusion component of the Co-System successfully generates a perception model and a vehicle state as input for the other algorithms within the Co-System.
- The Co-Pilot component as part of the Co-System interprets the output of the Data Fusion and chooses situation specifically the available manoeuvres for the automation. By the chosen manoeuvre the Co-Pilot then calculates trajectories for all detected lanes and for the MRM.
- The Mode Selection and Arbitration Unit as component of the Co-System manages the transitions between the automation levels and controls the visual HMI.
- The Command and Haptic Feedback Generation of the Co-System follows the best trajectory generated by the Co-Pilot. In the automation level Highly Automated the driver is able to remove his hands from the steering wheel. Alternatively, the driver is able to keep his hands on the steering wheel to feel the lateral automation behaviour in Highly Automated. The driver is also able to influence the automation behaviour in HA via the steering wheel.

The overall concept and architecture hold up nicely to its expectations. The transfer of know-how and algorithm from the Joint System demonstrator described here to other HAVEit demonstrators has taken place successfully. Overall, the work package “Joint System” of the HAVEit project has reached its goals and challenge 4.1 has been reached. With the successful integration of the Joint System into the demonstrator described in this section, we conclude that highly automated driving is no longer a theoretical vision, but can be experienced beyond simulators also in real vehicles.

6.2 Challenge 4.2: Brake-by-Wire Truck

HAVEit challenge 4.2 aims at the development and pre-homologation of a novel brake-by-wire system for trucks. Instead of conventional pneumatic actuators electro-mechanical ones are to be used which are operated in a 2E architecture. Overall objective of the new system is shortening the brake distance of the truck by about 15% compared to state-of-the-art solutions. The fully redundant 2E architecture is described earlier in this report. The new brake actuators developed in this project are presented in section 4.3.

This section summarizes the brake-by-wire system from the system’s point of view including the setup of the demonstrator and presents some measurement results.

Components installed

The demonstrator vehicle is a Volvo FH12 4x2 truck (see Figure 164) which has been equipped with:

- HMI components
  - Electric brake pedal
  - Electric parking brake switch
  - I/O module for auxiliary control
  - Diagnostics displayed on the dashboard from the brake system, via J1587
  - Re-use of existing steering angle sensor for stability control
- System Control Components
  - System Control Module (SCM), 2x
    - Power Distribution Module (PDM)
    - ESP module
    - Star Coupler
    - J1587 Diagnostics Module
  - XCC, 2x, delivered by USTUTT
  - J1939 Gateway delivered by Explinovo
- Actuators
  - Electric Brake Actuators (EBA), 4x, for service and parking brake.

Figure 164: WP4200 Demonstrator, Volvo FH12 4x2 Truck

6.2.1 System description

Main features

The Haldex EMB System is an electrically controlled brake system for trucks, intended for actuation of electromechanical disc brakes that control the clamp forces by an electric motor. The system communication is established via:

- FlexRay between the SCMs and the EBAs and between the SCMs and the XCCs
- LIN between the SCMs and the HMIs
- CAN between the gateway and the engine management of the vehicle
- J1587 for distribution of information from the SCM´s to the dashboard

The system has a redundancy in that it has four brake pedal sensors, four sensors on the parking brake handle and there are two communication circuits and two power circuits. See the overall architecture sketched in Figure 165.

The system provides the basic brake system functions such as service braking and parking brake, and automatic blending of brake force from retarders. Further, it has functions for preventing wheel lock during service braking or engine braking or excessive wheel spin during acceleration. There are also directionally stabilizing functions implemented preventing under- and over- steering.

The system interfaces and the interconnections between the different system components and the signal flow have been extensive parts of the integration work and are summarized below. The Brake by Wire demonstrator is built on the HAVEit architecture which has been described in deliverable HAVEit deliverable D23.1. The system is built as an x-split between the front and rear axle for both energy distribution and communication.

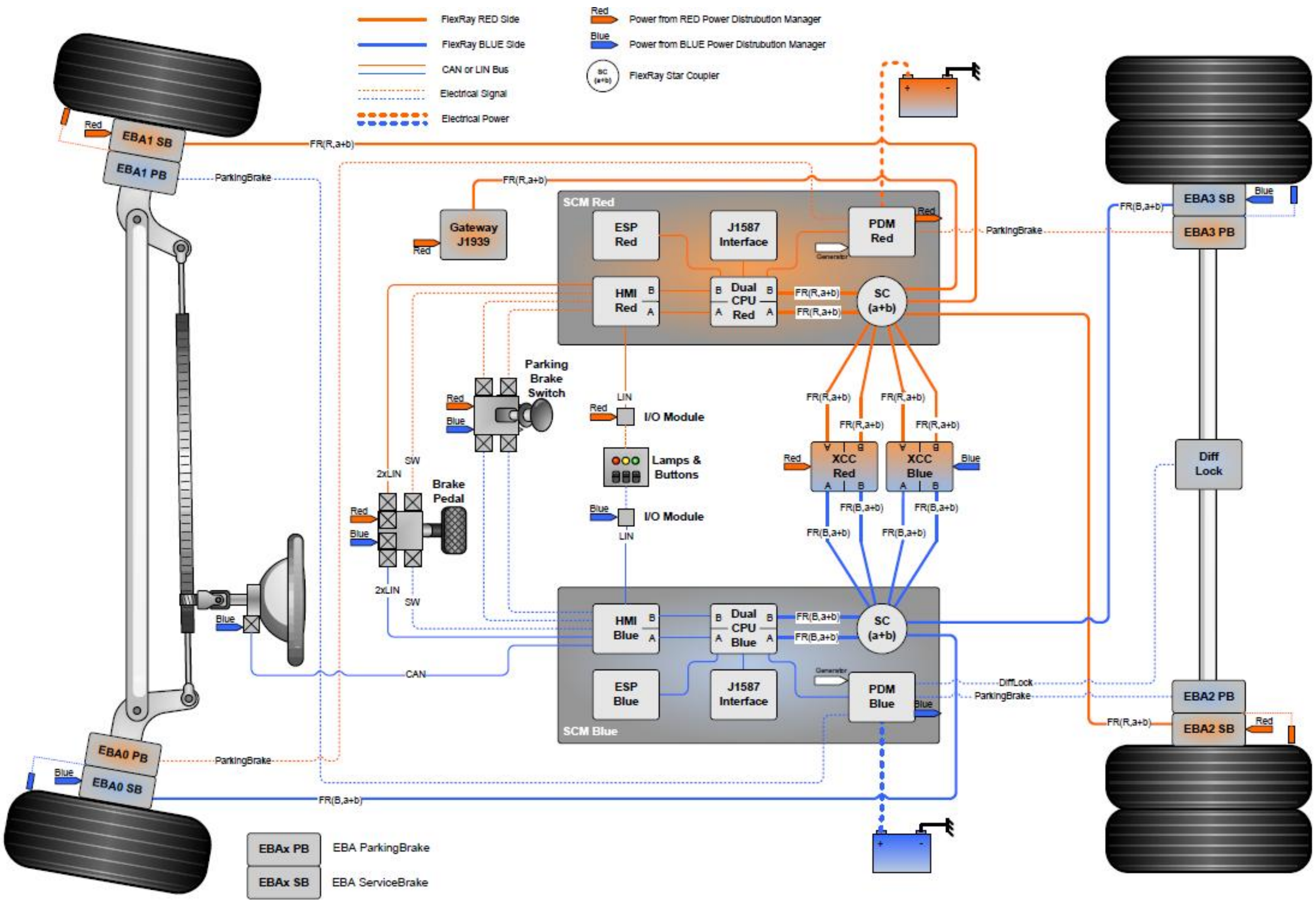


Figure 165: Brake-by-Wire architecture overview

Component functions

Electric brake actuators (EBA)

The system is intended for actuation of electro mechanical disc brakes which controls the clamp forces by an electric motor (see section 4.3 above for actuator details). The brakes are self-amplifying in a way that the inner brake pad is attached to a sledge that can move on a ramp (Figure 166).

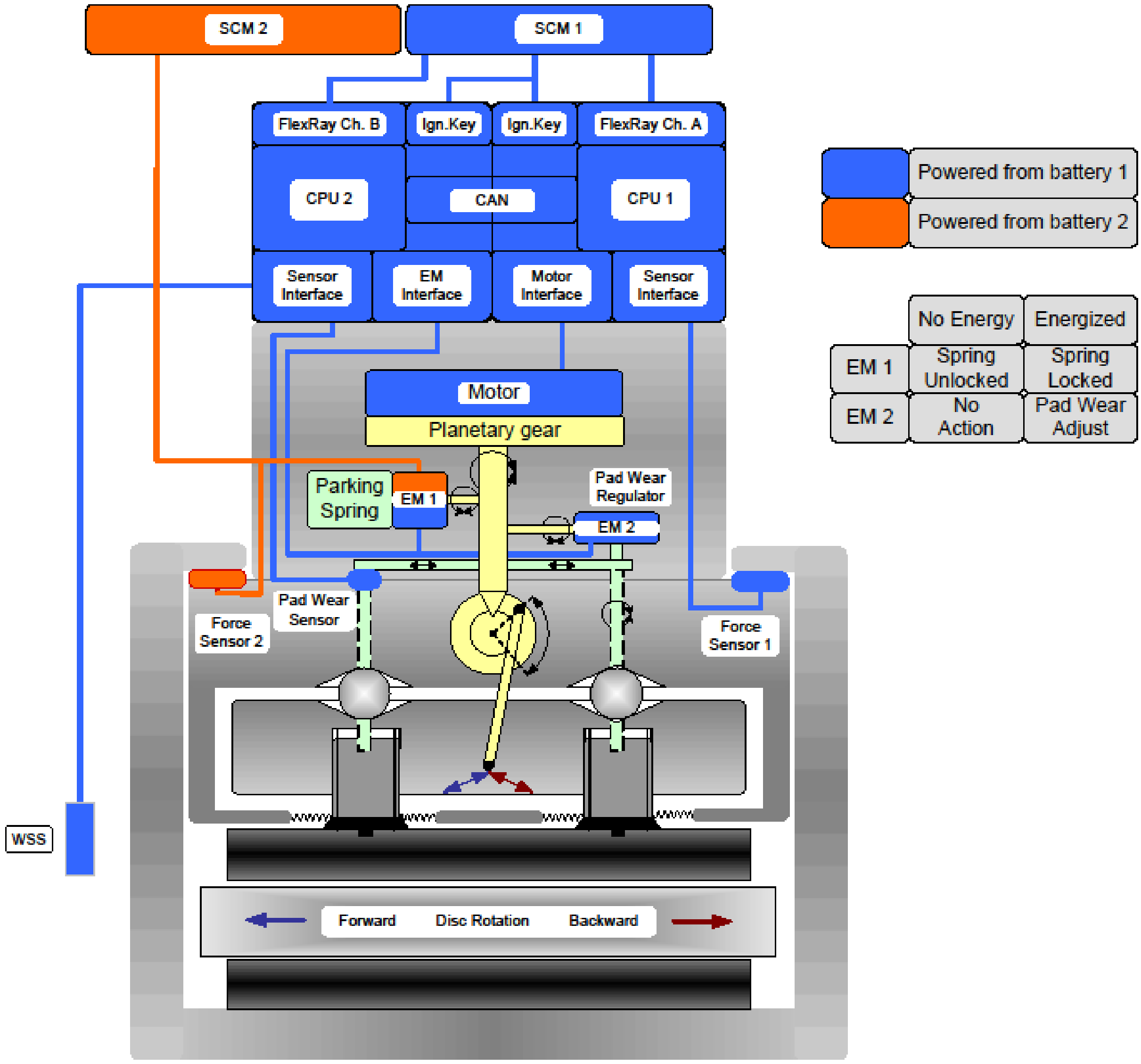


Figure 166: Principle scheme of the EBA belonging to the blue network circuit

The sledge is, via a mechanism, connected to an electric motor. In rest position the motor holds the sledge in its back position with the prescribed slack between pads and disc. When a brake torque command is received from the system, the motor drives the brake pad into contact with the disc and the pad-disc friction force will make the sledge to move on the ramp, helped by a roller. The ramp geometry is optimized for a certain pad-disc friction, which would require no torque from the motor to keep the clamp force steady. If the friction is higher than this, the brake becomes self locking and the motor must work for releasing the pad. The motor controls the sledge to give a certain clamp force corresponding to the command brake torque.

There are two force sensors for measuring the clamp force, one is connected to the brake ECU and the other is connected the system ECU (SCM). The brake has two separated processors for sensor signal processing and control, which can validate each other.

The electric motor is also used to adjust the slack between the pads and the disc. When the electric magnet (see EM 2) is energized the motor will be connected to two screws that can change the position of the pad towards the sledge.

A mechanical spring (parking spring) provides a power backup, when it is up-winded. This makes it possible to apply the brake even if there is no power available at all. The mechanism is used for generating the force for parking brake. The spring is held in winded position by an electro magnet (see EM 1) in normal run. When the electro magnet is released the spring will apply the pads in so-called parking brake mode. The spring is up-winded by the motor that controls the brake.

Wheel speed sensor

There is one Wheel Speed Sensor (WSS) connected to each brake actuator. It is a so called active sensor, consisting of hall sensors that measure the magnetic field and electronics for signal processing. A tooth wheel is mounted on the wheel hub and the sensors sense the teeth passing its surface when the wheel rotates. Each tooth passage generates two pulses and the wheel speed is derived from the time between those pulses. Each pulse contains coded information about the rotation direction and the status of the sensor. The sensor diagnostics reacts if the placement and assembly of the sensor is not correct such that the magnetic flux density variation is not adequate to provide a reliable speed signal. Further wheel speed signal processing is performed in the brake actuator CPU.

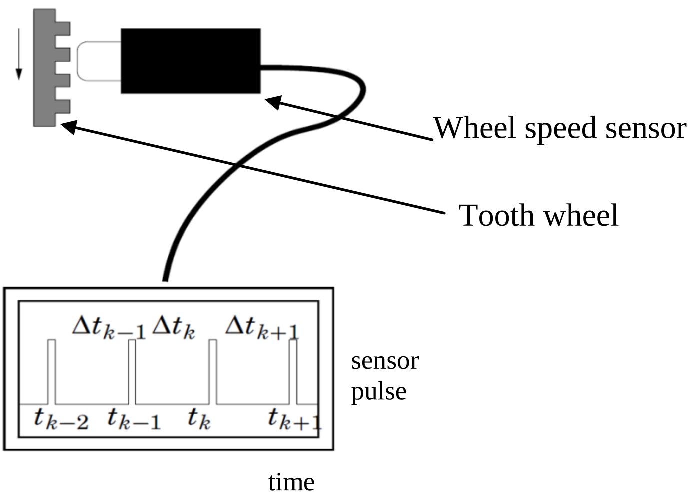


Figure 167: Function of wheel speed sensor. Each time a tooth flank passes the sensor it sends one pulse.

Steering angle sensor

The system includes a steering wheel sensor to detect the angular position of the steering wheel. The sensor transmits its measurements to the HMI-module in the SCM unit via CAN-protocol.

Figure 168: Steering angle sensor

J1939 Gateway

The J1939 gateway from Explinovo (see section 4.2 above for details) converts the messages that are sent to and received from vehicle from FlexRay to CAN. The CAN communication follows the Volvo specification for the current vehicle. The Microcontroller used for this application is a MC9s12XF512 with an internal FlexRay and CAN controller.

Electric brake pedal

The original brake pedal is kept in the truck, but the valve package is changed to a spring mechanism. The springs generate the reaction force towards the foot force from the driver. The sensor package, for measuring the stroke, includes four hall-sensors (Micronas HAL 2810[62]) that measure the distance to two magnets that moves with the pedal stroke. The sensors are two and two belonging to one of the two power- and network circuits within the brake system. The measurements are sent via LIN to the HMI module in respective SCM. There are also two switches mounted on the pedal to wake the system up such that service brake can be applied even when ignition is off.

Figure 169: Electric Brake pedal unit (left) and its installation on vehicle (right).

[62] Micronas, "Product information HAL 2810, Linear Hall-Effect with LIN Bus", 2008

Electric parking brake switch

The parking brake switch has four digital hall sensors to detect the position of the parking brake handle. As for the brake pedal, the sensors are two and two belonging to one of the two circuits in the brake system. The measurements are sent via LIN to respective SCM.

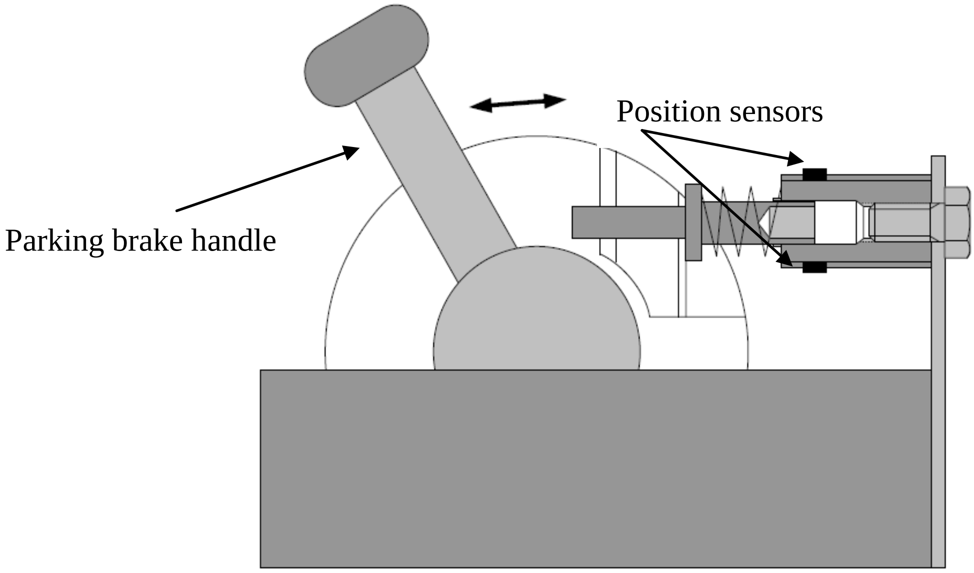


Figure 170: Sketch of the electric parking brake switch. The handle moves the magnet and the sensors (two showed on the picture) will sense its position (on/off).

Figure 171: Parking brake switch installed in driver cabin.

I/O module

The I/O module is connected to the buttons and control lamps on the dash board. It communicates with the HMI module in the SCM via LIN protocol, using the microcontroller UJA1023. The buttons and lamps connections are not redundant. They are connected only to one I/O module. The hill hold and terrain mode buttons are connected to the blue side and the diff-lock button to the red side.

XCC

The XCCs from USTUTT are the hubs for communication and have the main computational power. Each XCC includes contains the system application that controls the brake system actions. The XCC communicates with the aggregates, which are the PDM, HMI, ESP, the brakes and a vehicle system gateway using the FlexRay protocol. The XCCs also manage the interconnection between the blue and red circuits and issues related to redundant control. For more detailed information on the XCCs see section 4.1 above or deliverable D21.1, section 3.

System control module (SCM)

There are two system control units in the system. Each of them consists of three virtual sub modules. One power distribution module (PDM), one sensor module (ESP) and a module that controls the human machine interface (HMI). The PDM is the hub in the power network and assures that each brake in the circuit gets the power it needs, either from the circuit battery or, if this has failed, from the vehicle battery. The PDM is also connected to the force sensor and electro magnet of the brakes of transverse network circuit. The ESP sensor cluster consists of accelerometer sensors for the x, y, and z directions and a gyro for sensing yaw rate, roll rate and pitch rate. The HMI module processes the commands from the driver via the brake pedal, the park brake handle and the I/O Modules.

Physically the SCM unit contains one printed circuit board for the processors, two MPC5643L[63] and FlexRay connections. It has one further circuit board for the power control components of the PDM module. There are also two 12V batteries with a capacity of 10Ah in each SCM. See figures below.

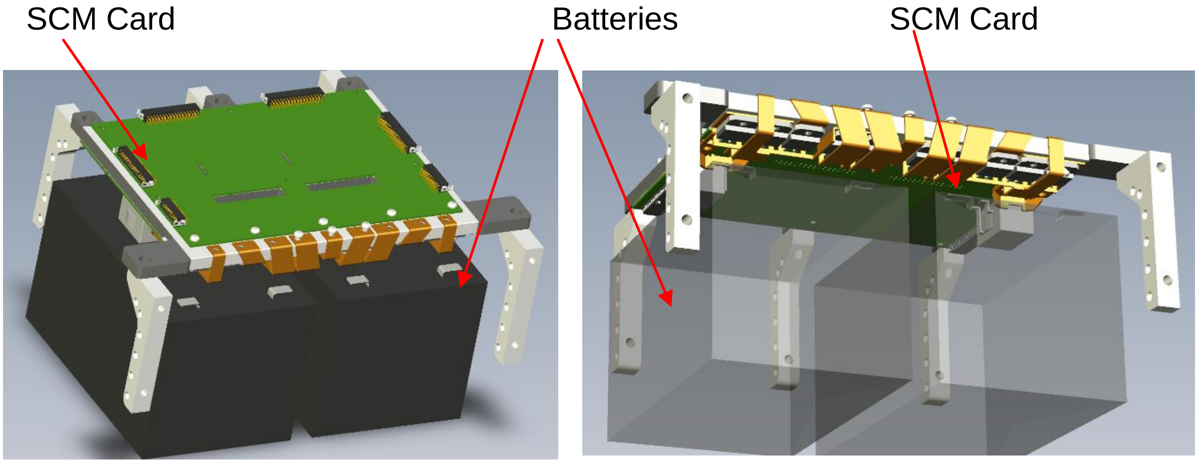


Figure 172: Drawing of the contents of the SCM including two batteries, SCM-card and PDM card.

[63] Freescale Semiconductor, "MPC5643L Microcontroller Product Brief", Doc no MPC5643LPB, 2010

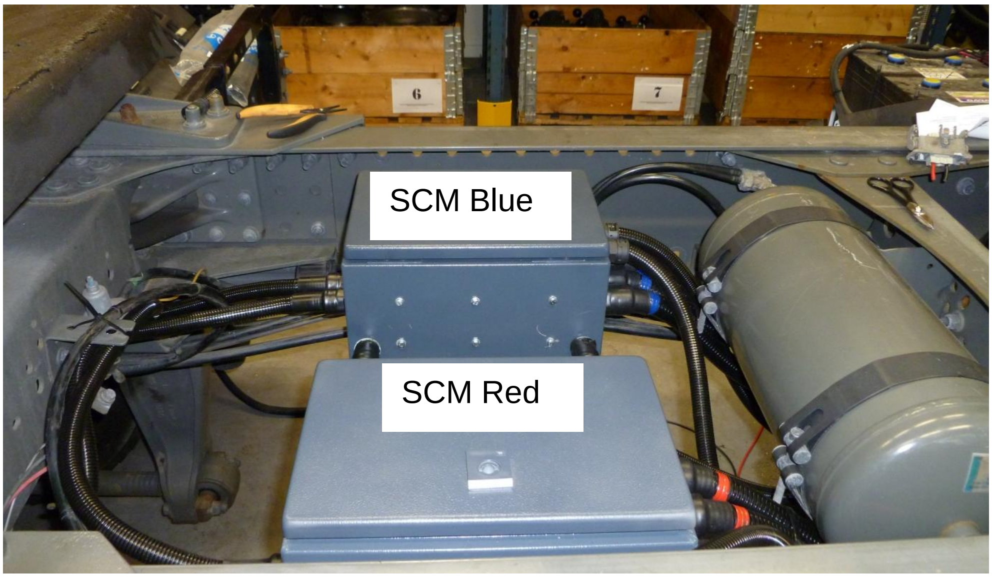


Figure 173: SCM Red and SCM Blue installed on vehicle.

System interfaces including FlexRay messages, control inputs and outputs are described in detail in HAVEit deliverable D42.1. For details on the interconnections between SCMs and XCCs, pinout etc. please be referred to D42.1 as well.

6.2.2 Signal flow and priorities

The system is redundant in the sense that it consists of two circuits (Blue and Red) including and connecting one of the brakes on the front axle, the diagonal brake on the rear axle, control units, pedal and motion sensors, and a battery. Each circuit has its own power and communication network separated from the other. The communication over the network is based on the FlexRay protocol, which is deterministic in the way that all sending signals are scheduled and no loss of data due to prioritization will occur. The signals transferred internally between the system components on the FlexRay are specified in D42.1 and in more detail in the communication matrix document[64].

The brake pedal signals and the parking brakes switches are transferred via LIN in separate private cables. LIN is a master/slave protocol[65]. The steering angle sensor transmits via private CAN, so no data loss due to priority issues will occur there either.

Additionally, to the communication networks described above, each SCM has analog connections to the brakes of the other circuit. These connections make it possible for one side to read one of the clamp force sensors and also to control the electro magnet for the park brake spring on the transverse side. This is to ensure some diagnosis and control even if one circuit looses power or completely fails. The signal flow is shown in Figure 174.

System redundancy

The SCMs consist (virtually) of a power distribution module (PDM), a sensor module (ESP) a module that controls the human-machine interface (HMI) and a XCC. The PDMs control the power and the XCCs handle the overall platform redundancy.

[64] Haldex Brake Products, “FlexRay communication matrix in the HAVEit-project”, Jacob Svendenius, 2010
[65] LIN Consortium, “LIN Specification Package”, Revision 2.0, 2003

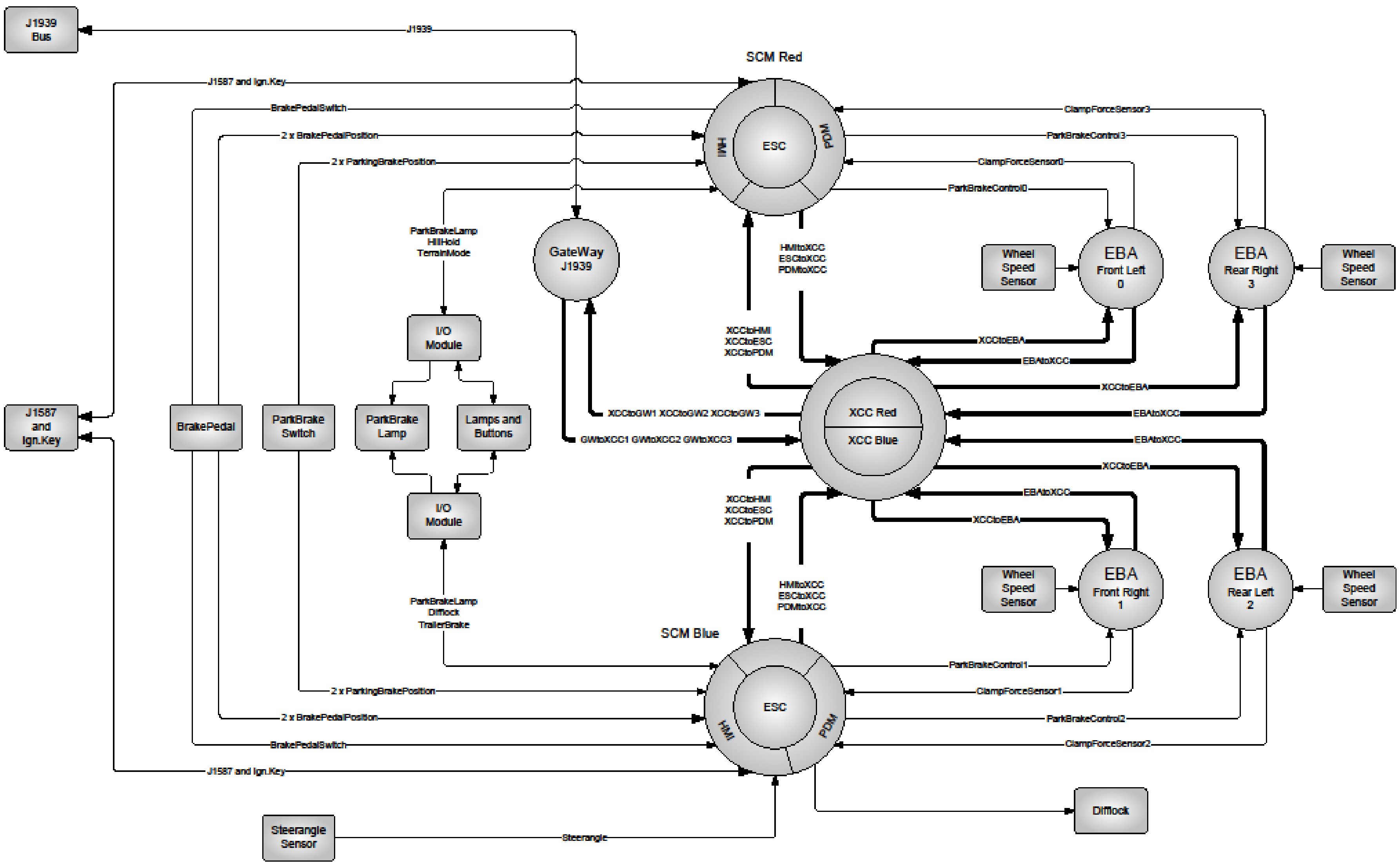


Figure 174: Schematic illustration of system signal flow.

The XCCs are connected to both the blue and red communication circuit. This means that the system application running on one XCC can send control signals to the modules and units on the both circuits. Each XCC has two lanes, which each runs the application code and transmits on to respective FlexRay net. The messages that each application transmits are received by each of the receiver's lanes. Figure 175 shows more in detail how the FlexRay frames are sent and received on the network.

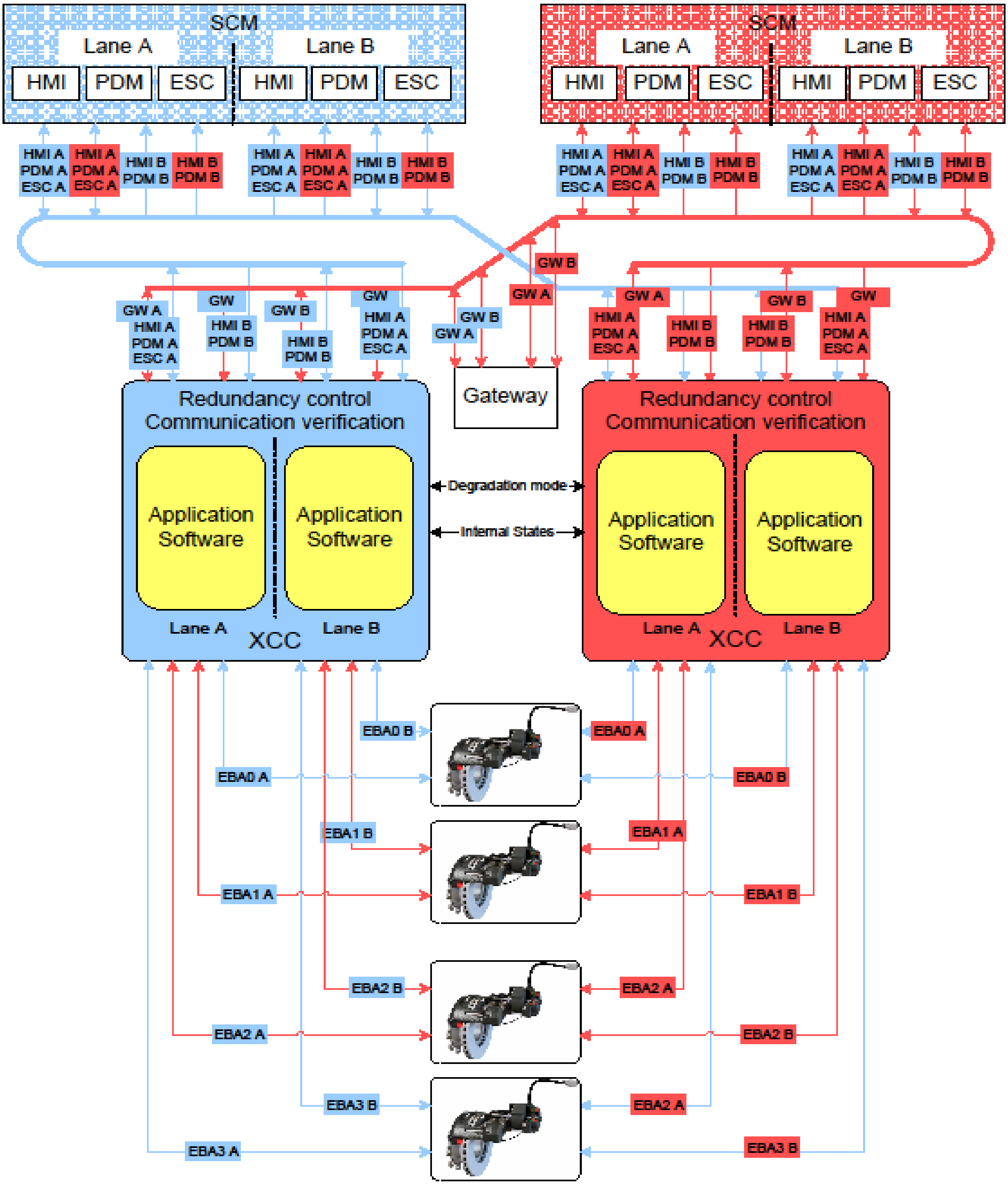


Figure 175: Schematic figure showing the flow of FlexRay frames on the communication network.

Remark that the figure virtually shows the signal flow. All red signals are sent on the same red network vice versa for the blue network. Assume that the FlexRay controller on lane B in the blue XCC sends an HMI-frame on the red network. Then the background color in the figure will be blue and the signal line color red, with the message name HMI B. This message is then received by the red SCM, both on lane A and B. The J1939 Gateway is a simplex aggregate which means that it has only one processor and it is also only connected to the red side. There is no separation in Lane A and B on the signal from the gateway. The frames in HMI are either HMI to XCC or XCC to HMI depending on direction on the arrow.

Signal redundancy and communication verification

The XCC units have two primarily tasks. One is to execute the application software which is the brake system function that translates the commands from the driver into commands to the executing units, such as the brakes and retarders taking the actual circumstances and safety issues into account. The other task is to supervise and control the communication over the two networks and provide information such that the four different executing applications can run accordingly in parallel. The XCC receives and transmits data on both the blue and red FlexRay network and on both lanes. The redundancy management takes care to determine the best available, fault free subsystem chain for execution of the system function.

The receivers must get information which of the blue and red signals to listen to. This is important since the red and blue side are asynchronous and do not exactly coincide with each other. Therefore one XCC is the master. The other, the slave, is running and prepared to take the master role after a certain time period if something is failing in the master XCC. The XCC's have internal checks on the two different lanesThe XCC adds a header in the FlexRay message frame to each aggregate that state which of the XCCs is the master. Depending on the header information from the XCCs (NA:Not Available). Each XCC tells whether it is Master (M), Master intermediate (MI) or Slave (S). If the information is ambigous the receiver must enter an idle-mode, which is a way to handle that it get no reliable inputs.

Figure 176 shows how the signal frames are received in the HMI, ESP and PDM modules in the red SCM.

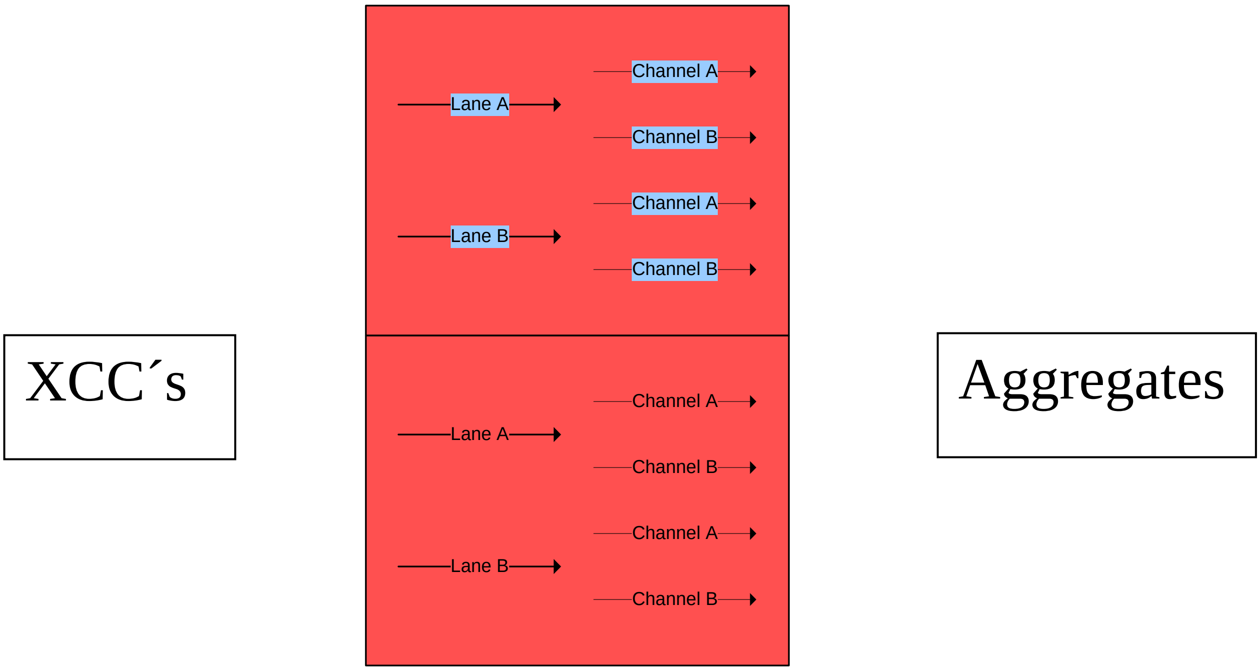


Figure 176: FlexRay signals coming into each aggregate

Four frames with the origin from the blue XCC two lanes (A, B) and two channels (A, B)) are received by the module and the same from the red side. This means that from eight incoming signals the aggregate must choose one blue side and one from red side to use. If there are no failures, the four signals from each (network) side should be equal. Between the sides the signals should be similar, but there might be small difference due to that the nets are asynchronous, without the occurrence of any failure. To assure the correctness of the signal frame that the aggregate will use a fitting pair has to be chosen out of the received frames via Channel A or B. If not the signal is marked as missing. Then the headers from each network side are compared for if they agree on who is the master. Master intermediate means that the XCC probably is not master. The header information from the both sides must be unambiguous. If it does not agree the receiving module must enter an idle mode. Table 32 shows how the module should switch modes depending on the header information from respective XCC. When the slave is in master intermediate (MI) in some cases the master can command the receiver into idle mode.

Red XCC status	Blue XCC status	Module control by
M	M	Idle
M	MI	XCC Red
M	S	XCC Red
M	NA	XCC Red
MI	M	XCC Blue
MI	MI	Idle
MI	S	XCC Red
MI	NA	XCC Red
S	M	XCC Blue
S	MI	XCC Blue
S	S	Idle
S	NA	Idle

M=Master, MI=Master Intermediate, S=Slave, Idle=passive mode

Table 32: Scheme of the mode handling in the modules (PDM, HMI, ESP, and EBA)

Also the XCCs have verification and voting rules to determine the reliability of the signals frames from the modules. From all frames entering the XCCs the application gets one signal set from each EBA (4), one from the HMI, one from the GW, and two signal sets from the PDM and ESP (there are two PDM cards and ESP sensor clusters to control and to diagnose).

Application redundancy

As noted previously, the control application software is running on four processors in parallel. This may cause problems if the input signals are not identical. The input of the two lanes of each XCC is consolidated by the redundancy management such that it is identical in both lanes of the XCC, but the inputs to respective red and blue XCC may differ. To prevent drift of internal states in the slave XCC the internal states are transmitted between the units each cycle and only the states from the master XCC will be used together with the input signals to compute the output. The XCC's have rules for determine which should be the master. Each XCC has the possibility to degrade the reliability of the application output. If some of the input signals are not reliable or available the XCC will degrade the application according to the degradation matrix in Table 33. Based on a comparison between the degradation of respective XCC, master and slave is determined unambiguously such that the best possible system path is chosen to be activated.

Deg-Level	Description	EBA	Pedal	Par-king	Switch	ESP	PDM	GW
0	No failures	at least 4 at Deg(0)	at least Deg(0)	at least Deg(0)	at least Deg(0)	at least 2 at Deg(0)	at least 2 at Deg(0)	at least Deg(0)
1	Failures that do not affect any functionality	at least 4 at Deg(0)	at least Deg(1)	at least Deg(1)	at least Deg(0)	at least 1 at Deg(0)	at least 2 at Deg(0)	at least Deg(0)
2	Non stabilizing system functions deactivated	at least 4 at Deg(0)	at least Deg(1)	at least Deg(1)	No matter	at least 1 at Deg(0)	at least 2 at Deg(0)	No matter
	Stabilizing system functions deactivated	at least 4 at Deg(0)	at least Deg(1)	at least Deg(1)	No matter	No matter	at least 2 at Deg(0)	No matter
3	Brake failure, at least one working brake per side	at least one per vehicle side at Deg(0)	at least Deg(1)	at least Deg(1)	No matter	No matter	at least 1 at Deg(0) right comb	No matter
4	Brake pedal or park brake switch failure	at least one per vehicle side at Deg(0)	at least Deg(2)	at least Deg(1)	No matter	No matter	at least 1 at Deg(0)	No matter
	Brake pedal or park brake switch failure	at least one per vehicle side at Deg(0)	at least Deg(1)	at least Deg(2)	No matter	No matter	at least 1 at Deg(0)	No matter
5	Brake failure, one side without working brakes	At least one at Deg(0)	at least Deg(1)	at least Deg(1)	No matter	No matter	at least 1 at Deg(0) right comb	No matter
6 (N)	No braking possible	No matter	No matter	No matter	No matter	No matter	No matter	No matter

Table 33: Description of the application degradation in the respective XCC

Deg(0) means that the application receives a valid signal from the component. Deg(1) means that the signal is received and reliable, but not from all four sensors (brake pedal or parking brake switches). The messages from the HMI are splitted into the pedal, parking and switch in the table. The application receives messages from four EBA, one HMI (pedal, praking and switch), two ESP, two PDM and one GW.

6.2.3 Description of the system functions

Since the brake actuator has computational power of its own and the wheel speed sensor is connected directly to the brake, the wheel speed controlling functions such as traction control and slip control are performed locally in the brake.

Service brake system

The brake pedal stroke is transformed to a retardation request, which is recalculated and distributed to the brakes as brake torque requests. It is possible to graduate the braking. The system is designed to be able to give a total braking force of at least 110 kN per axle within 0,3 s, which theoretically under non faulty conditions should give brake rates above the prescription. Due to the redundancy of components and communication between the service brake control and the brake actuators a single fault on the control line will not affect the service brake performance. Further, the components of the control line have diagnostics and there are means to monitor the transmitted signals such that failures will be detected directly and indicated to the driver. The electrical brake actuators measure the applied brake force and can supervise the condition of the brake pads, such that eventual failures can be detected by the system. The power control model of the system supervises the battery conditions and will warn the driver if the function of the system is deteriorated. Even the energy source is redundant in that there are two batteries dedicated to the system and if even these both batteries are weak the power control module can take energy from vehicle battery into the brake power circuit. Auxiliary systems are switched off if the overall battery voltage is below the warning level.

Secondary brake system

The brake system contains redundant components and communication lines such that the service brake system is able to produce a total brake force above the force required for secondary performance. The secondary braking system will therefore be controlled from the brake pedal. The secondary brake system if further described in ECE-R13 section 5.1.2.2.

Parking brake system

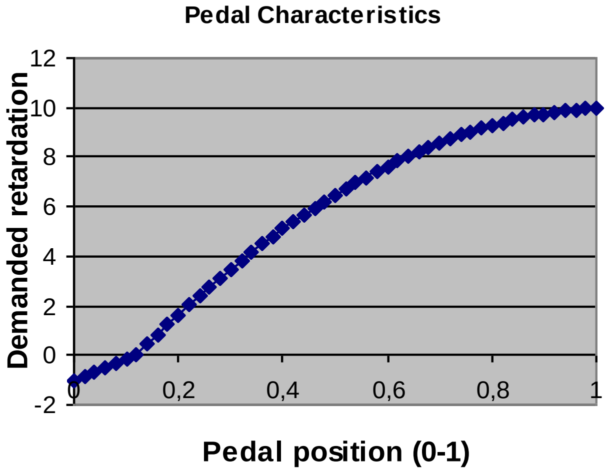
The brake actuators receive the parking brake request through the communication network from the parking brake control that is placed so it can be reached from the driver's seat. The electromagnetic clutch is disengaged, releasing the mechanically stored energy from the parking spring unit. The parking spring unit is acting with a constant torque on the motor shaft that applies the brake to a "parking brake level" to keep the vehicle at standstill by purely mechanical means. If the vehicle is not at rest when the driver commands parking brake the system will use the service brakes until standstill.

Brake assist

The function monitors the time for the driver to press the brake pedal down to a certain position. The retardation request corresponding to the pedal position will increase after this position if the depress time is fast. The function is schematically described by the retardation and pedal position relation in Figure 177. Normally the pedal characteristics follow A (A is here plotted as a straight line to simplify). When the pedal is pressed very fast, with short time between position "1" and "2", the pedal characteristics follows B.

Load apportioning

The dynamic vertical load distribution on the vehicle axles changes continuously, with the cargo position and acceleration and braking. The load apportioning function distributes the in proportion to the vertical force working on each axle. The geometrical data are assumed to be known and an estimate of the CoG-position is made from the rear axle load signal based on the air bellow pressure. Respective axle load is calculated according from a model using the CoG-position, the geometrical data of the vehicle and the acceleration measurements. An example can be seen in Figure 178. The trailer is in the example unloaded. "Mf" is torque on front axle, "Mr" on rear axle and "Mt" is the total brake torque of the trailer.



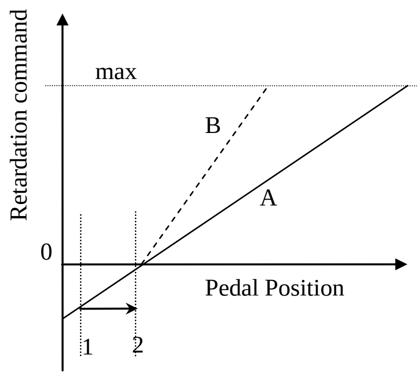


Figure 177: The left most diagram show the relation between the pedal position and retardation command. The principle function of brake assist is shown to the right.

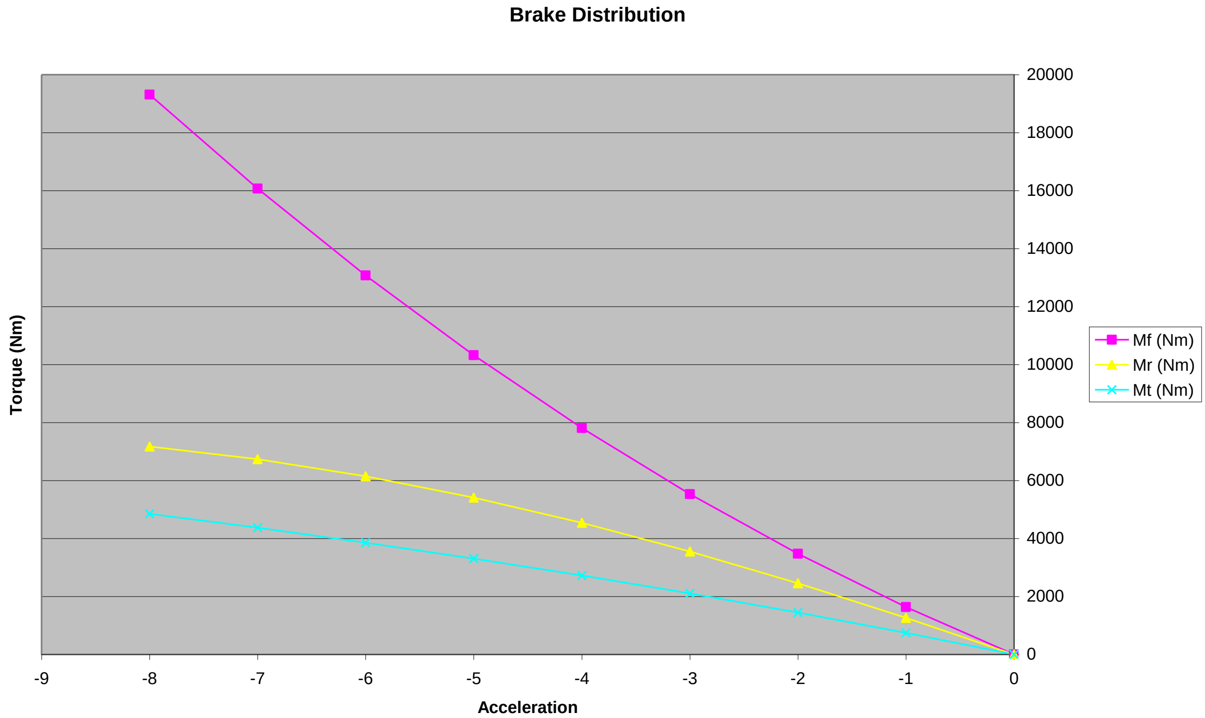


Figure 178: Brake torque on a axle in proportion to vehicle braking for a truck with trailer.

Brake blending

The main goal with a retarder system is to increase pad and disc life as well as to reduce the brake temperatures on both truck and trailer. With Brake Blending, the endurance braking systems are invoked automatically when possible and the function is enabled by putting the retarder stalk arm in “automatic” position. When the driver commands braking with automatic blending enabled, the brake force request is divided in such a way that both service braking system and endurance braking systems are applied simultaneously. In the present application there is both an engine retarder and an exhaust retarder. The endurance braking demand value is sent via the J1939 CAN bus.

The brake system is also listening to the response from the retarders on the same bus and corrects the service brake command and trailer brake command such that the demand retardation will be reached.

Slip control

The control of the tire slip is managed locally within each brake. The system provides for each wheel a reference speed (calculated from the estimated vehicle speed) to the brakes. The wheel speed sensor is connected directly to the individual brake ECU's which means that the time delay of wheel control is minimized.

Slip friction control: The system prevents sudden large brake force difference between the left and right wheels on the front axle due slip control. This could happen when the vehicle is braking on split friction surfaces. The brake application rate at the brake on the high friction surface will then be reduced and also the brake force has to be reduced in order to avoid large movements of the steering wheel. The brake force on the high friction side will be raised to its maximum, but more slow so that the driver will have time and possibility to counter steer. See Figure 179.

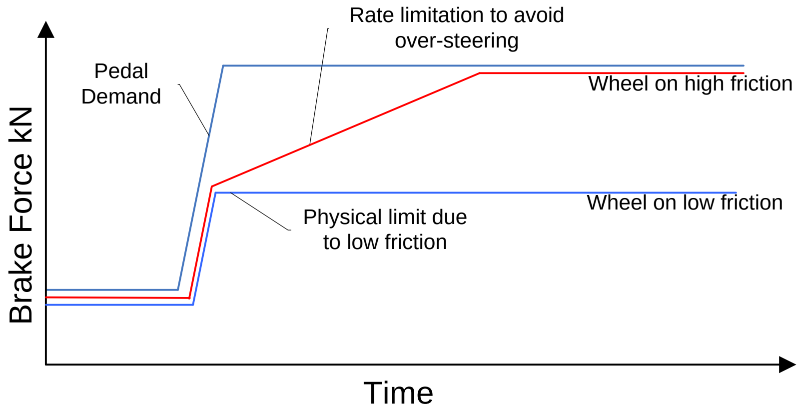


Figure 179: The behaviour of the front wheel when braking on split friction surface.

Traction control: The system prevents excessive wheel slip when the truck is accelerating. The traction control is managed by a local slip controller within the brakes together with limitations on the accelerating engine torque. Traction control performed by the brakes is an auxiliary function which is active only if the energy capacity is sufficient. The traction control is also only active below 40 km/h. There is a special "off-road" mode on the traction control function that may be commanded from the driver by a push button on the dash board. The "off-road" mode allows higher slips on the rear wheels.

Engine drag torque control

Engine braking can in slippery conditions cause high brake slip on the driven wheels. To avoid vehicle yaw instability due to this, the system will demand a torque from the engine to control the wheel slip.

Hill hold

The Hill Start Aid function helps the driver to prevent rollback when starting to drive in cases when vehicle is placed in an incline. The function will put the vehicle into a Hill Hold (HH) mode after the vehicle has come to rest using the service brake. It may also be entered upon release of the parking brake. The HH mode holds the vehicle stationary until the driver shows definite intent to set the vehicle in motion.

The function can be selected (switched on/off) by the driver. If the driver selects the function, it works as follows: when the driver has stopped the vehicle, the function commands the brakes at all wheels to keep the vehicle at stand still until any of the release conditions are met.

The default state for the hill start function is the off state. When the driver manually selects the function, it shall be actived during one “brake-stop-start”-process. Hill hold service application is limited to one minute, after which park brake will be “applied” in place of service brake. Hill hold function remains active unless the park brake is applied by the driver. During hill hold, the service brake control may be fully released by the driver.

A signal is placed on vehicle CAN bus indicating hill hold operating. The hill hold state is excited when: The service brake is again pressed beyond the HH brake force and then the brake demand should be reduced as the service brake is released. This applies even if the park brake time-out mode has been entered off if the park brake is requested or an appropriate gear is selected and the torque from the internal combustion engine (ICE) is sufficient to move the actual mass of the vehicle upwards/downwards the current incline.

Differential lock

The aim of the differential lock control is to engage the differential lock without damaging the differential to increase traction during acceleration when driving on surfaces with different friction coefficients on the left and right side of the vehicle.

The driver commands engagement of the differential lock by pushing a button on the dash board. A lamp on the button will start blinking until the lock is engaged. Subsequently, the light becomes permanent.

When the drive axles shall be locked by the differential lock, the difference in wheel speed between the driving wheels must not differ more than 2 km/h (this is a parameter that shall be set) for hub reduction and 7.5 km/h (this is a parameter that shall be set) for single reduction prior to the moment when the differential lock becomes engaged.

Active synchronisation of the wheels shall only be performed 4 seconds (parameter that shall be possible to change) and if the wheels are not synchronised by then they have to be synchronised by themselves before the differential lock is engaged. The synchronisation shall be done in cooperation with the traction control function by extra limiting of the engine torque if necessary. The traction control engine control algorithm shall be used for this purpose with a request from the differential lock synchronisation function. Observe that the differential lock synchronisation shall not be performed when the traction control function is in off road mode.

6.2.4 System validation by winter tests in North Sweden

Brake performance test performed during winter tests shows a better control of the wheel speed deceleration when comparing the current pneumatic system (EBS) and the new Brake-by-Wire system using Electro Mechanical Brakes (EMB), see Figure 180. Due to the optimized slip control a shorter braking distance is achieved.

As part of the validation tests the HAVEit Brake-by-Wire truck (see Figure 164) was brought to Haldex test facility in Arjeplog of northern Sweden. Primarily the evaluation of initial ESC was performed (see measurement results in Figure 180).

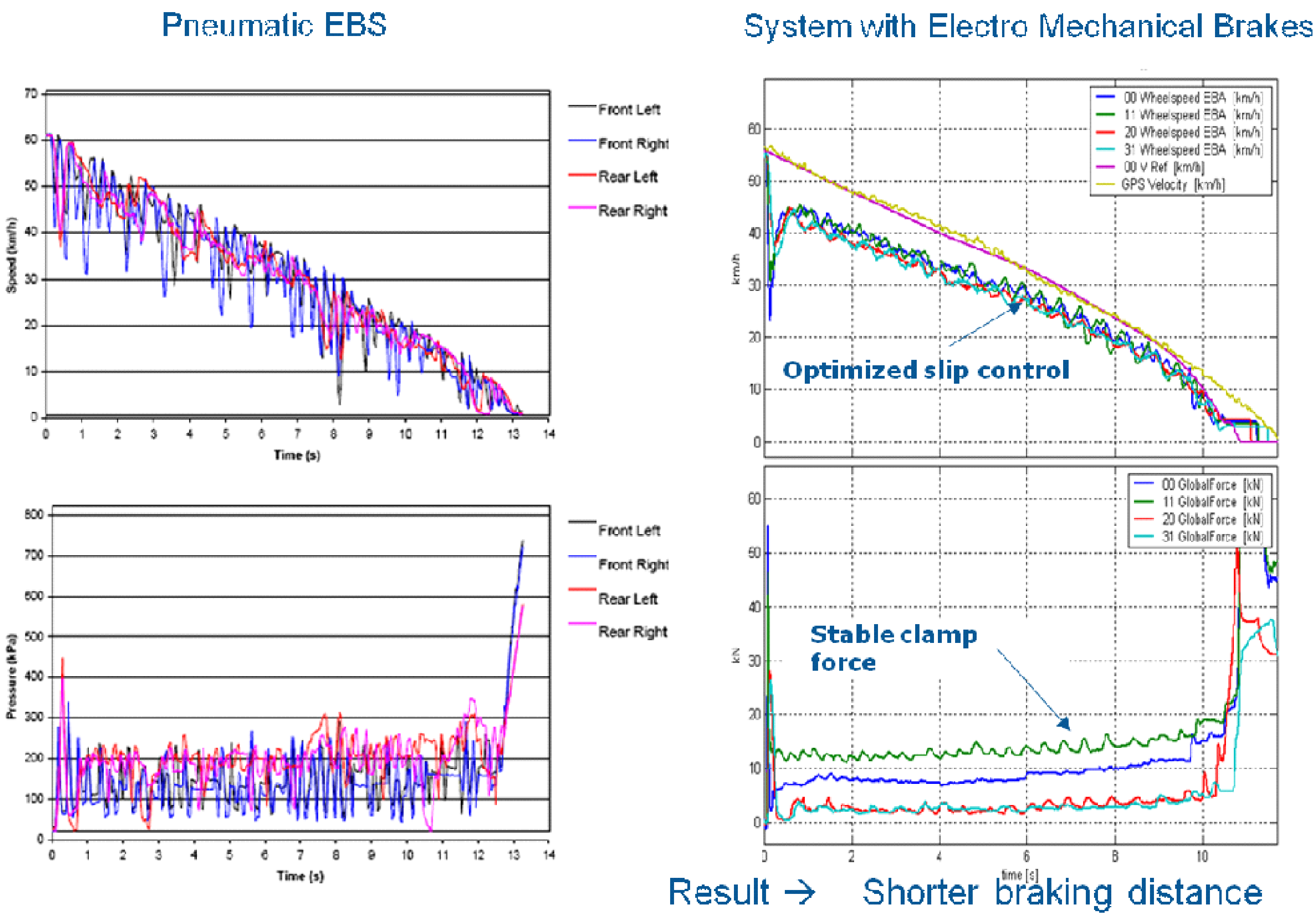


Figure 180: Electro Mechanical Brakes

6.2.5 Pre-homologation

One of the objectives set up at the beginning of the project was to perform a pre-homologation of a Brake-by-Wire system for a truck. Earlier attempts of developing Brake-by-Wire on heavy vehicles (see for example the SPARC project, a STREP project within FP6 with contract No. 507859) have mainly been focusing on brake performance improvement. Within HAVEit focus has been on system safety when using a Brake-by-Wire system. To verify that the system safety aspects were met and also to move the position forward towards a system prepared for serial development, a close co-operation have been done during the project with TÜV Nord in Germany.

This result will serve as a good base for further development in this area and as mentioned in the executive summary updates are suggested in certain areas of ECE-R13 in order to fully benefit from the system architecture and its performance. Based on internal version of an EMB product specification, further in the document referred to ID_EMB, the whole structure of ECE-R13 has been reviewed in the light of the Brake-by-Wire system shown and described in D42.1.

Working out the documentation to cover ECE-R13 in this pre-homologation process has been very valuable to understand where the existing regulation needs to be updated. However in most cases the EMB system proved to be able to cope with the existing version of the ECE-R13as the intension of the regulation is not to prescribe a specific technology to be used but the performance and safety level of the system.

The work being done will serve as a good platform for future developments in this area. In addition, below the summary statement from the TÜV Report[66].

[66] Technical Report, Approval Report No: EB177.0E according to ECE - Regulation No. 13/11, TÜV NORD Mobilität GmbH & Co. KG, IFM – Institute for Vehicle Technology and Mobility, Adlerstraße 7, 45307 Essen, Germany

19 Summary

Based on the submitted documentation presented by the manufacturer **Haldex Brake Products AB**, the **Haldex EMB**+ system as described in ID_EMB has been technically assessed according to the "Table "Scope of assessment" of paragraph 2.2 taking also into account of its footnotes.

As described in detail in paragraph 2.4 the new EMB technology is not been fully covered by the technical requirements of ECE-R13. This is due to the fact that the EMB produces actuation forces using electrical energy instead of hydraulic or pneumatic energy in the case of conventional braking systems. Therefore, due to the nature of the electro-mechanical braking system the Haldex EMB+ system cannot literally meet all of the technical requirements of ECE-R13.

For the requirements which cannot be met proposals are made (see paragraph 2.4) for updating ECE-R13.

This report assesses the **Haldex EMB**+ system for vehicles of categories N_3 and M_2 /M_3 according to the requirements of ECE-Regulation No. 13 including Draft Supplement 7 to the 11 series of amendments with regard to the items/functions/requirements as specified in the Table "Scope of assessment" of paragraph 2.2 with its numerous footnotes which indicate, e.g.

- deviations from ECE-R13 or
- tests which are not yet carried out but prescribed by ECE-R13 (in particular see footnote 2).

Deviations from ECE-R13 in general are specified in this report in particular in paragraphs 2.2, 2.4, 4.1.1 (Special additional requirements for service braking systems with electric control transmission) 4.1.3 and 6).

Essen, 27th May 2011
TDB/Gaupp
Order-No.: 8107636263

TÜV NORD Mobilität GmbH & Co. KG
Institute for Vehicle Technology and Mobility (IFM)

Technical Service for Braking Systems

Dipl.-Ing. Winfried Gaupp

Accredited according to DIN EN ISO/IEC 17025:
D-PL-11109-01-00 / Designated as Technical Service by Kraftfahrt-Bundesamt: KBA-P 00004-96

Figure 181: Summary statement of the TUEV report on pre-homologation of the HAVEit Electro Mechanical Brake System

6.2.6 Conclusions

During the implementation process all system components have been verified individually before bringing the system together. As some deviations in both the SCM ECU and the PDM ECU were found during this verification process a second ECU generation had to be ordered.

A number of integration workshops have taken place, primarily between Haldex and University of Stuttgart, ILS, to perform the implementation between XCCs and SCMs.

In order to create a base for future work many signals were implemented which increased the demand for processor power in the system. Therefore, the sampling times between the SCMs and the XCCs were increased from 10ms to 15ms.

Beneficial during the implementation was the use of the USTUTT software development processes and the opportunity to perform a direct download procedure onto the XCCs.

System functionalities have been implemented and successfully tested. Key functionalities, e.g. shortening of the stopping distance will be demonstrated during the HAVEit final event in June 2011.

In parallel with the system implementation preparations for the homologation together with TÜV Nord were done. The results hereof are reported in detail in D42.2. The summary of this report is presented in the previous subsection, more details are available in D42.2.

Summing up, this means that all objectives for HAVEit challenge 4.2 have been reached.

6.3 Challenge 4.3: Architecture Migration Demonstrator

The Architecture Migration Demonstrator was intended to show a migration path of HAVEit technology towards mass production. Instead of realizing the HAVEit architecture using proto-typing equipment, the goal of this demonstrator was to realise the Joint System with automotive standard electronics, namely the Chassis and Safety Controller (CSC) from Continental.

The base vehicle, a Volkswagen Passat, was equipped with actuators controlled through CAN bus interfaces. Longitudinal and lateral controllers were developed to accept the HAVEit Motion Control Vector to actually drive the car. Two Continental ADAS sensors - a radar sensors and a mono camera - allow simple lateral and longitudinal automation within one lane.

The HAVEit Joint System has been deployed on six CSCs connected by four CAN buses, according to a slightly adapted HAVEit common architecture, which has been set up using the AUTOSAR methodology. The team developed a special software tool to save part of the manual work of the AUTOSAR configuration process.

The HAVEit applications (or functions) developed within the sub-project SP3000 (Joint System development) such as the co-pilot and MSU were adapted where transferred to the embedded controller of the CSC. All of them are running successfully, showing that the Joint System can operate on an electronics platform compliant with automotive standards.

6.3.1 Demonstrator configuration

The demonstrator platform is a VW PASSAT Variant Highline, TDI 3C, 2.0l, 125KW, equipped with a double clutch automatic transmission. The vehicle is based on the Volkswagen PQ46 platform (see Figure 182). In this vehicle several electronic control units are originally supplied by Continental, e.g. the transmission ECU and engine ECU, which made it easier to add electronic interfaces. These CAN based interfaces are used to control the vehicle's actuators (steering, engine, transmision, brakes) and to realize its motion in case of using an automated HAVEit driving mode.

Figure 182: Architecture migration demonstrator

Figure 183: Reworked trunk layout for presentation

Various components of the original VW Passat had to be modified or replaced in order to realize the HAVEit functions. Those modifications were described in detail already in D43.1. In the following, just a brief summary is given. The figure below shows the installation of the HAVEit signal processing units in the trunk of the demonstrator.

The components in the trunk, mainly forming the execution layer, are covered by a wooden lid (here shown opened). Underneath, there are only the batteries and some fuses. The HAVEit Joint System / command layer is built of the six CSCs. It is entirely placed inside the 19'' rack housing. The housing is mounted vertically in a hole of the trunk, so it can be easily removed, e.g. to work in the laboratory.

Vehicle control function (execution layer)

The Vehicle Control Model is developed in Matlab/Simulink and comprises of the following features:

- The *real time interface blocks* (*DSpace RTI CAN block sets*) which provide the controller model with all the necessary *car sensor* data.
- The *Motion Vector* and *Joystick inputs (for testing purpose)*.
- The *Longitudinal Control Module*
- *The Lateral Control Module.*
- *RTI CAN block sets* to relay the values of *engine torque, steering actuator torque, brake torque* etc. which are computed by the controllers to the respective *actuators* in the car.

Longitudinal control module

The controller makes use of the basic longitudinal force equations of vehicle dynamics to arrive at a plausible relation between the acceleration requested by the command generation and the Torque required at the wheels to realize that acceleration. Once the relationships have been established, the same are modeled in Simulink and calibrated for the test vehicle by intensive on-the-track testing.

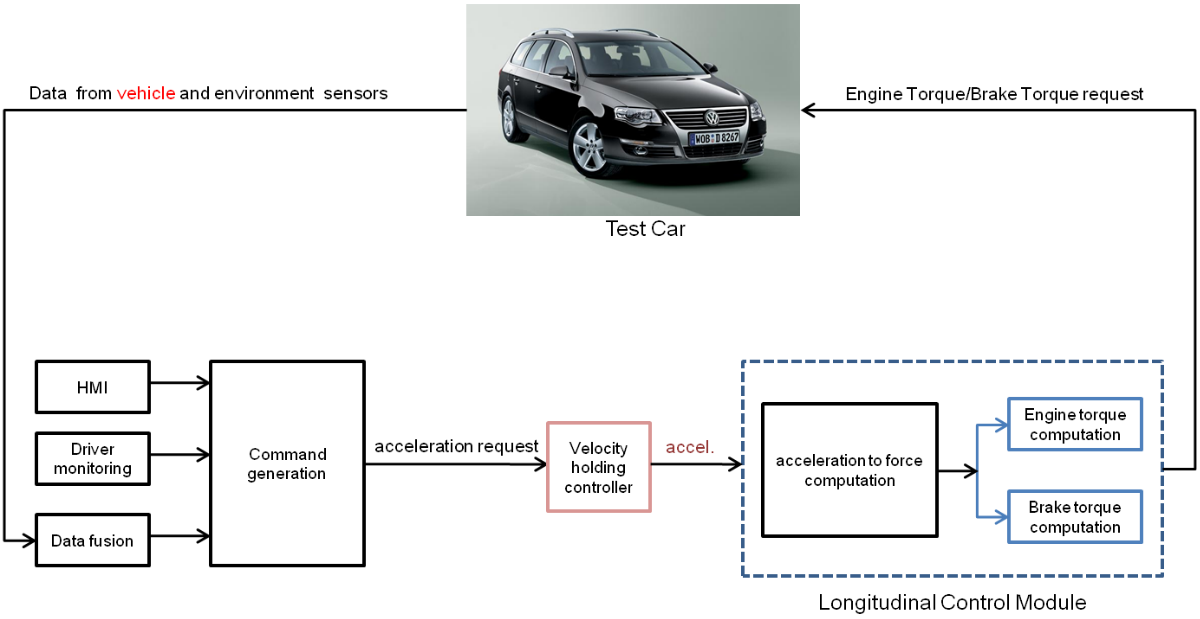


Figure 184: AMD implementation of longitudinal control

The Longitudinal Controller module comprises the following:

- An engine model which outputs a net force value 'F' based on the acceleration requested by the command generation.
- A look-up table which, depending on the engine rpm and the minimum rate of fuel injection, produces a comparative force value based on which the controller either activates the brake system for decelerating, or sends a torque request to the engine ECU for accelerating the ego vehicle.

- An engine torque computation module which converts the force ‘F’ (in case a positive accel. is demanded) required into a corresponding torque value to be supplied by the Engine.
- A brake torque computation module which converts the force ‘F’ (in case a negative acceleration is demanded) required into a corresponding brake torque value which is sent to the recuperative brake system control unit.
- A PI controller positioned before all the above mentioned modules for velocity holding when the acceleration request is zero.

Lateral control module

The Lateral Control module is responsible for making the car hold the designated ‘curvature’ on the road. The curvature demand is sent by the command layer after having processed all the data about the immediate stretch of road in front of the car with the help of the camera and radar.

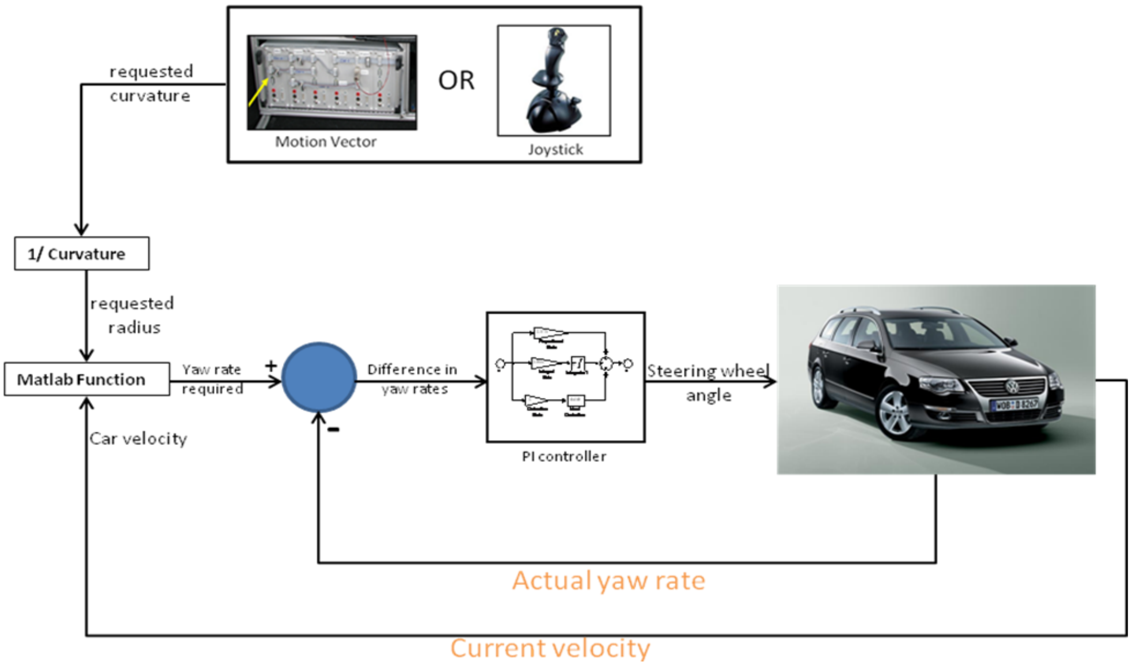


Figure 185: Overview of the lateral control logic with PI control

Once the curvature is processed by the command layer it is sent to the lateral control module. There, the requested curvature is transformed into the needed yaw velocity using the equations of the single track bicycle model.

The yaw velocity is then compared with the ‘measured or actual’ yaw velocity of the car which is provided by the Vehicle Observer sensor cluster. The difference of the required and actual yaw velocities is fed as input to a parameterized PI controller which then sends a control input in the form of a ‘steering angle’ to the EPAS motor attached in the steering column.

Both longitudinal and lateral vehicle controllers were tested extensively at the test track of Continental in Regensburg in several scenarios. Having successfully completed the execution layer developments, the controllers were integrated with the command layer.

Command layer

The main functional blocks of the Command Layer for the Architecture Migration Demonstrator were derived from the generic ones developed in the sub-project 3000 of the HAVEit project. To make the applications run on the embedded controller platform of the Chassis and Safety Controller (CSC), some adaptations were necessary.

Co-pilot

The co-pilot calculates optimal subject vehicle trajectories and speed profiles with respect to the environment detected by the perception layer, Figure 186 and Figure 187. The Architecture Migration Demonstrator uses the legal safety-based trajectory planner. It is an all-in-one co-pilot algorithm that takes into account the different aspects of driving, e.g. keeping distance to objects, avoiding right overtaking, adapting road conditions, respecting speed limits and adapting speed to curves (see section 5.1 above for details). The character of the co-pilot (e.g. comfortable, green, and sportive) is matched to that of the vehicle and driver. One trajectory per lane is calculated to be used during normal system operation (0A, 0B, 0C) and one trajectory per lane is determined to be used in case of a system failure (FA, FB, FC). In the demonstrator, the trajectory in the lane of the subject vehicle for normal system operation (0B) is used for controlling the vehicle.

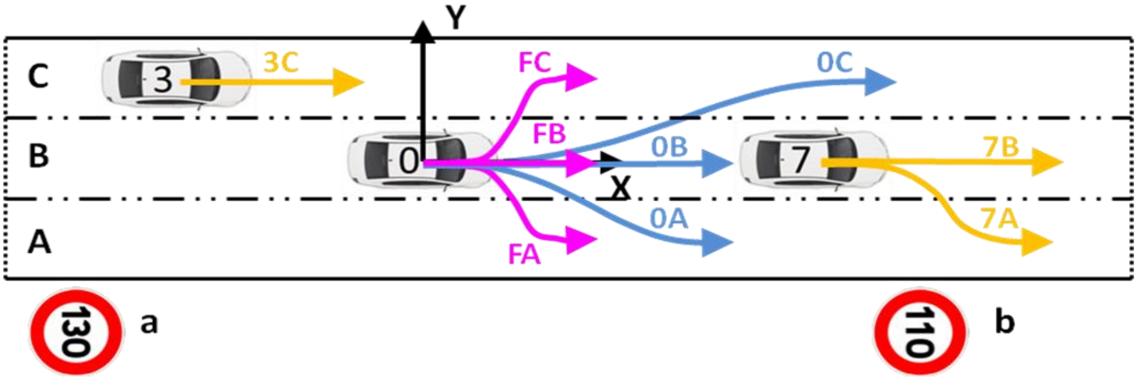


Figure 186: Trajectories of subject vehicle (0) during normal system mode (0A, 0B, 0C) and system failure mode (FA, FB, FC). Trajectories of object vehicles (3,7).

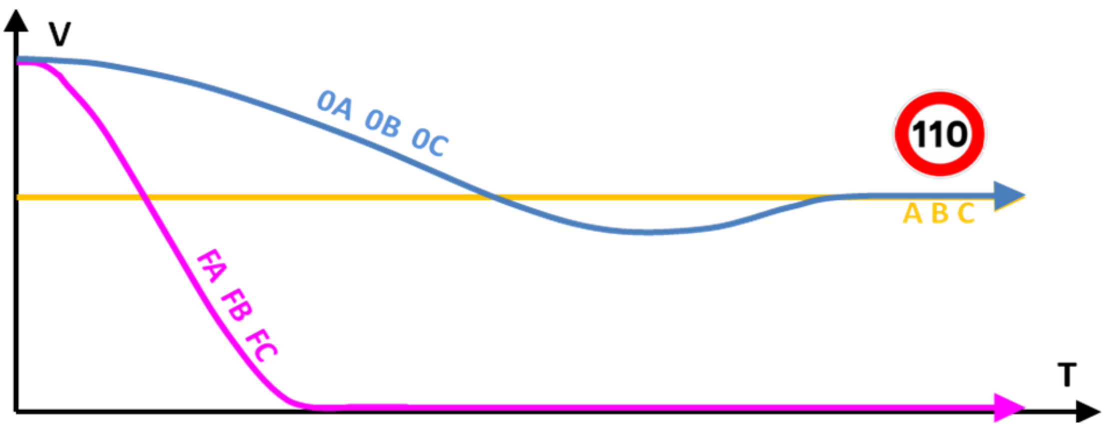


Figure 187: Speed profiles of subject vehicle during normal system mode (0A, 0B, 0C) and system failure mode (FA, FB, FC).

The co-pilot takes the traffic rules of the United Nations Vienna Convention on Traffic Safety (1968) as a basis to the predict object trajectories, as indicated in Figure 188 and to predict the presence of possible *phantom* objects outside of the perception horizon, Figure 189. The algorithm ensures safe driving (assuming traffic rules are respected also by all other traffic participants). In case of potential hazards, collision avoidance/mitigation functions are included.

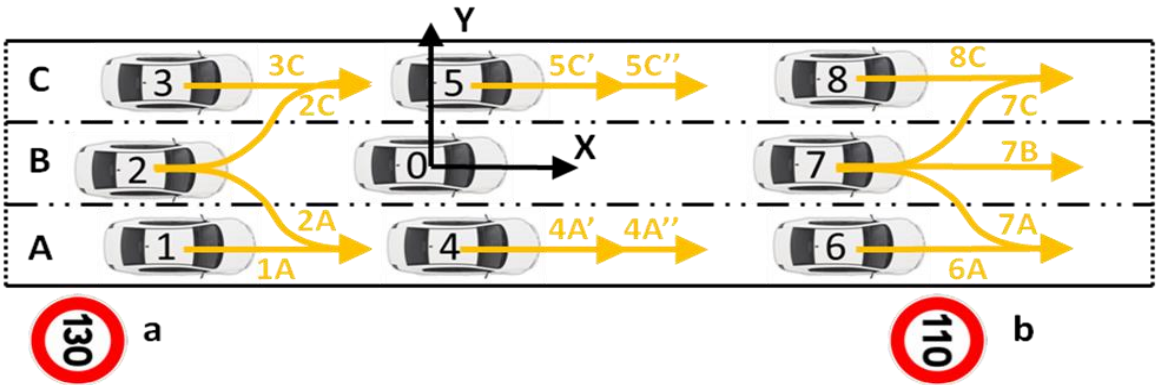


Figure 188: The prediction of object trajectories according to legal safety

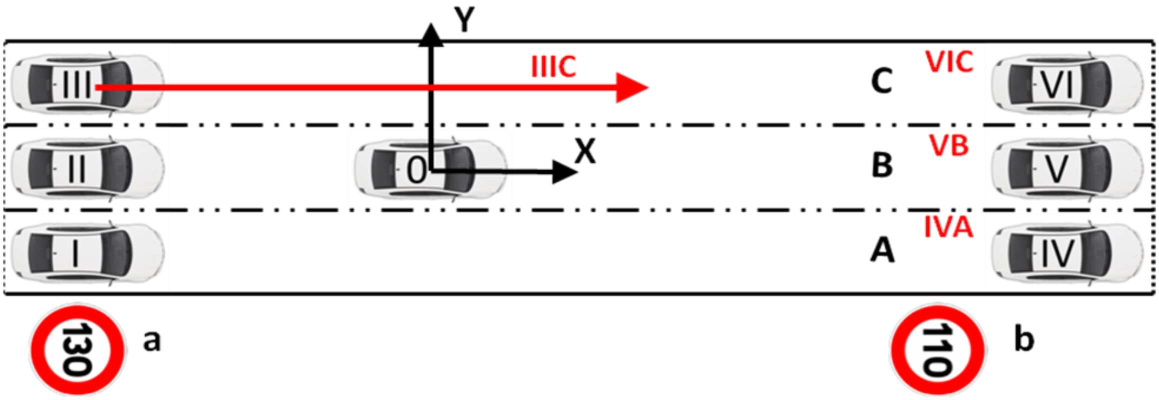


Figure 189: The prediction of phantom trajectories according to legal safety

The complete co-pilot algorithm functionality is implemented on the Architecture Migration Demonstration vehicle and its performance was optimised till project end. The cycle time of the algorithm on CSC is 20ms, including reading and writing of CAN data, and including the calculation of trajectories, other than 0B, for extending system functionality; i.e. trajectories to right and left lanes 0A and 0C and failure safety trajectories FA, FB, FC. The short calculation time ensures the stability of the perception, co-pilot and control automation loop and the possibility to migrate the HAVEit co-pilot towards automotive embedded platforms.

Command generation

The command generation and validation module controls the vehicle along the trajectory and determines the speed profile proposed by the co-pilot. It produces acceleration demands for the low level longitudinal control and a desired curvature value for the low level lateral control components. Different command generation algorithms were developed and tested by the Joint System development team. The algorithm implemented in the longitudinal command generation of the Architecture Migration Demonstrator, is a PID controller with noise filtering on the acceleration proposed by the speed profile. For the lateral command of the vehicle, a PID controller with adaptive parameters (adaptive in terms of vehicle speed and trajectory curvature) was realised.

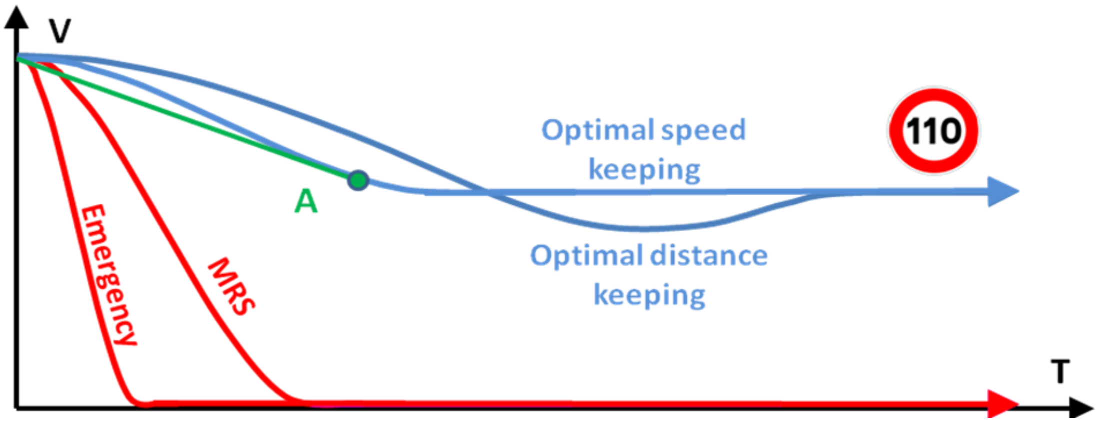


Figure 190: Longitudinal command of the vehicle on a speed profile

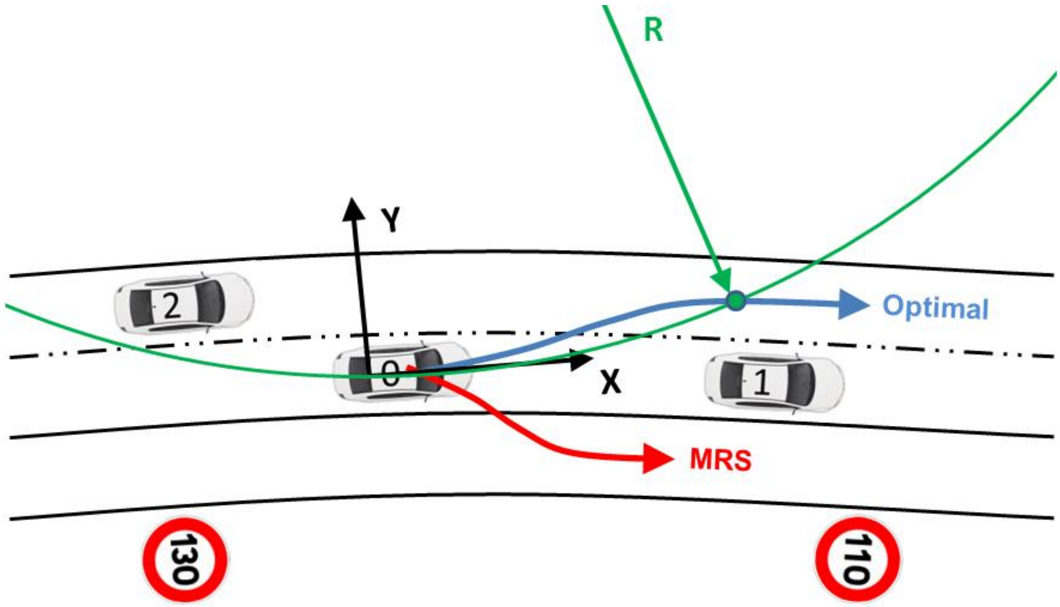


Figure 191: Lateral command of the vehicle on a trajectory

The command generation algorithm performs fast calculation times of 10 ms on the CSC, including CAN reading and writing. Thus, it demonstrates and validates the possibility of migrating the HAVEit command generation and validation towards automotive compatible embedded systems.

Mode selection and arbitration unit

The optimization work regarding the Mode Selection and Arbitration Unit (MSU) mainly consisted of corrective modifications (e.g. in vehicle start up initialization) and parameter tuning (e.g. in setting the speed limits for the automation modes Highly-automated and Semi-automated to 80 km/h and 120 km/h respectively). During the optimization process several updates of the MSU software were provided by partner DLR and installed on the MSU CSC in the demonstrator car.

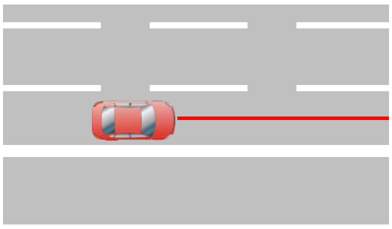
6.3.2 AMD system validation

As mentioned, the functional content of the Architecture Migration Demonstrator (AMD) focuses on vehicle automation in one lane. Nevertheless, the grade of functional performance proofs the feasability of the chosen concept for system realization with embedded electronics and standard CAN bus communication. The following section gives an overview about the use cases of the AMD as presented in D43.1 and adds the results of the performance measurements that were carried out during the optimization phase of the HAVEit project. Table 34 summarizes the common parameter description for the understanding of the data logs to follow.

Parameter symbol (unique identifier)	Parameter meaning
VEH_STATE_VS:	ego vehicle speed
AUT_LEVEL_ACT:	current requested automation level
DIST_OBJ_AHD_MID:	distance to host vehicle on the same lane
DIST_OBJ_AHD_LE:	distance to host vehicle on the left lane
VEH_STATE_AC_LGT:	ego longitudinal acceleration
VHM_CRVT_REQ:	motion vector requested curvature value
VHM_AC_REQ:	motion vector requested acceleration value

Table 34: Command layer parameter meaning

Use case "Normal driving in a lane without obstacles and activation of different automation levels possible"



This use case shows the driver initiated activation and de-activation of different automation levels while staying in a lane without obstacles. All possible transitions between the automation levels are covered.

In this use case Highly Automated driving corresponds with a Cruise Control (CC) as a longitudinal action, combined with a Lane Keeping System (LKS) as a lateral action.

Switching between different automation levels is illustrated in the performance log plot shown in Figure 192:

- From DA to SA (starting at timestamp 45): The ego vehicle stays with constant speed (until timestamp 48), cruise control is successfully invoked and active.
- From SA to HA (starting at timestamp 50): Longitudinal and lateral control are active, the ego vehicle stays at its speed, the command generation output (green curve) starts with a high value to reach the higher set speed and is slightly decreasing after set speed has been reached.
- From HA to SA (starting at timestamp 56): Cruise control again is still active.
- From SA to HA (starting at timestamp 58): Longitudinal and lateral control are again active, the ego vehicle stays constantly at the set speed.
- Deactivation of HA and HAVEit at all (timestamp 66) by changing into Driver Only mode.

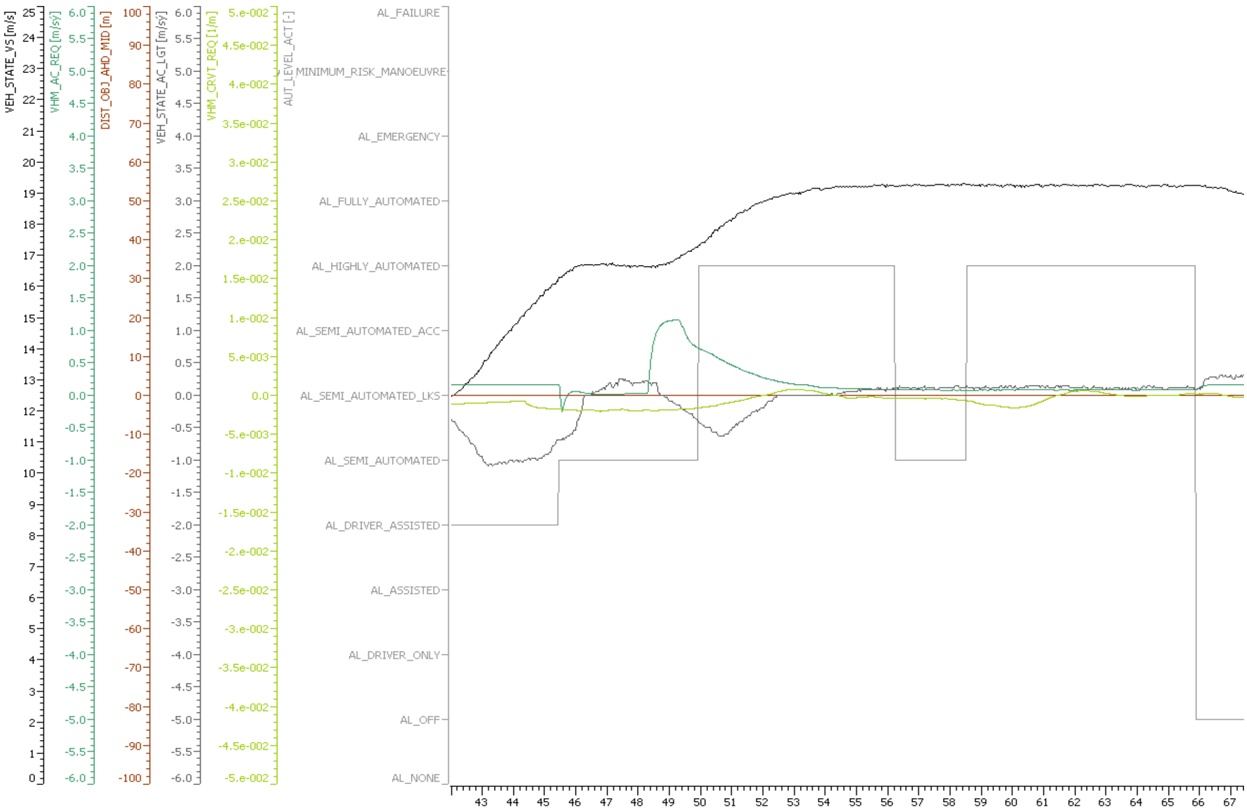
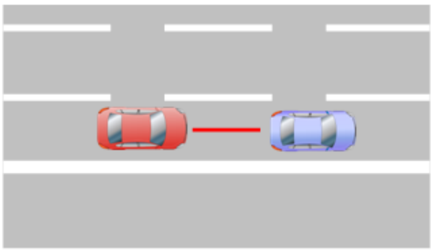


Figure 192: Command layer performance in use case "Normal driving in a lane without obstacles and activation of different automation levels possible"

<u>Use case "Normal driving in a lane with obstacles and activation of different automation levels possible"</u>



In the presence of an obstacle in the lane of the ego vehicle the same transitions of the automation mode as in the previous use case occur. The system can then act as an Adaptive Cruise Control (ACC) with additional stop-and-go functionality instead of a CC in the longitudinal direction. In the lateral direction, the system corresponds to an LKS, as in the previous use case.

The performance plot for this use case is shown in Figure 193. A leading vehicle (object) is present in own lane:

- SA (Semi-Automated) is active; the leading vehicle is about 50m in front of the ego vehicle.
- The distance between the ego and the leading vehicle is kept constantly at about 50m, independent from the activated automation level (SA or HA).
- A transition from HA (Highly Automated) to SA is starting at timestamp 56: The set speed not changed by the driver, thus cruise control remains active.
- HA is active between timestamps 86 and 90: Both longitudinal and lateral controls are active. The command generation produces the requested curvature value to hold the ego vehicle between the left and the right lane mark.

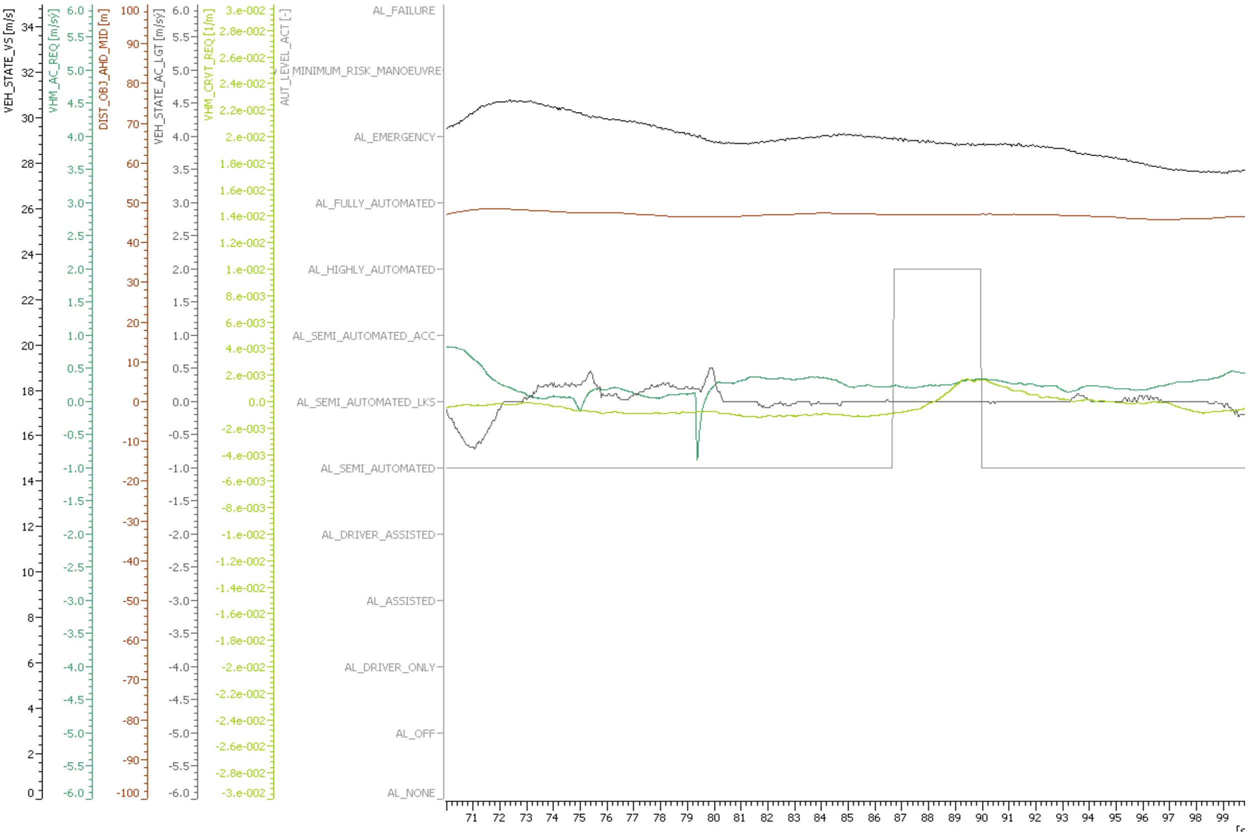
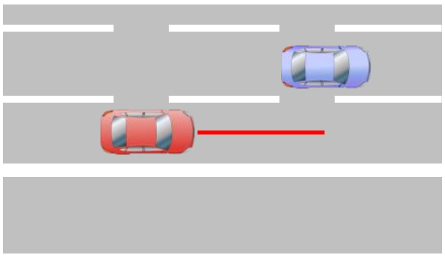


Figure 193: Command layer performance in use case "Normal driving in a lane with obstacles and activation of different automation levels possible"

Use case "Driving and avoidance of right overtaking"



This use case covers a specific driving situation when a slower vehicle is in a lane left of the ego vehicle. Depending on the automation mode, the system helps to avoid right overtaking by inviting the driver to slow down or by actively controlling the vehicle with an ACC functionality applied to the target vehicle on the left lane.

The command layer performance in this scenario is illustrated in Figure 194. A leading vehicle (object) in front of left lane is present:

- SA is active; the leading vehicle is present in about 32m in front on the left lane.
- Starting at timestamp 6.5 negative acceleration values are produced by the command generation to avoid right overtaking. This results in braking of the ego vehicle.
- Right overtaking is successfully avoided at about timestamp 10, the ego vehicle speed is automatically adapted to the object present in the left lane and distance to this object is kept stable.

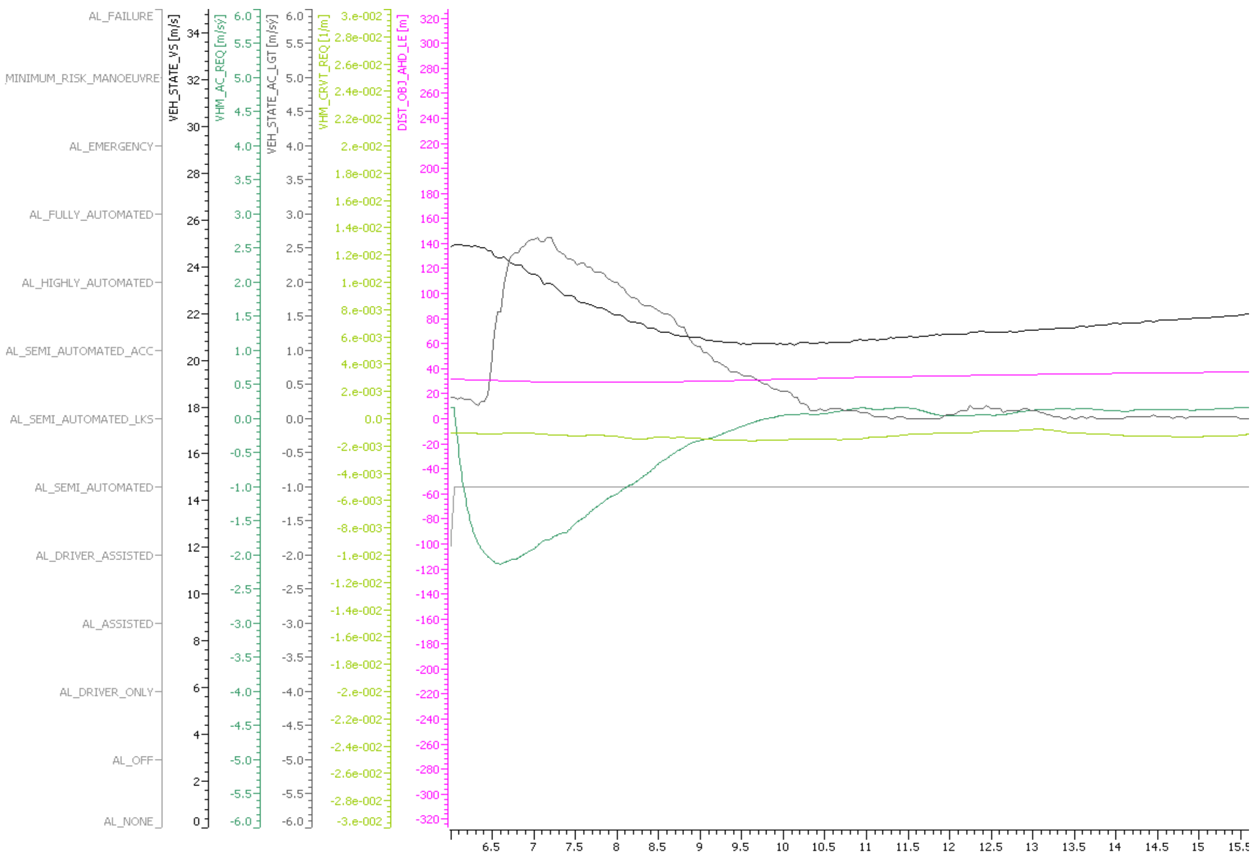


Figure 194: Command layer performance plot in the use case "Driving and avoidance of right overtaking"

<u>Use case "Driving and activation of automation mode not possible"</u>

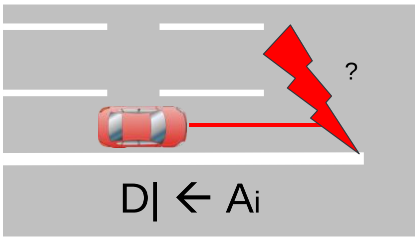
In this use case the driver wants to activate a specific automation level when the preconditions are not fulfilled, e.g. when driving too fast or in a different environment than a highway.

Examples that are covered are the refused transition from Driver Assisted to Semi Automated or to Highly Automated as well as the refused transition from Semi-Automated to Highly Automated.

A data log graph is not useful in this particular use case. However, during the test drives the desired Joint System behavior was verified:

- Whenever the HAVEit system is not capable to detect the lane, the Highly Automated mode is disabled.
- At speeds above 120km/h the highly automated mode is disabled for the Architecture Migration Demonstrator.
- At speeds above 140km/h also the semi automated mode is disabled in the AMD.

Use case "Driving, driver unresponsive and transition to Minimum Risk Manoeuvre"



In certain situations the system requests the driver to take over the control of the vehicle, e.g. if the preconditions for a specific automation level are not longer fulfilled.

When the reaction of the driver is too slow, a Minimum Risk Manoeuvre (MRM) is executed by the system in order to avoid dangerous situations. The MRM of the Architecture Migration Demonstrator is designed with a moderate deceleration till standstill in the lane of the ego vehicle.

Figure 195 presents the command layer performance plot for this use case. The driver doesn't react to a system-initiated take over request:

- First, Semi-Automated is active. As the driver doesn't react to the take over request, the automation level changes to MRM by the Mode Selection and Arbitration Unit (MSU) at timestamp 29.
- Consequently, a constant negative acceleration value is requested by the command generation and validation module.
- The negative acceleration results in braking the vehicle to a complete stop (black ramp starting at timestamp 29.5).

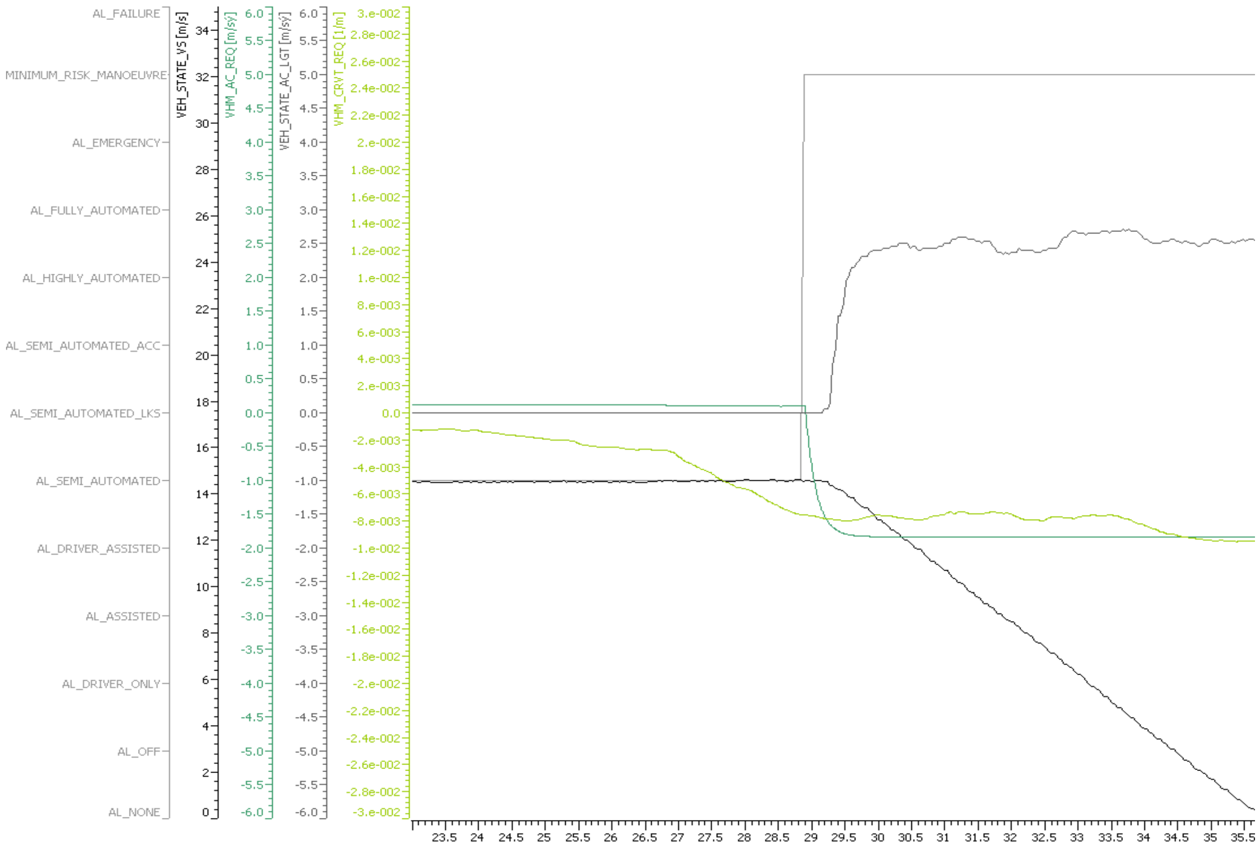


Figure 195: Command layer performance plot in the use case "Driving, driver unresponsive and transition to Minimum Risk Manoeuvre"

6.3.3 Summary and conclusion

Much work has been spent on the optimization of the functionality of the Architecture Migration Demonstrator to be able to present interesting showcases at the HAVEit Final Event.

Many test drives and several joint integration workshops with the HAVEit partners contributing to this vehicle helped to build the basis to demonstrate what is possible with state-of-the-art available automotive control technology. A full set of vehicle automation functions following the integrated HAVEit concept has been realized and allows the user to experience that these functions are not that far away from being used in normal public traffic.

This is even more valid since the smoothness of functionality was never the original goal of the work package. Its focus was to demonstrate the way of architecture development and implementation, using a modern top-down design strategy and after all putting it on automotive embedded control units rather than on PCs. With the extensive use of AUTOSAR methodology it was possible to do a quite straightforward deployment and system integration that caused significantly less errors than usual. At no time during the development phase the basic structure of the architecture was in danger of requiring a general rework.

Also the porting of the originally PC developed SW algorithms to the embedded CSC platform went surprisingly smooth, taking into account that mostly scientific partners programmed them, not being experienced in industrial SW development processes.

Thanks to the AUTOSAR methodology, most of the specific items produced during the architecture development and implementation can be reused and transferred to the serial product development departments, thus enabling them to shorten the time-to-market and decrease the development costs.

The result of this work package is a highly automated demonstration vehicle that shrinks the gap between science, advanced development and product development to a smaller dimension than ever, bringing the HAVEit philosophy of vehicle automation close to the reach of the user.

In summary it can be said that all of the work package's goals are attained to full extent. The Architecture Migration Demonstrator has successfully reached its goal: HAVEit is ready for product development.

Figure 196: Highly Automated driving with the AMD

The experience gained during the work on the Architecture Migration Demonstrator points out a number of interesting perspectives for further development.

- First, for coming architectural tasks the use of an integrated architecture development tool is not only recommended, but mandatory. The benefit will be a faster development, easier implementation of changes and even less errors.
- During the initial operation of the embedded Joint System it turned out that less than six CSCs, probably only four, would have been sufficient to implement the Joint System, thus reducing the necessary number of bus connections and/or bandwith. More powerful ECUs that are already in development could even reduce the number down to as little as two or even one, giving a good perspective on cost reduction by fewer components as well as less dvelopment time. This trend will be further supported by the use of faster bus technologies such as FlexRay or Ethernet.
- Faster bus technologies are also the key to the use of more sensors detecting a more complex environment, which gives an outlook on broadening the space of applicability of the HAVEit vehicle automation concept. Together with an enhancement by sophisticated V2X communication technologies and services the environmental detection will reach new dimensions and so will the performance of vehicle automation.

7 Vertical Challenges: Highly Automated Vehicle Applications

In this section, results achieved within the frame of HAVEit's vertical challenges in cluster 5 are described in detail. To summarize:

- Challenge 5.1: Automated Assistance in Roadworks and Congestion (ARC)
- Challenge 5.2: Automated Queue Assistance (AQuA)
- Challenge 5.3: Temporary Auto-Pilot (TAP)
- Challenge 5.4: Active Green Driving (AGD)

7.1 Challenge 5.1: Automated Assistance in Roadworks and Congestion

Introduction

The overall objective of the HAVEit project is to develop technical systems and solutions that improve automotive safety and efficiency. Continental Teves contributed to the overall objective by developing the safety and comfort focused application: Automated Assistance in Roadworks and Congestion (ARC). This highly automated vehicle application is fundamentally intended to support the driver in overload situations like driving in narrow lanes of roadwork areas with lot of vehicles driving next to the ego vehicle.

The ARC is a passenger car application which supports the driver on motorways and motorway similar roads and particularly in roadworks with different levels of automation in longitudinal and lateral control of the vehicle. In roadworks the automation will work at speeds between 0 and 80 km/h. The automation spectrum is as follows:

- Highly-Automated: automated longitudinal and lateral control
- Semi-Automated: automated longitudinal control (ACC)
- Assisted driving: assisted lateral control (Heading Control)
- Intervening safety functions: automated emergency braking

This guarantees that the driver gets the best possible support available, in particular with respect to lateral vehicle control. ARC thus will contribute to traffic safety. The most critical use case for the ARC application is driving hands-off through a road construction. This use case and all other relevant use cases were integrated and successfully tested.

The Lane Keeping system is realized by a combination of a Lane Centring Assist (see D51.3 for details), which softly (<2.5Nm) tries to hold the vehicle in the middle of the lane and a virtual Wall algorithm (see D51.3 section 1.1.2 for details). The virtual Wall algorithm is designed to push back the vehicle from lane boundaries like guard rails, by superposing a high (<4Nm) torque on the steering wheel. With this, the driver could feel some virtual Wall, on which he could even lean on.

The guardrails are detected by a low level fusion of the radar reflections and a 3D reconstruction from the mono camera picture. Vehicles driving besides the own vehicle with the same speed are detected by Short Range Radars (SRR). They are handled like a wall with the same curvature as the detected guard rails. This was identified as the most complex part to realize.

For better integration of new functions, the software structure was aligned, so there is a clear cut between perception, functions and control algorithms. Additional use cases (compared to the list defined at project start (see D11.1) were implemented:

- prevention from curve over speed

- too narrow lanes
- automated deceleration if the driver is inattentive

They could be integrated between the perception and the controllers. Therefore an arbitration unit selects the most necessary intervention from the different functions. The used development framework described in deliverables D51.1 and D51.2) was further enhanced, by integrating better measurement handling and improved failure detection.

This section presents a brief overview on the demonstrator setup and the results achieved in testing the ARC application in the relevant use cases.

7.1.1 Demonstrator configuration

Architecture

The ARC architecture follows the HAVEit architecture explained earlier in this report. Modifications to the ARC demonstrator for all 4 HAVEit layers are described in detail in HAVEit deliverables D11.1, D11.2, D12.1 and in particular in D51.1, D51.2 and D51.3. Here, just a brief summary is given.

The vehicle is equipped by Continental with a 360 degree view realized with serial production sensors (radars and camera). With these the relevant objects in the surrounding of the vehicle are detected (Figure 198). The complete environmental perception is described in detail in Chapter 2.1 of the Deliverable D51.3.

Figure 197 presents an overview of the software modules of the ARC demonstrator. Derived from the HAVEit architecture the sensors are in the upper part of the overview. The rear radar cluster is not shown to get a clearer arrangement. In fact, it would look the same as the SRR radar sensors. Also for the camera only one object detection module is shown. In fact there are three: speed limit sign detection, lane recognition and vehicle detection. The area covered by the ARC perception system is illustrated in Figure 198. For details on these sensors please see D51.3, section 2.1.

The speed limit sign detection and the lane recognition run on the camera hardware (Continental CSF200). The vehicle detection runs on a PC. Because it is one information path, it is shown as one block in the overview. Also the radar has an object detection running on its hardware. These objects are afterwards fused on object level.

Besides, the raw data from the radar and the camera are used to detect the guide walls ore beacons. Therefore optical flow is computed from the camera image. With a 3D reconstruction it is possible to generate 3D information of the surrounding.

Measurements from the 3D reconstruction with a certain height and measured radar reflections with certain energy are the fused in a stationary Grid. Afterwards the guide walls could be detected in the grid. The detected guide walls are also fused on object level for a better consistency of the data.

With all the information from the sensors it is then possible to assess the situation and to generate an intervention decision for certain functionalities. With the HAVEit project it got clear that there will be more then one function and therefore more than one software module in the vehicle at the same time in different configurations.

As mentioned there is a difference between situation assessment, decision making and control methods. The functional levels inside the HAVEit project are distinguished by their control methods. The higher the automation is, the more situations have to be assessed and the more possible interventions have to be arbitrated. Following the HAVEit structure, the selected mode from the Driver Assessment and the Mode Selection and Arbitration Unit are handled to the Command plausibilization. Depending on the selected mode, the intervention method for the control vector is switched on or off.

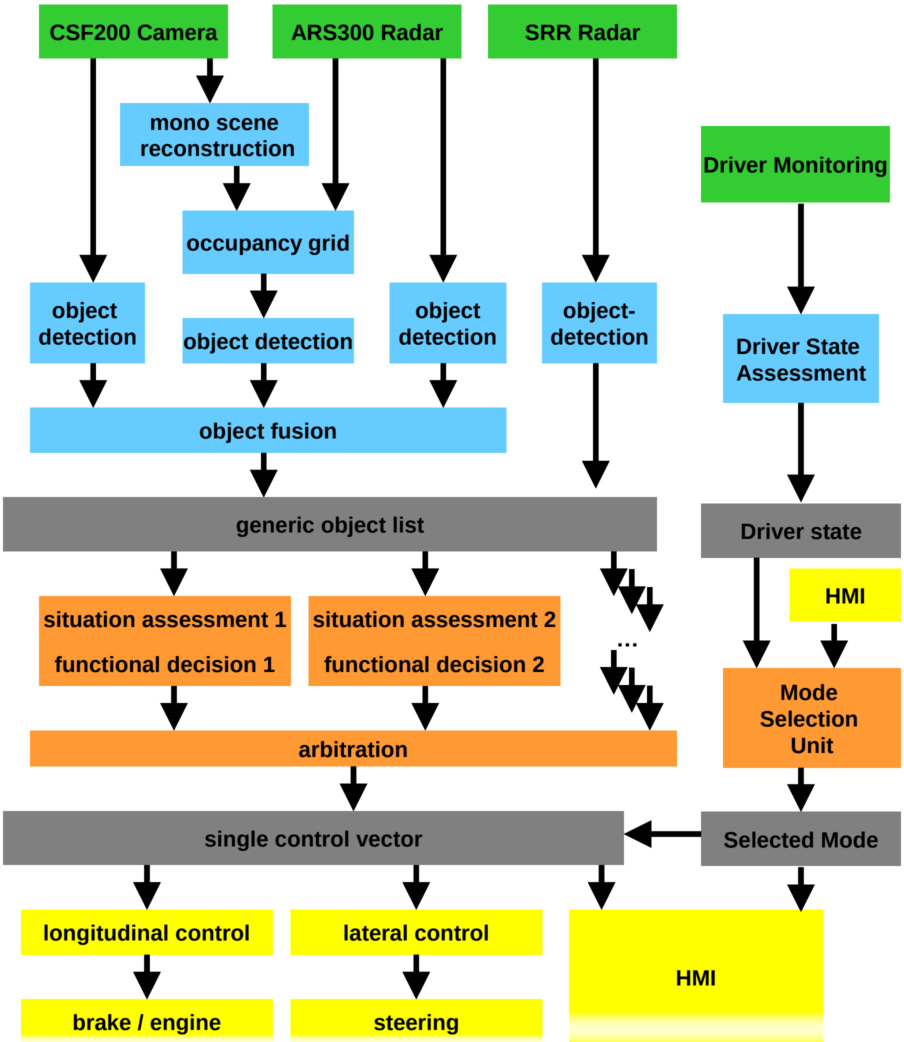


Figure 197 ARC software structure with Driver Monitoring and Mode Selection

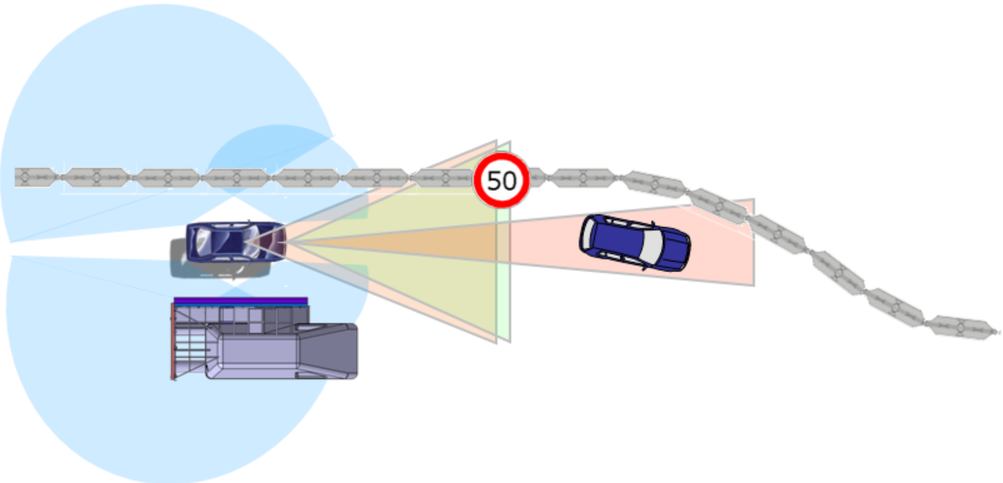


Figure 198: Sensor beams in regard to surrounding objects

Perception layer

3D reconstruction

The usual way to detect objects is to search for model based information in the picture, like described in the last chapter. This approach is not sufficient for roadwork areas, which practically cannot be fitted into such a model. In contrast, a pixel orientated approach was used. As this is specific for the ARC application, it is briefly described here.

First, optical flow is computed for the camera picture (Figure 199). With this and the accurate information of the other and the own vehicle movement, it is possible to generate a 3D picture of the vehicle surrounding beside the own driving tube. It is possible to compute the optical flow for each pixel. For the roadwork assistance, the optical flow is only calculated for characteristic sub parts of the picture to reduce computation resources for online calculation. Characteristic parts are those, were the gradient of the pixel brightness is very high.

Now the features of the first image have to be found again in the next image (feature matching). A pyramidal implementation of the Lukas Kanade Tracker[67] is used to identify those features, with the minimum displacement in the next picture. The Lukas Kanade tracker was chosen, because it showed the best ratio between performance and computation costs.

The movement of the features is caused by the movement of the own vehicle and the camera. But it could also come from moving obstacles. Optical flow from moving obstacles will lead to misinterpretations. This is why moving obstacles detected with the radar are excluded from the following steps.

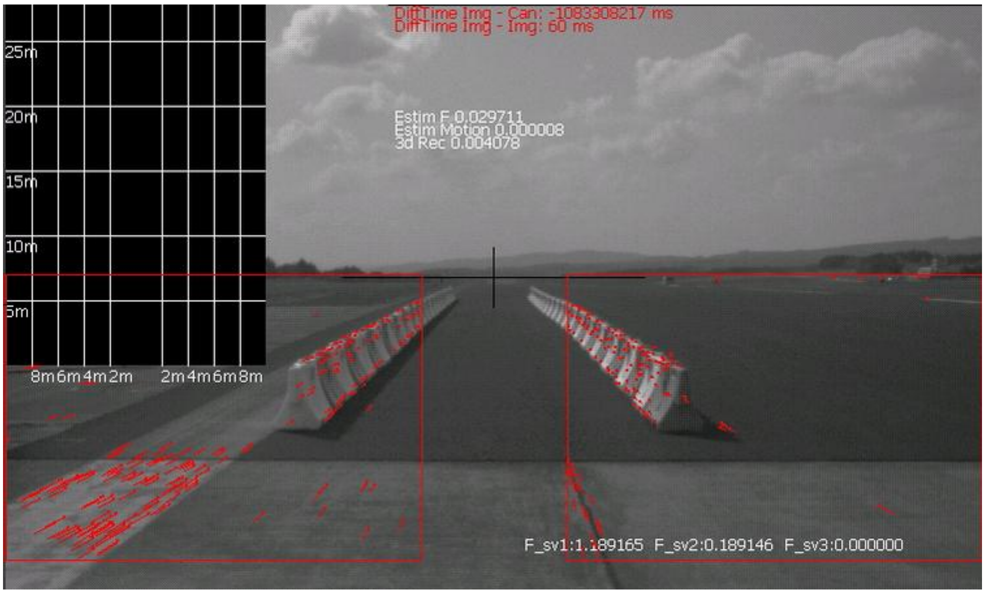


Figure 199: Optical Flow

Using all flow vectors, which result from stationary obstacles, a 3D picture of the surrounding is calculated. For this, a very exact knowledge of the camera movement between two pictures is needed to calculate the 3D position of the tracked features by using classic triangulation. Calculation of the Camera movement is done partly by using the inertial sensors of the vehicle. With this, translation and yaw angle could be estimated very well. For all other needed move-

[67] J.-Y. Bouguet, "*Pyramidal Implementation of the Lucas Kanade Feature tracker: Description of the algorithm*", Intel Corporation, Microprocessor Research Labs, 2002

ments and rotations, the epipolar geometry is calculated. It describes the geometric relationship between two pictures.

The more the features are displaced and the shorter the time between two pictures is, the more accurate is the calculation of the 3D representation of the surrounding. As it could be seen in Figure 199, the displacement of the features get smaller, the nearer they are to the centre of expansion. For the roadwork assistance this is not a constraint. The beacons and the guide walls are besides the own driving tube. Objects in front, like other cars are detected by the classic, model based extraction algorithms.

Raw data fusion

As shown in Figure 197, multiple paths for the environmental sensing exist, even for one sensor. The ARS300 radar for example is used in two ways. The first path starts with the classic object list of tracked objects. The objects are fused on object level in a Kalman filter together with objects detected by the camera. Those could be cars, trucks or motorcycles.

For the ARC demonstrator, another approach is used in parallel. All points from the 3D point cloud with a defined minimum height are mapped into a stationary occupancy grid. Also all radar reflections are mapped into this grid. In the occupancy grid the vehicle surrounding is fielded into stationary cells ($m_{x:y}$) with a constant edge length. For the radar $m_{x:y} = 1$ means, that at the place of the cell, radar energy is reflected. $m_{x:y} = 0$ means, that there is no radar energy reflected. In case of the 3D scene reconstruction from the camera $m_{x:y} = 1$ means, that there is an object with a certain height. $m_{x:y} = 0$ means, that the points from the 3D reconstruction are too low to interpret them as not drivable. Under the assumption, that the cells are independent from each other, the occupancy could be estimated with a Bayes filter. Afterwards, the lane boundaries could be estimated.

Unlike a classic occupancy grid (Figure 200), objects like bots dots or turfs are also recognized, because they have a certain radar reflection. Those would then be detected as a wall, not as a line. This failure seems to be acceptable, because it is only a failure in the type of the object, not in the object itself. The classification could be corrected later on, with the information from the 3D reconstruction.

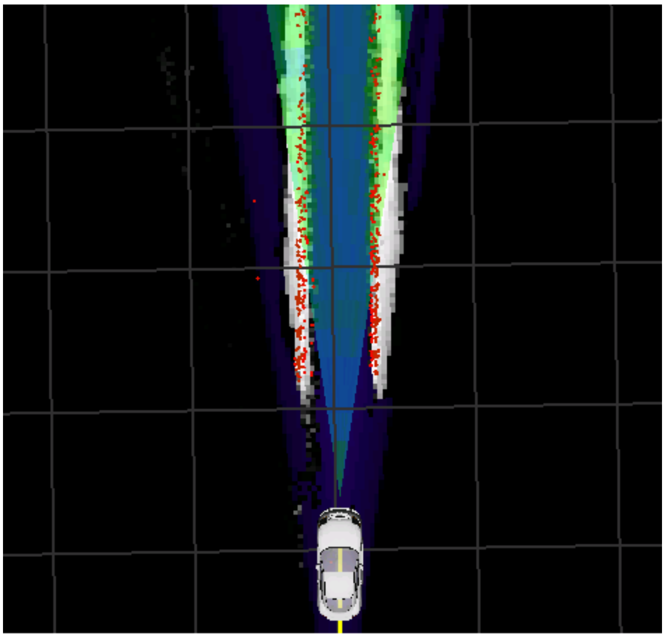


Figure 200: Occupancy Grid

Sensing lane markings is basic for lateral control on normal streets, like highways and countryside highways. In roadwork areas, this information is needed for separation of the single lanes.

Boundary estimation

The lane boundaries are estimated from the occupancy grid. The relevant area for the boundary detection in front of the vehicle is transformed from stationary to vehicle fix coordinates. Like every other lane marking the boundaries are detected by evaluating the leap between empty and occupied grid cells, like it is done for lane markings with the leap between dark and bright pixels.

Two valid areas for measurements exist, one on the left front and one on the right front of the vehicle. If there are two ore more valid measurements, the nearest to the middle of the lane is taken. As for the series lane detection a Kalman filter with a clothoid model is used for the estimation of the lane boundaries. Parameters for the clothoid are lateral distance to the vehicle, angle to the vehicle and curvature. The change of the curvature is not estimated, because the sensor accuracy in high distances of the radar and the detection range of the camera is not sufficient. But this parameter is not needed for speeds below 80km/h, so the chosen parameter set is well suited for construction sites.

As for series lane marking detection, the estimated clothoid parameters are used to continuously adapt the measurement areas. Observing the measurement points associated to the tracked clothoid, it is possible to detect a start and an end of the boundary. This information is very useful for the Situation Assessment and trajectory planning.

Additionally it is possible to estimate the detection quality by accumulating the amount of measurements associated to a lane boundary. Due to the fact, that there are not only continuous lane boundaries, but also beacons used in construction areas to separate the lanes from the roadwork area, the boundary estimation is also able to handle lane boundaries with very few measurements.

At the end, the tracked lane boundaries are handed over to the generic object list. Together with all other information like vehicles, speed limits and lane markings, the boundaries could also be used for other functions like an emergency brake assist. For example an earlier brake intervention is appropriate, if avoiding collision with the obstacle by steering is not possible.

Rear sensor data fusion

The rear sensor fusion aims in a combination of radar sensors with different field of views (see sensor coverage marked blue in Figure 198) and different time responses. This fusion has to be implemented in real-time on a VPU (Versatile Processing Unit). The rear sensor subsystem consists of a LRR (Long Range Radar sensor) ARS300 and two short range BSD radar sensors. These sensors supply the VPU with tracked objects. For further processing, the sensor data have to be fused. This part of data fusion was developed by partner University of Amberg-Weiden (UAM).

The target of the rear sensor fusion is the combination of all sensor objects, received by the connected radar sensors. The result is an object list, whereby objects received from overlapping areas of the sensors are merged, if they are duplicates. Thereby the selection of objects relevant for the driver assistance system is intended to be simplified and the data overhead is reduced.

The rear radar sensor fusion is done in two steps. The first step is a hybrid fusion via a logical templating; the second step is a sorted neighbourhood fusion[68], which can be processed very fast. Using a Kalman filter for object estimation as an alternative would be improper, because the overhead is too large and is not necessary for a track-to-track processing.

The logical templating is clustering the sensor objects by the decision, whether an object is located in a defined area or not. As shown in Figure 201, the pre-fusion of the BSD objects checks the overlapping area, of the left and right sensor for including objects, inside the cyan coloured line. In the second step of the fusion, the ARS300 and the BSD sensor are also filtered by the logical templating, as shown in Figure 202. The objects have to be inside the red line.

After this step the resulting objects are forwarded to a sorted neighbourhood algorithm. To assure a minimum of processor and memory cost, the sorting is done in a single iteration. The sort key is defined by the attributes for the search window. Depending on the size of the window, the sorted neighbourhood algorithm finds more or less objects inside the window. In our case, the search window is defined by the tolerance cells of the radar sensors. All objects are compared, whether inside their tolerance cell is a neighbour or not. If an object has a neighbour, it will be assigned to a sorted list of duplicates. If an object has no neighbours is can be copied immediately to the outgoing object buffer.

The last step combines the duplicates, whereby the mean values of the position and speed of the doubles are calculated. These steps could be calculated with following iterations, but it is not necessary because of the high quality of the provided radar sensor objects.

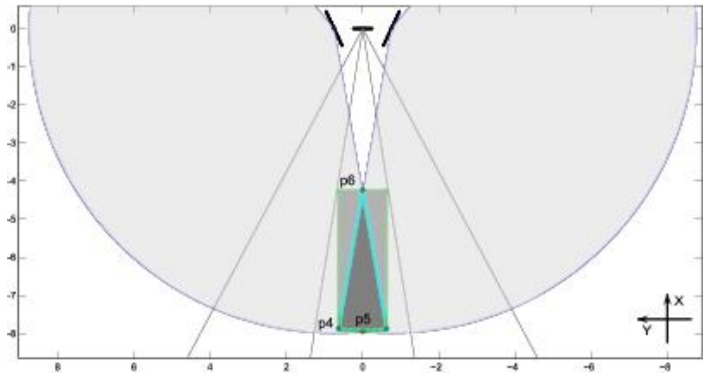


Figure 201: Overlapping area of the BSD-sensors

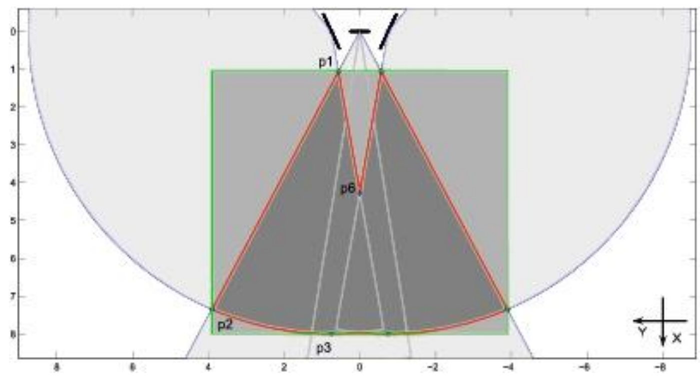


Figure 202: Overlapping area of the ARS300 with left and right BSD sensors

[68] M. A. Hernández, S. J. Stolfo: The Merge/Purge Problem for Large Databases. Technical report. Department of Computer Science, Columbia University, New York, 1995

Command layer

The general command layer modules are similar to the ones described in sections 5.1-5.3 and 6.1 above. The demonstrator specific realisation of all these modules and their integration into the demonstrator vehicle are described in detail in D51.3. As the ARC application from the functionalities point of view differes from other HAVEit vehicle applications, there are specifities in the Mode Selection and Arbitration Unit (MSU).

A use case catalogue was set up with all situations that are relevant for the ARC. Many driver assistance systems where already developed in Continental before. They were developed to handle critical situations and with this they are the basis for highly automated driving. Resulting from the possibility of hands-off driving in the highly automated mode, further functions had to be developed. All functions now available in the demonstrator are:

- Lane Departure Warning (LDW)
- Virtual wall (vWall) including Heading Control
- Lane Keeping Assistance System (LKAS)
- Lane Change Assist (LCA) including blind spot surveillance
- Emergency Brake Assist (EBA) including distance warning
- Full Speed Range ACC Stop & Go reacting on vehicles in front, static vehicles, speed limit signs, curves and narrow lanes.

All functions are described in section 1.1 of deliverable D51.3. With all these functions running in parallel an additional software module is needed to handle all the states of the sub-functions as the driver could not handle 12 different switch states. Therefore, the Mode Selection and Arbitration Unit (MSU) was developed together with partner DLR. Several optimization loops and psychological tests of the several HAVEit systems in a driving simulator where necessary to develop the MSU shown in Figure 203. It supports three main states: Driver Assisted (Figure 204), Semi Automated (Figure 205) and Highly Automated (Figure 206).

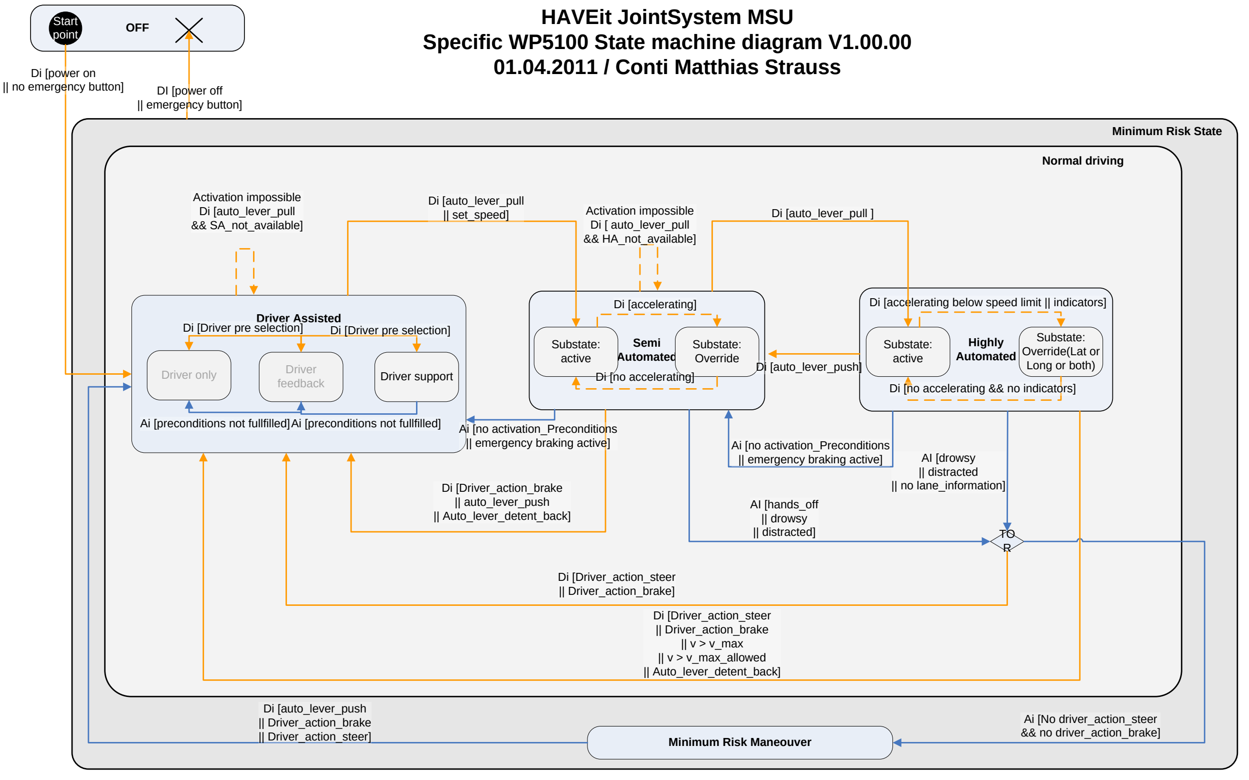


Figure 203: WP5100 Mode Selection Unit

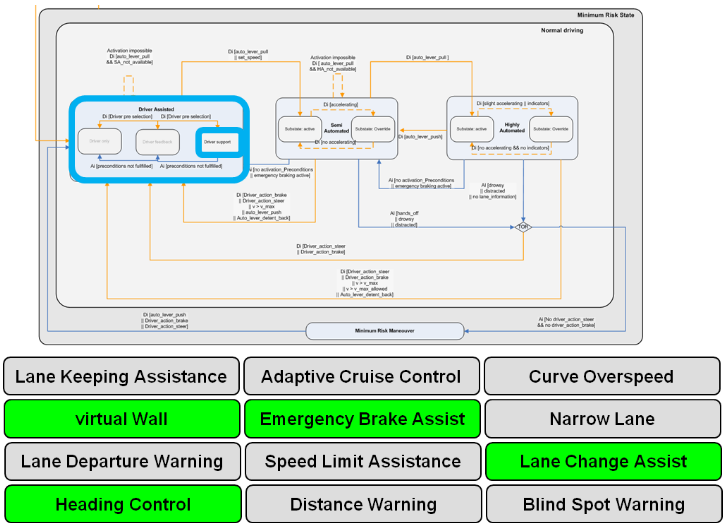


Figure 204: Mode Selection Unit with state: Driver Assisted

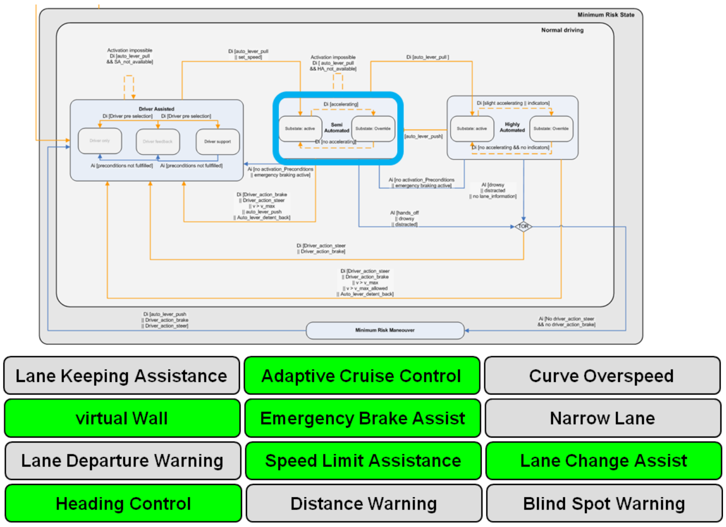


Figure 205: Mode Selection Unit with state: Semi Automated

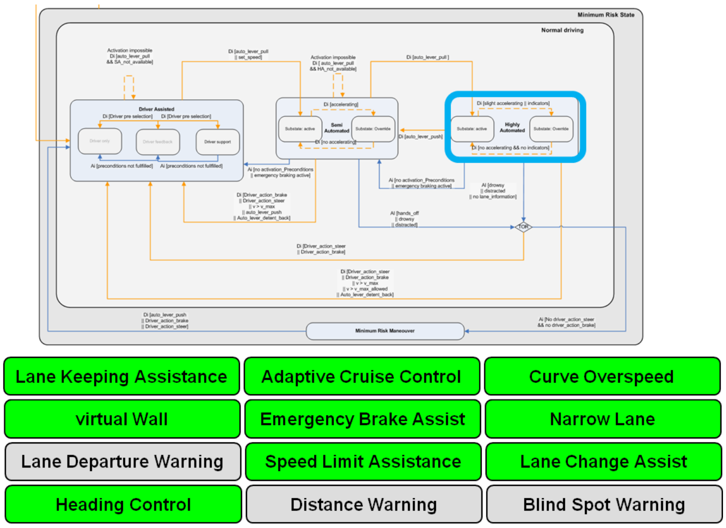


Figure 206: Mode Selection Unit with state: Highly Automated

The mode Driver Assisted is split into three preselectable sub states: “Driver only” (no support at all), “Driver feedback” (only acoustical and optical warning) and “Driver support” (acoustical, optical and haptical warning including Emergency Brake Assistance and virtual Wall). These sub-states could be selected by the driver, whereas Driver Support is the standard selection. In current vehicles the preselection of these three states could be integrated into existing Sport/Normal/Comfort mode concepts.

In general the driver is always able to switch the modes if all preconditions are fulfilled. This is mainly a working system and valid lane information for the Highly Automated mode. If an emergency brake situation occurs, the system switches back automatically to Driver Assisted.

If the driver is drowsy or distracted and doesn’t react on a takeover warning, the system switches into the Minimum Risk State, where the vehicle is stopped autonomously while the lateral control stays active. The same will happen, if the driver tries to drive hands off in Semi Automated or if there is no lane information is available in Highly Automated mode.

In all modes, the driver is able to take back control of the vehicle. If he steers with a defined torque or if he accelerates with at least 90% acceleration pedal position during an emergency braking, the control is given back to the driver. In the case, that the driver takes over control or tries to drive faster than allowed, or tries to make a lane change without using the indicators, the system is automatically switched to Driver Assisted. The same happens if the Minimum Risk State ends.

Execution layer

The execution layer, in particular the developed longitudinal and lateral controllers that consider the dedicated roadwork assistance requirements as well as the modified steering system, is described in detail in deliverable D51.3.

HMI elements to integrate the driver in the automation loop

The HMI concept was well harmonised between all vehicle owners in HAVEit. Specifities of the ARC demonstrator HMI elements are described in detail in D51.3. In order to avoid repetitions, the description of the ARC HMI is not included in this document.

7.1.2 System validation

In deliverable D11.1 a first use case catalogue was developed that was further extended and detailed throughout the course of the project. Several use cases apply to a couple of HAVEit demonstrators. These shall not be reported here, as they were all successfully passed by the ARC application and led to similar results as for other demonstrator(s). Instead, this section focuses on scenarios specifically relevant for the automated roadwork assistance application. For the ARC demonstrator, the hands-off driving through complex scenarios (roadworks) is the most important function.

Scenario “Speed limitation”

The speed limit detection is series software and runs on the camera. It delivers its output on the private CAN. The detected speed limit sign is read out. If a speed limit sign is detected, the ACC set speed is automatically adapted. Figure 207 shows an automatic adaption of the ACC set speed after a speed limit sign (70km/h) is detected. In this situation no vehicle was in front (red). At the blue marker line, a new speed limit sign is detected (magenta). In this moment, the ACC set speed is set to the detected speed limit.

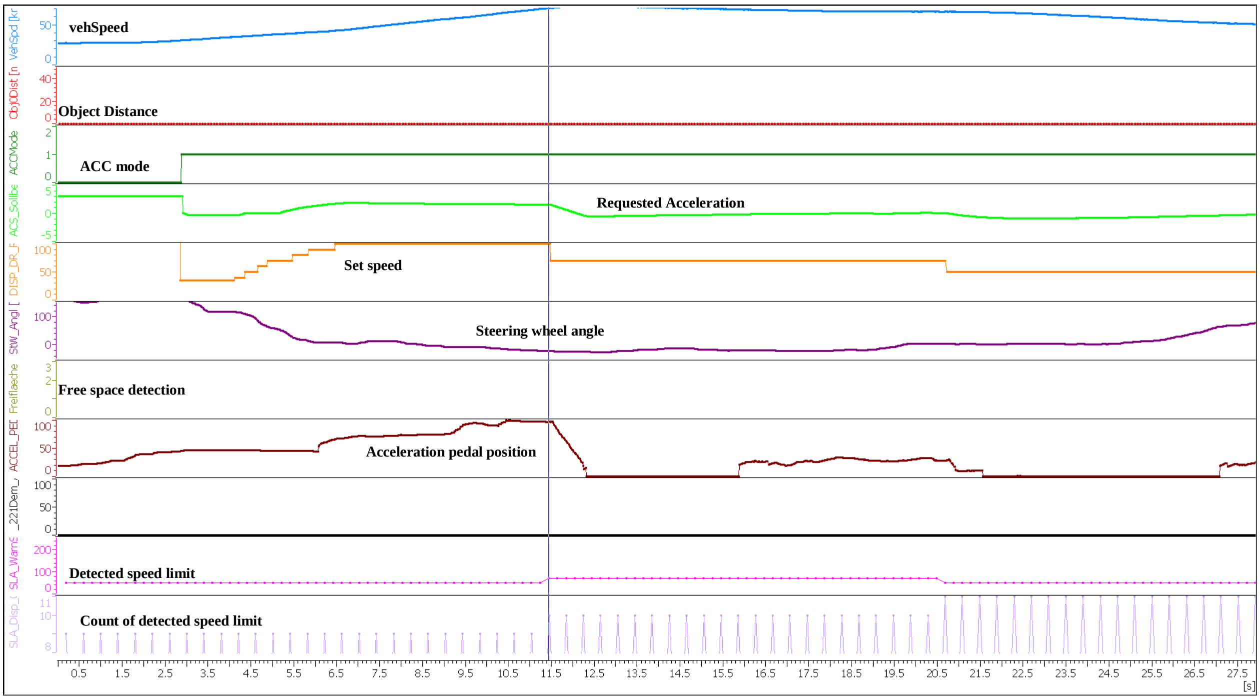


Figure 207: ACC set speed adaption by SLA without vehicle in front

The ACC acceleration requests (light green) switches from a positive value, to a negative one, to decelerate the vehicle to the current speed limit. After that, the speed is held, until a curve comes and the speed goes down, because of a high steering wheel angle. The same test setup was done with a vehicle in front. As only the set speed is adapted and no speed request is given to the ACC controller, the vehicle still adapts its speed for more critical objects like vehicles.

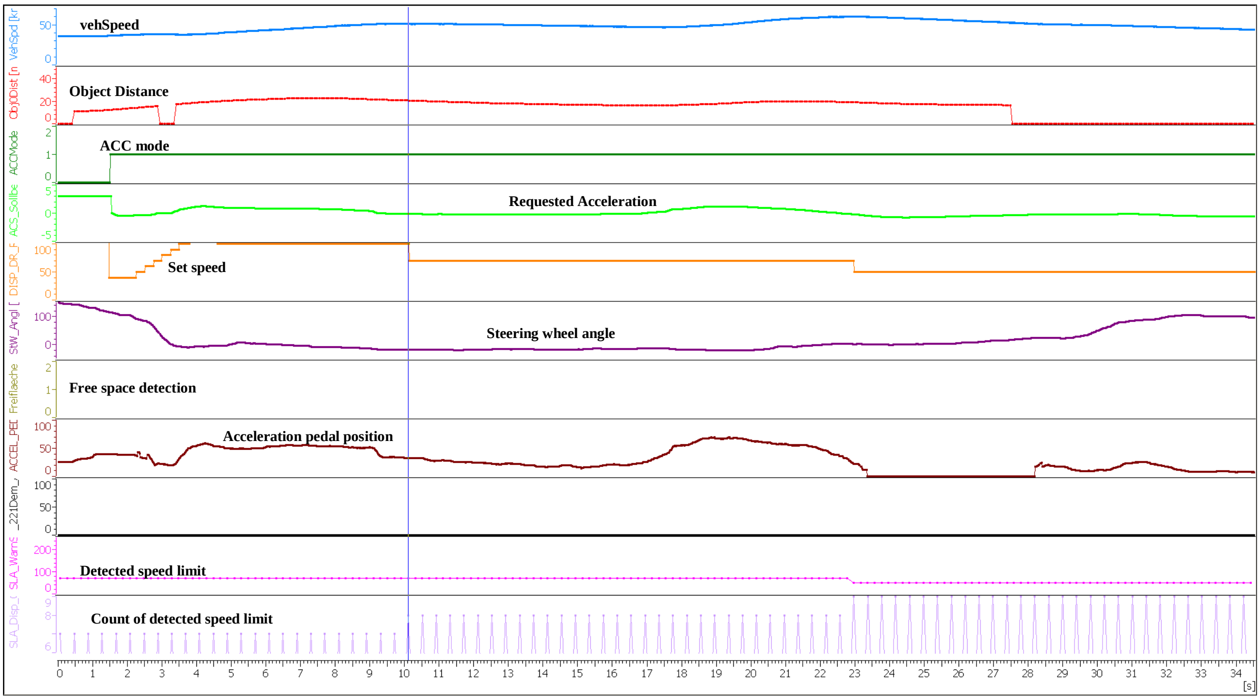


Figure 208: ACC mode with SLA and vehicle in front

In Figure 208 one can see that the vehicle speed is below 60km/h even if a speed limit sign of 70km/h was detected. With the detection of the 50km/h speed limit sign (time = 23sec), the vehicle is decelerated to this speed. This could be seen at the brown line. The acceleration request is set to zero until the desired speed is reached. The main problem of an automated adaption of the ACC set speed to the detected speed limitation are speed limit signs, which are not valid for the own lane. Those are speed limits from streets nearby the own or speed limit signs on trucks.

In series systems, the driver has to accept the new set speed by pressing the set button on the ACC lever. This is not recommended for the Highly Automated mode. It is assumed, that a slower speed than allowed is better for safety. Otherwise there is the risk of driving fast inside a roadwork area, which could lead to situations, where the needed steering torque is higher than 4Nm or the sensor range is too small to detect all relevant objects in time.

Scenario "Presence of an adjacent road"

As described, series sensor software is used to detect the lane markings. The presence of an adjacent road is also a topic in the series development. Until now, no mis-interpretation from the series software occurred. In Figure 209 the test track in Pferdsfeld is presented with a typical deceleration lane of a German highway. The next diagram in Figure 210 shows the measurement from the Lane Centring Assist. Over the whole measurement, the vehicle was held inside the lane and no misdetection has taken place. All lines are continuous and no jumps could be detected.

Figure 209: Adjacent road on test track

Scenario "Driving in Highly Automated mode"

This is one of the most important scenarios for the HAVEit project. All functions have to work together and the complete driving task is handed over to the system. In highly automated mode, ACC is activated together with Lane Centring Assist and vWall.

The lateral position inside the lane is presented in Figure 211. The red upper and lower lines represent the detected lane marks. It has to be regarded, that the measurement is quite compressed. The length is more than 60m long and the width is only 6m. The green line is the centre of gravity of the vehicle. The black line besides is the middle of the lane. Caused by the slope of the test track, there is a small (0.1m) offset between the vehicle centre and the middle of the lane. The two blue lines are the edges of the vehicle. It could be seen that the vehicle is held in the middle of the lane without any problem.

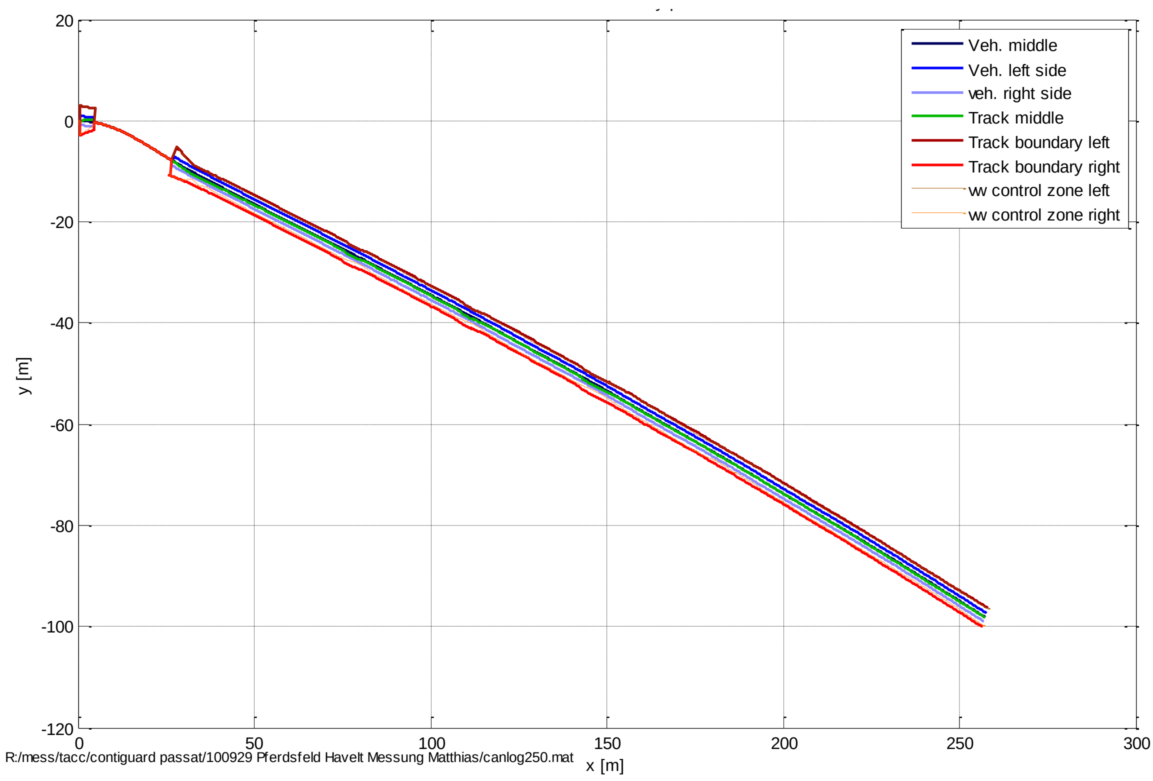


Figure 210: Measurement of adjacent road

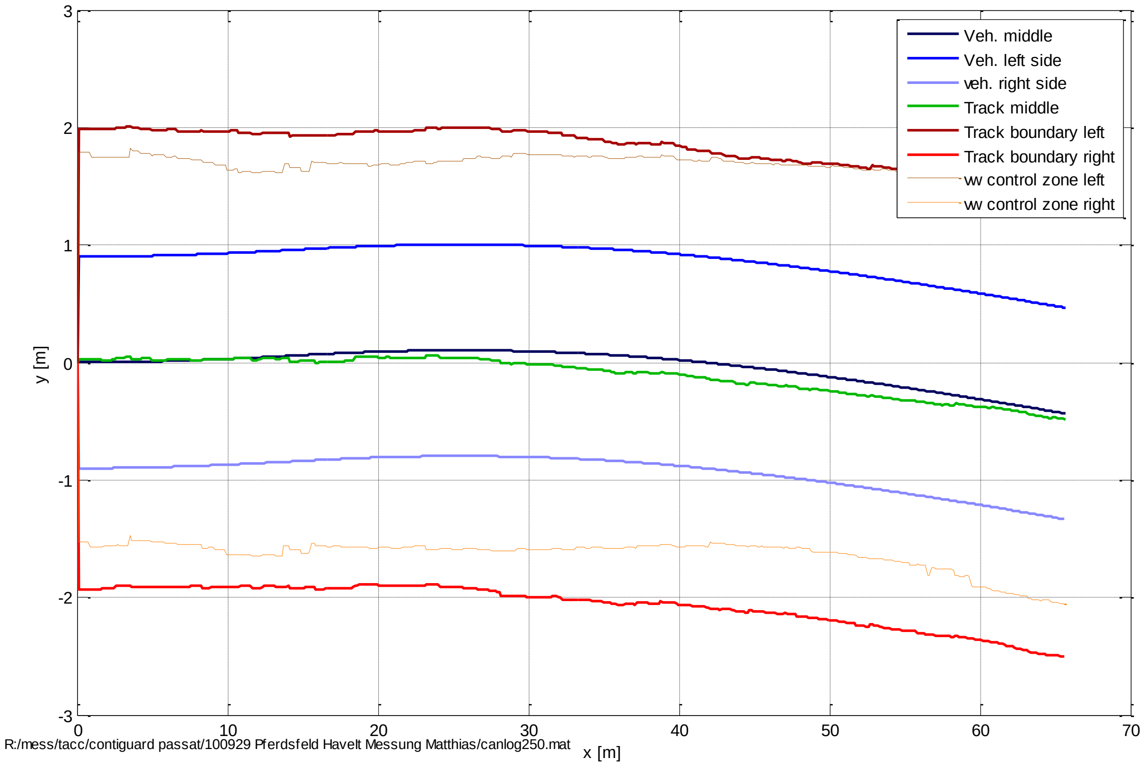


Figure 211: Lateral control between lines with hands off

While the lateral control was active, the ACC system was also switched on. The set speed was 50km/h. The speed characteristic is presented in Figure 212. The upper line is the current speed of the vehicle. It increases until it reaches 50km/h. After that the speed is held. The real speed of the vehicle is a little bit below the set speed. The set speed is adapted, to the speed shown to the driver in the instrument cluster. The speed shown there has to be a little bit higher than the real vehicle speed by law. This law should prevent the driver from driving faster than allowed.

The lowest line is the status of the ACC system. The ACC was switched on all the time. The middle line in Figure 212 represents the drivers steering torque. It is below 1Nm and reacting very fast. This is the indication that the driver has his hands not on the steering wheel. Otherwise the characteristic would be much smoother and the torque would be even higher.

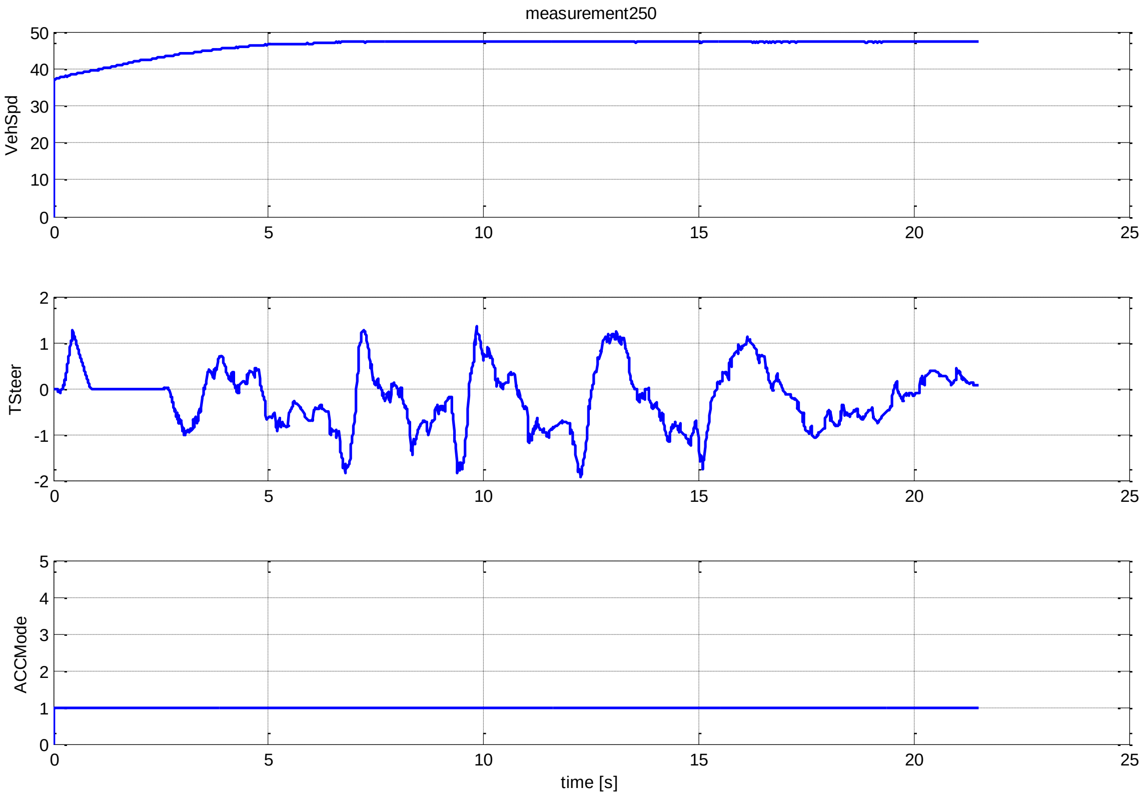


Figure 212: Longitudinal control while driving hands-off between lines

By this measurement it is proven that driving in Highly Automated mode is possible in the ARC demonstrator. With the results from the Mode Selection and Arbitration Unit also the transition to Highly Automated is implemented in the vehicle.

Scenario "Driving through a roadwork area"

Driving through a road construction is the most important scenario for the ARC application. The lateral control has to work with the lane boundaries, detected by raw data fusion of series sensors. In this scenario shown below, the Semi-Automated mode is activated. This means ACC is switched on and the vWall function is active. The driver drives through the road construction with his hands on the steering wheel. The longitudinal control is with the ACC and the driver gets a counter torque in the steering wheel, when he drives too close to an object besides. Then the automation mode is switched to Highly Automated. The driver now takes his hands off the steering wheel. At the same time, the ACC still controls the vehicle in longitudinal direction. By this, the driver could drive through the road construction, without using his hands or feet.

For a better understanding, the scenario is shown in Figure 213. First, the speed limit sign is detected, than the vehicles drive into the road construction. The truck is the only right boundary. There are no lines in the construction site.

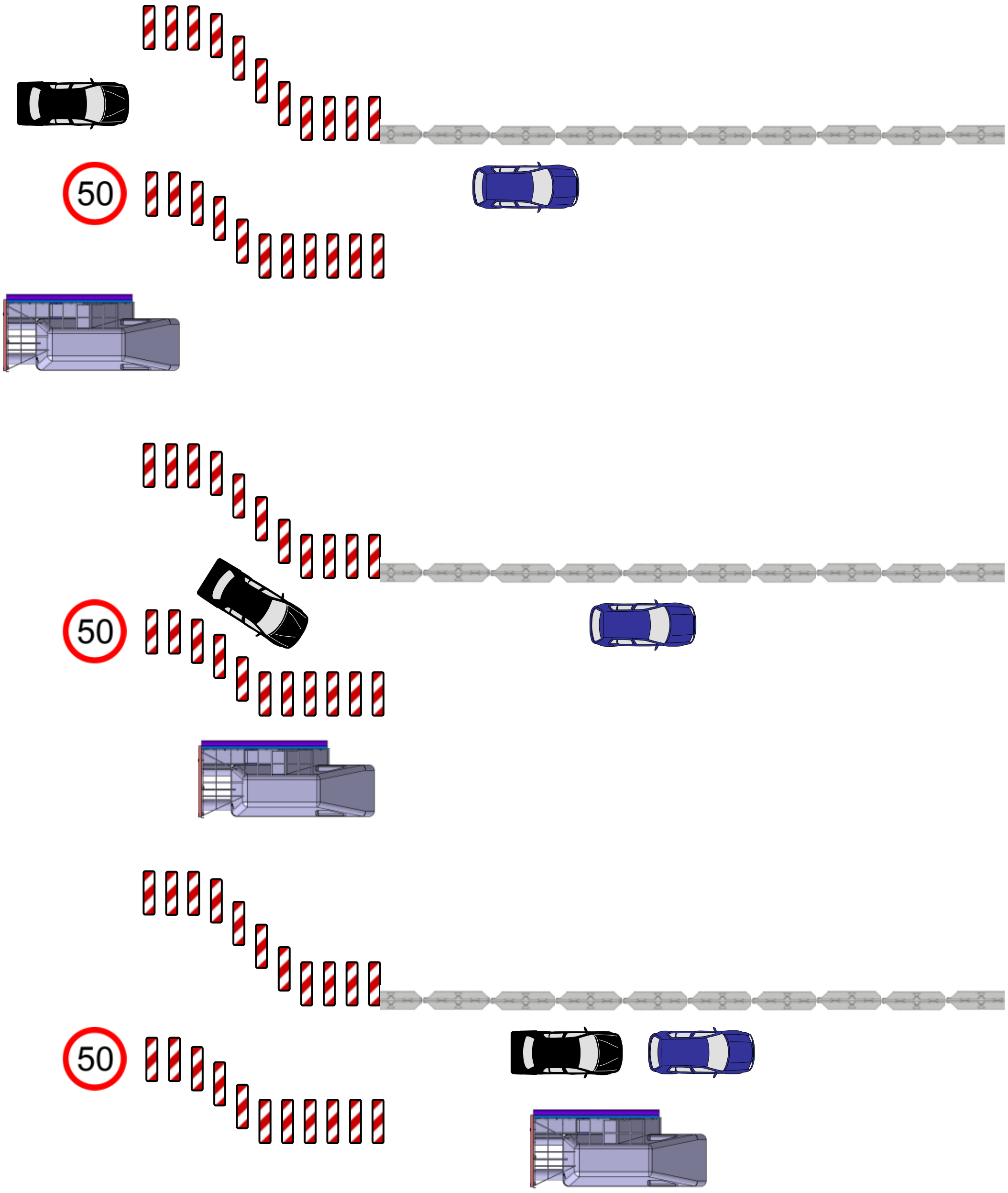


Figure 213: Scenario: driving through a road construction

Driving through the road contruction site Semi-Automated

In Semi Automated mode the ACC and the vWall is active in the ARC demonstrator. The driver has to hold his hands on the steering wheel. In this mode, only one line or guide wall has to be detected. There is no need for holding the vehicle between two lane boundaries. If the vehicle gets to close to a boundary, a line or another vehicle, a counter torque is superposed to push the vehicle back from the dangerous zone. This scenario was driven without a vehicle in front.

This has to work for straight boundaries also as for curves marked with beacons. In Figure 214 the position of the vehicle regarding the road construction is presented. The vehicle entered the road construction, without any lane markings before. Leaving the beacons behind (at 150m) only the wall on the left side is remaining. The driver has driven through the entrance himself. ACC was activated. In the right turn he was instructed to “forget” to steer. As the vehicle comes to close to the beacons, the vWall algorithm pushed the vehicle back into the middle of the lane.

While driving besides the guide walls, the driver was instructed to steer the vehicle against the guide wall and then hold the steering wheel. In this situation a counter torque from up to 4Nm was superposed to push the vehicle back from the guide walls. If then the driver steers away from the wall, the intervention was stopped immediately and started again, when the driver changed the driving direction of the vehicle onto the guide wall again. The steering torque characteristic is shown in Figure 215.

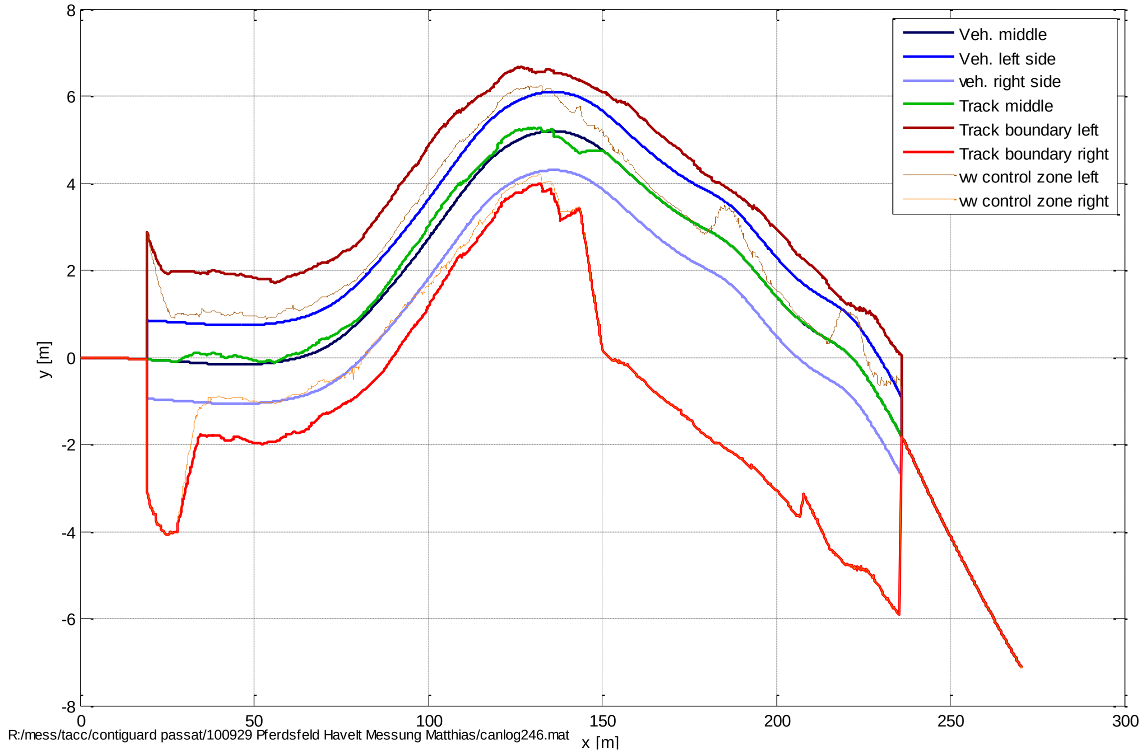


Figure 214: Driving through a road construction in Semi Automated mode (position)

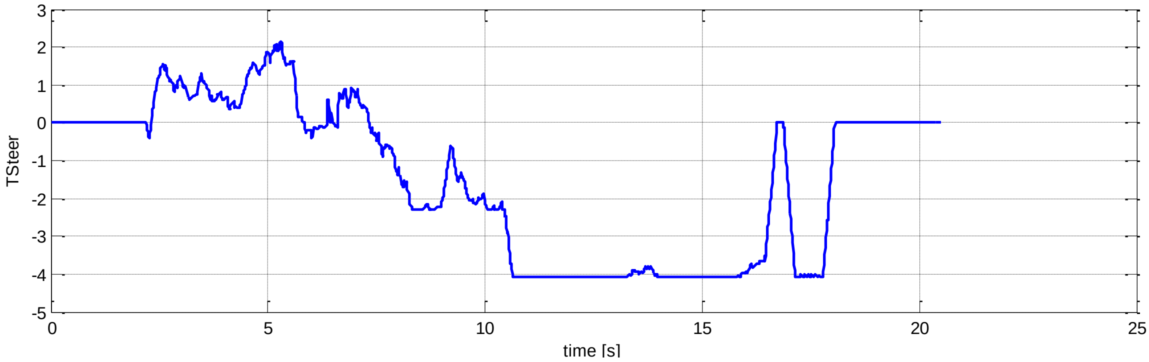


Figure 215: Driving through a road construction in Semi Automated mode (torque)

The same was done with the truck simulator driving besides the ARC demonstrator. Now a lane boundary is given on both sides for the whole length of the road construction. The driver was instructed to hold the steering gently. By this the vehicle was pushed from one side to another. This is not a realistic driving scenario, but it shows quite clear, how the system is working.

In this situation the steering torque requested was short intervention as it could be seen in Figure 217 in the upper blue graph. The steering torque from the virtual wall function was now lower, than in the mode without the vehicle. This is because the algorithm computes its intervention on both objects. With this the vehicle won't hit the object on the opposite side.

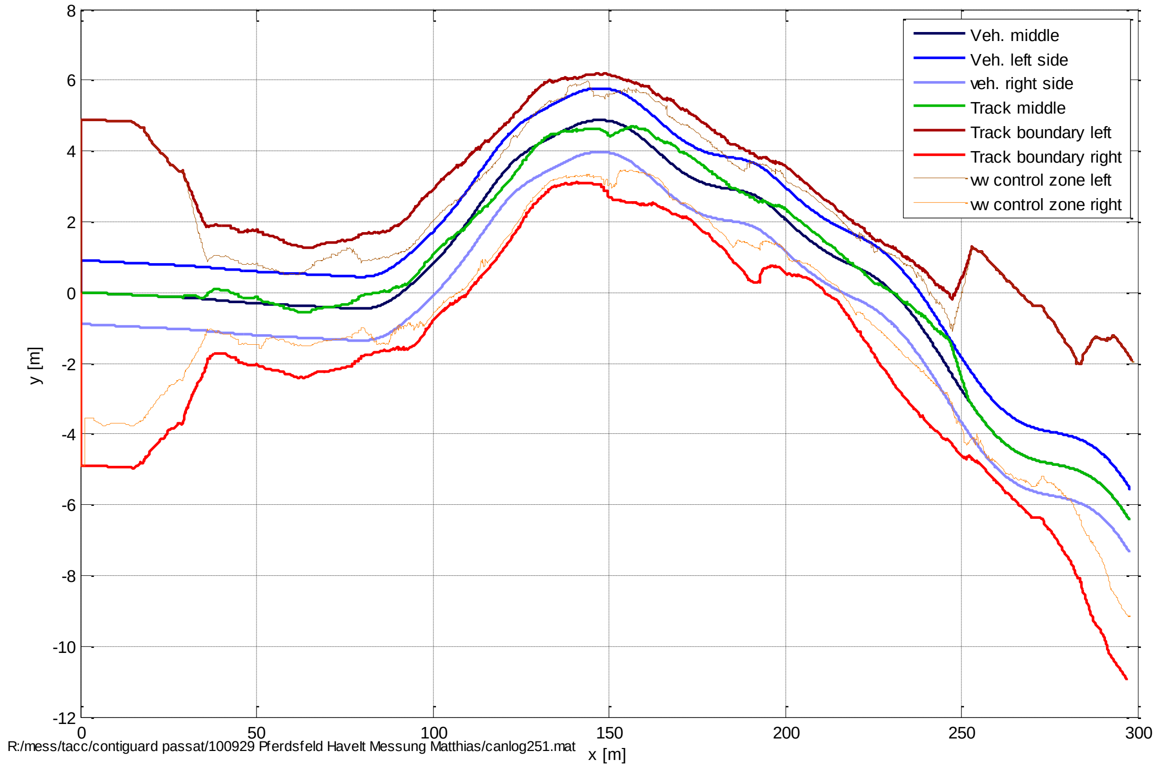


Figure 216: Driving through a road construction in Semi Automated mode with truck (position)

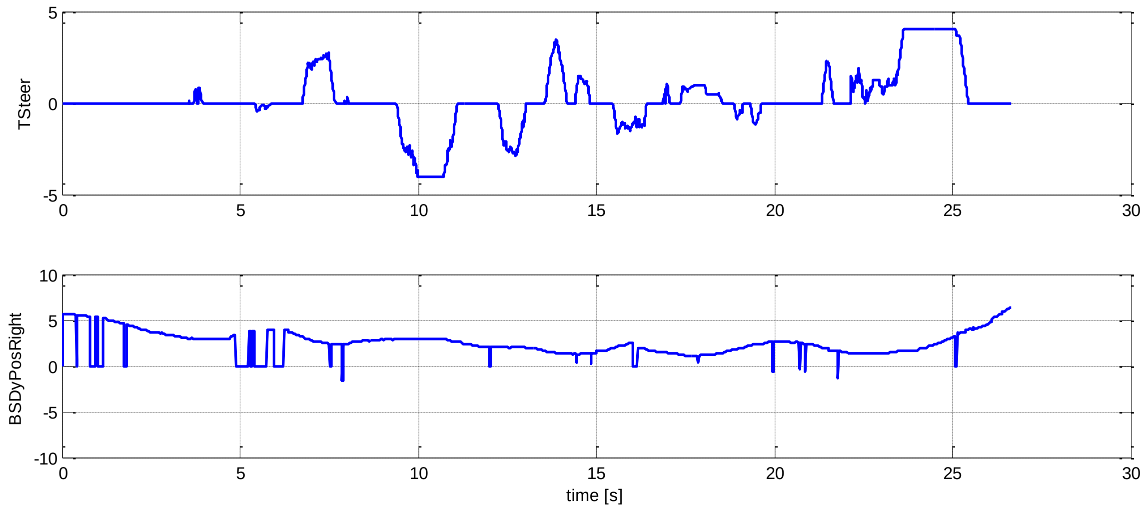


Figure 217: Driving through a road construction in Semi Automated mode with truck (torque+SRR)

In the lower graph in Figure 217 the measured distance to the truck is presented. There are several losses of the object and some jumps in the distance. Due to this, the signal is filtered and losses have to be ignored. By this, the lower red line in Figure 216 is acquired for the section, were no line or guide wall is on the right side of the vehicle. It is obvious, that the signal is not very well. For this reason the control algorithms are adapted to the distance to the vehicle. The yaw angle and the curvature are attained from the static guide wall.

Driving through a construction site in Highly Automated mode

The most challenging part for the ARC demonstrator is driving through a road construction site in Highly Automated mode. In this mode, the vehicle is fully controlled by the system. The longitudinal control is done by the ACC Stop&Go system and the lateral control is a combination of Lane Centring Assist and vWall.

For the test described below, the same scenario was build up as for the final event (Figure 213). In the first step, the Semi Automated mode is activated. Now the ACC system and the vWall function are active and as soon as the left and right lane boundaries are detected, the Highly Automated mode gets available. As soon as the highly automated mode gets available, it got active. This automation was done due to the short distances in front of the road construction on the test track.

Figure 218 shows all relevant values for the lateral control algorithm during the Highly Automated mode. The outer red lines are the detected beacons, guide walls and vehicles beside. The lighter red lines are the intervention zones for the vWall function. In difference to the vWall only (Figure 216) it could be seen, that the vehicle position (dark green) is held smoothly in the middle (light green) of the lane, without any dynamic interventions.

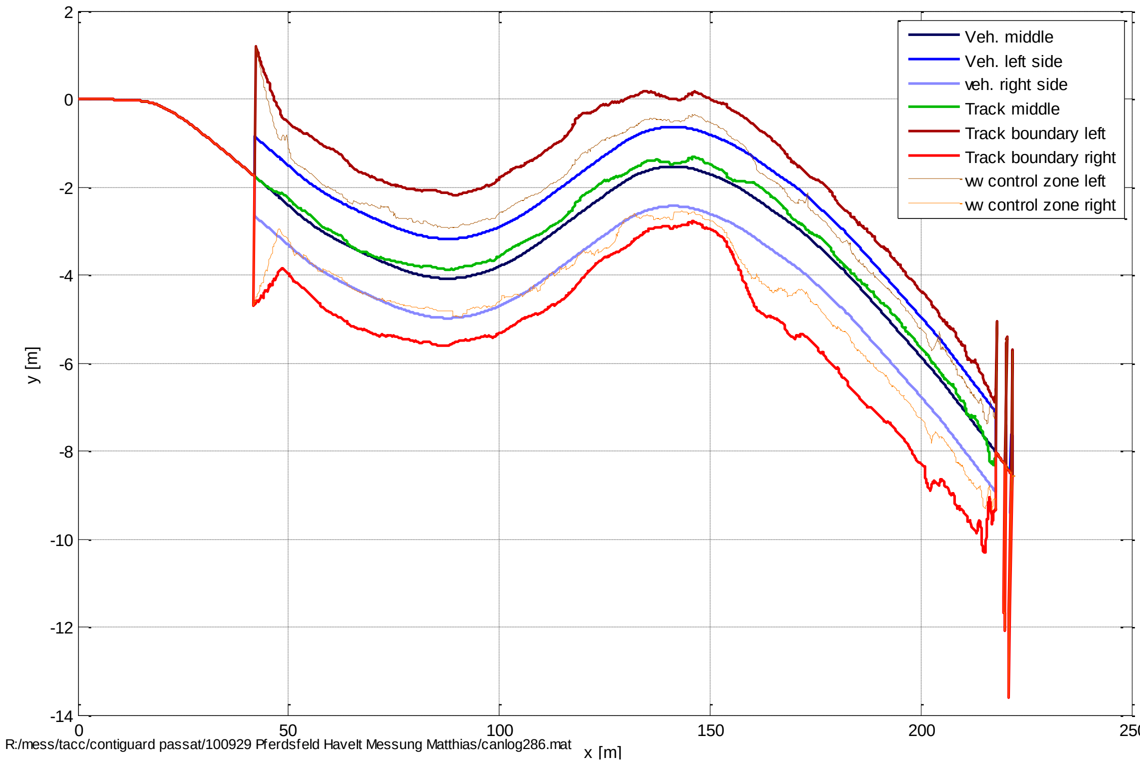


Figure 218: Lateral control path driving through the road construction in Highly Automated mode

At 150m the vehicle passes the last beacons. Now the lane is between the guard rails and the truck. The change from beacons to truck is filtered. It could be seen, that the truck is a little bit more to the right side than the beacons. Because the system holds the vehicle a little bit nearer to the guide wall than to the truck, the middle of the lane (light green) stays nearly at the same position.

In Figure 219 one can see, that there was no vWall intervention necessary. The steering torque stayed below 2.5 Nm. Besides the lateral control of the vehicle, the control algorithms have to be stable against driver interventions. When the driver puts his hands on the steering wheel the inertia of the steering wheel changes rapidly. On the other side, it goes down to a very small value, if the driver takes his hands off the steering wheel.

When this happens, the vehicle could become instable, because small torques on the steering wheel now lead to high reactions of the steering system and with that, also from the vehicle. To avoid this effect, the measured steering torque from the driver is taken into account by the control algorithms. In Figure 219 the measured torque is presented, it's the third blue line called TDrv. The signal is quite noisy, caused by the missing inertia of the drivers hands. But even with this the controller is stable over all the time.

The lateral control algorithm is switched of at lower (< 10 kph) speeds. When the vehicle comes to standstill, also the tracking of the guard rail objects stops (Figure 218). For the Stop&Go case inside construction areas, the directions have to be stored at a defined speed. After that, the lateral control algorithms have to rely on that data and the current data from the side and front radar.

That is not a big issue to realize, the vehicle starts from the standstill with a defined latency after the vehicle in front. With this it can be assured, that the field of view for the sensors in the vehicle gets big enough for detecting the guard rail objects again. In Figure 220 the signal from the side Short Range Radar sensor cluster is displayed (BSDyPosRight). The signal is still available in standstill. With this driving off without a collision with a vehicle besides is possible.

At the same time, as the lateral controller is active, the longitudinal control is too. The ACC system was switched on over the whole measurement time (ACCMode + VehSpd in Figure 219). The ACC system decelerated the own vehicle, because of the vehicle in front until a complete standstill of both vehicles. The distance between the vehicles is the lowest line in Figure 220. The line above is the detected speed limit sign. In Figure 219 it could be seen, that the ACC system holds the speed below 50kph. The driver switched on the ACC system on with a set speed of 60kph.

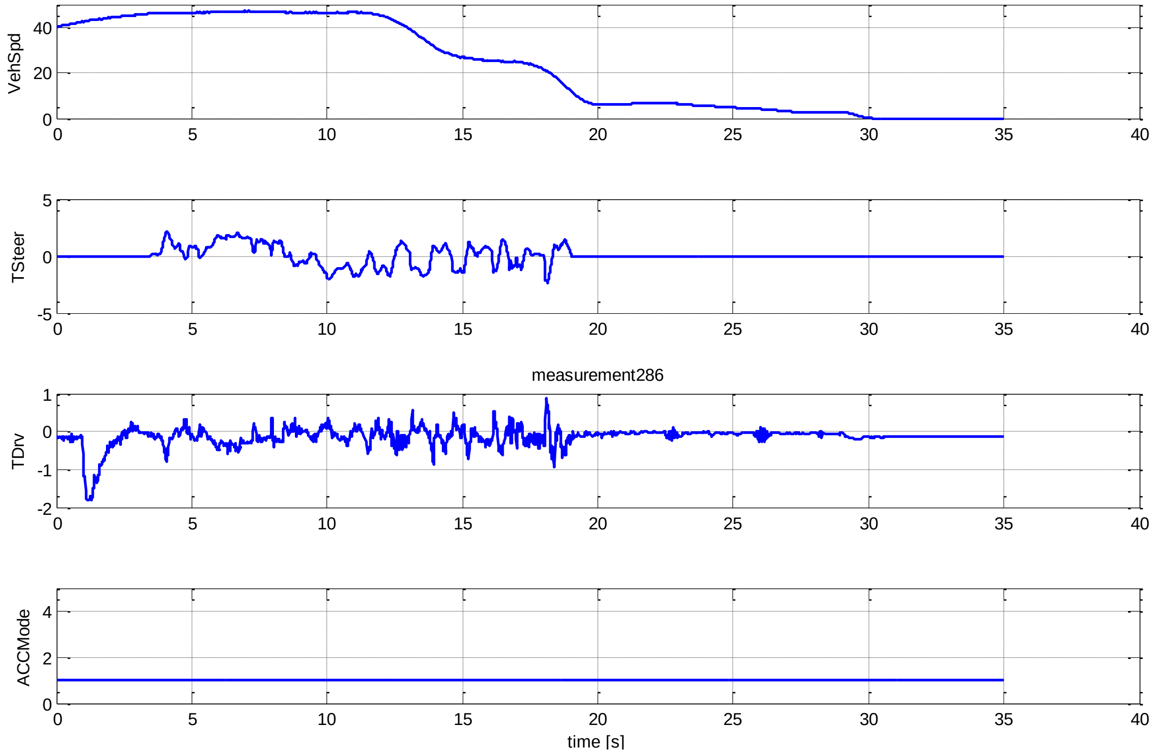


Figure 219: ARC internal parameters driving through a road construction in Highly Automated mode (1)

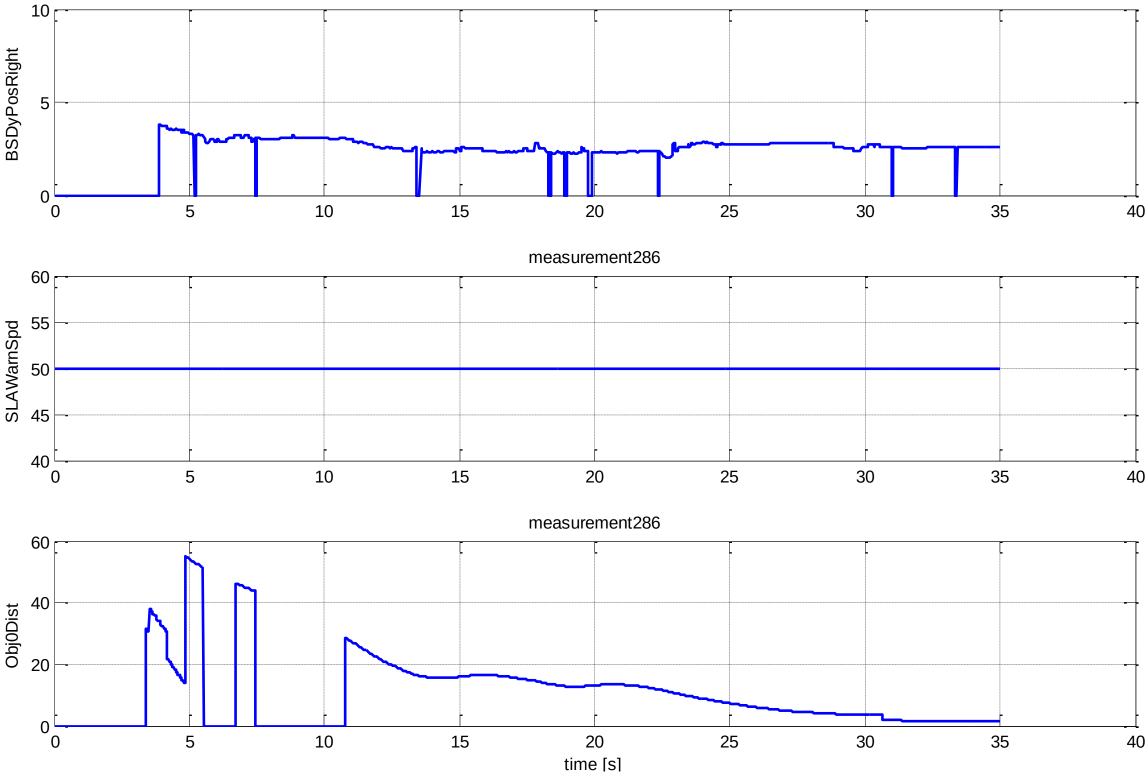


Figure 220: ARC internal parameters driving through a road construction in Highly Automated mode (2)

Scenario "Prevention of curve overspeed"

Usual ACC systems hold the set speed and don't accelerate, if the steering wheel angle gets to high. For Highly Automated driving an adaption of this speed in regard of curves is needed to prevent the vehicle from leaving the lane. For this reason the curve overspeed assistant was developed in HAVEit.

Currently, there is the limitation that the information about the path the vehicle will drive, is only available from the onboard sensors maximum 50m in front. For a good performance also at high speeds it requires information about the minimal radius the vehicle will drive and in which distance this minimal radius will be reached. Only with this information, the application can calculate a deceleration strategy at higher speeds.

Today, the path detected by camera and radar is modulated by a clothoid curve, where only the current curvature, not the change of the curvature could be used. For that reason, the application has knowledge about the currently driven curve. For driving at high speeds on curvy roads, this will not be applicable. But it is enough to ensure driving safety on highways and roadworks, even if the function is sometimes uncomfortable. With this it is still well suited for the HAVEit application.

If the path is valid for at least a defined time and its width is less than the maximum width of a lane, the current radius is calculated. In all other cases, the path is assumed straight. The minimal oncoming radius is calculated as follows: radius= 1/curvature. If the radius gets smaller, always the newer radius is taken into account. If the radius gets wider again, it is filtered. This is done to ensure, that a small radius will lead to a deceleration in time. For instance, this can look like in Figure 222.

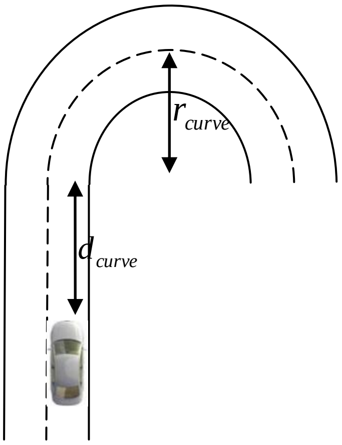


Figure 221: Curve overspeed scenario*:* The subject vehicle with the advanced HAVEit assistance system is driving on a road, which has one curve with a curve radius r_{curve} and a distance d_{curve} between the curve and the subject vehicle.

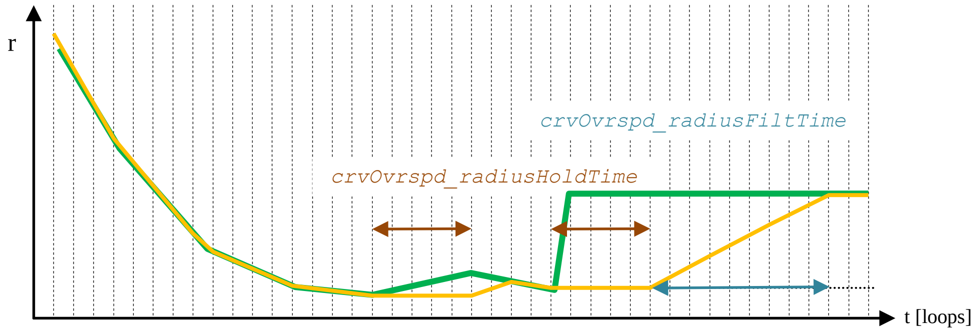


Figure 222: Curve overspeed radius calculation

The maximum speed for a radius is defined by the maximum possible steering torque. It was tested in the vehicle, which torque at several defined speeds will lead to which driven radius. With the maximum torque allowed in the controller (Table 35) the minimum curve radius could be determined at every speed. The maximum control torque has to be adapted to every speed otherwise the controller could get unstable.

Speed	maxTq
15	1,37
30	2,3
60	2,3
75	2,6
105	3
220	3

Table 35: Maximum Torque depending on vehicle speed

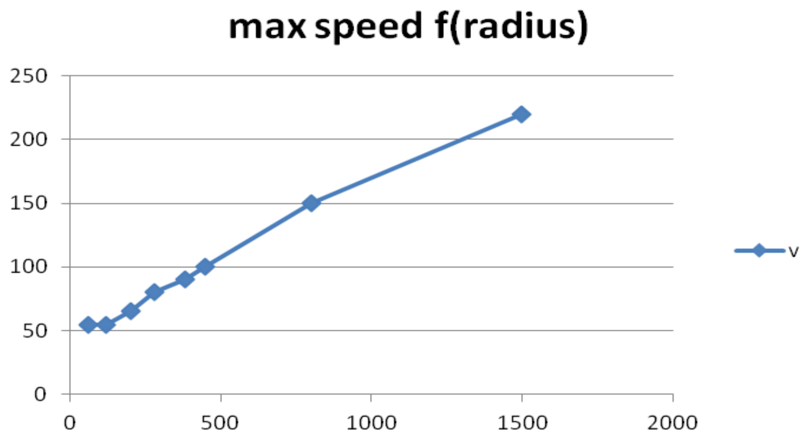


Figure 223: Possible speeds as function over radius

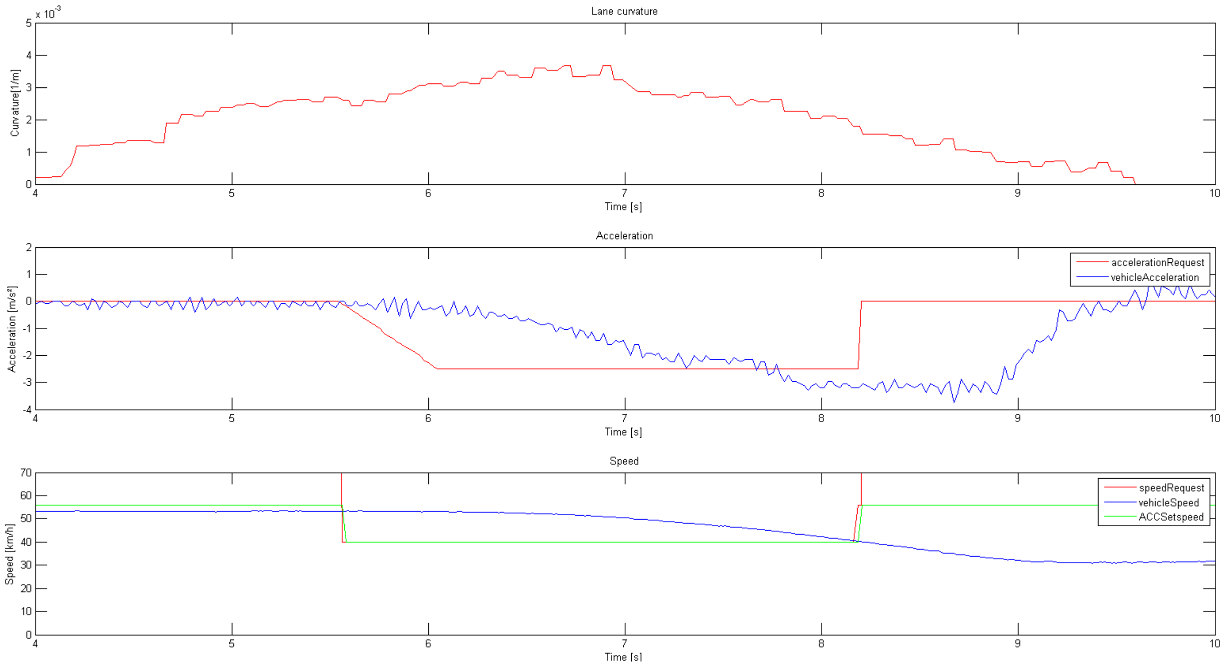


Figure 224: Curve Overspeed function in demonstrator

Figure 224 illustrates how the function is working in the demonstrator. The upper graph is the curvature. If it gets to high, a deceleration request is send out to the arbitration module (red line in second graph). With a time delay resulting from the series ACC control implementation, the vehicle reacts on this request. In further developments, the request will be handled by the active booster and will be much faster than now. In the current development state, the ACC will switch of, if the booster is activated.

Scenario "Presence of a vehicle in front, automated emergency braking"

This is a common situation that is almost guaranteed to occur when the driver gets into his car. If the driver is located on a highway or on a rural road – there is mostly a vehicle in front of him. Depending on the behavior of this target vehicle, the driver of the subject vehicle might have some difficulties. For example, if the driver of the target vehicle suddenly brakes, the driver of the HAVEit vehicle needs assistance to cope with the situation, which is illustrated in the figures below. Due to the narrow lanes in roadwork areas, the scenarios become very important for the ARC application.

The scenario of a vehicle in front is very similar to the use case of traffic jam. The speed of the own vehicle is adapted to the speed of the vehicle in front automatically, even until complete standstill. Current ACC systems just decelerate until a given maximum. This was 0.3g in the past and is now already 0.5g in some systems. If this maximum deceleration is reached, the driver gets a warning to support the system. Some systems even switch off in such situations.

This is not acceptable for Highly Automated driving. Therefore, the series ACC system has been improved to enable decelerations above 0.5g. Still, the system will try to decelerate the vehicle comfortably (<0,3g). Higher decelerations are only applied if necessary. For decelerations with more than 0.4g an additional brake pressure is superposed by the active brake booster. This is illustrated by the diagrams in Figure 226. The needed deceleration (light green) gets too high during the second brake phase of the vehicle in front (light red). When it exceeds more than 0.4g, additional brake pressure is superposed by the active booster (black).

When this happens, the ACC system switches off after 1sec (time=19.2sec). This is done, because an emergency brake intervention is a critical situation. After that, the driver has to switch on the system again. For Highly Automated driving this intervention will lead to a shift to the Driver Assisted mode.

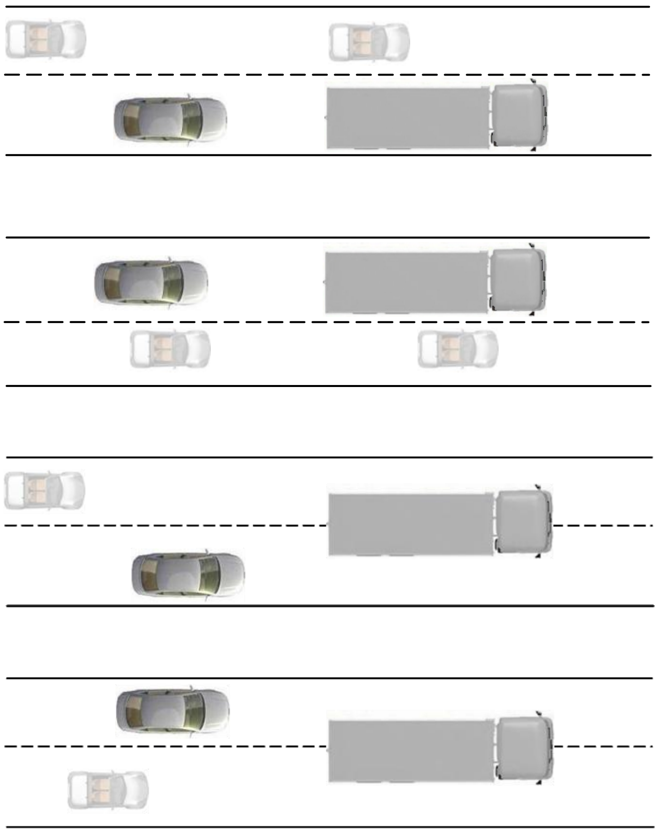


Figure 225: Emergency braking scenario in roadworks*:* There is a forward vehicle which partially or totally fills the same lane as the subject vehicle which is endowed with the HAVEit assistance system. There may also be another vehicle in the lane next to the forward vehicle or next to the subject vehicle.

The objects are detected with radar and camara fused on object level. These detected objects are also used for the ACC control. With this information ACC is able to react on a priori standing object vehicles at speeds up to 50km/h. At higher speeds, the deceleration gets to high and the ACC gets an emergency brake intervention. This intervention is limited due to the sensor performance to 70km/h.

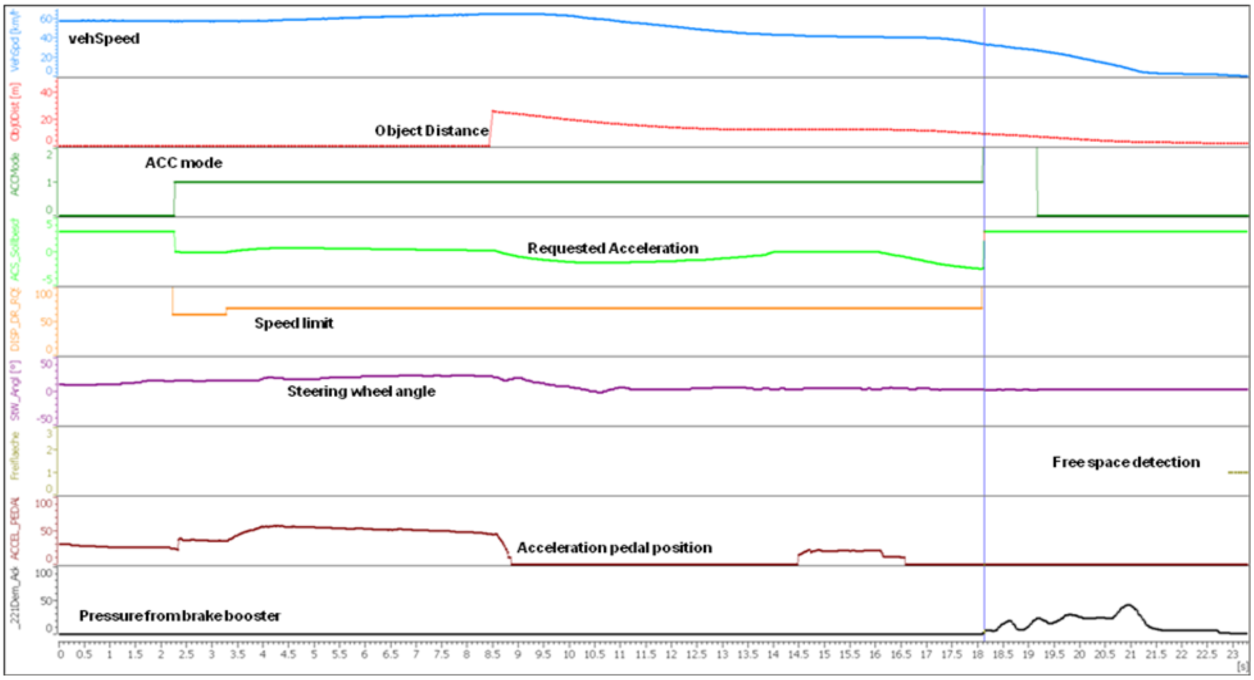


Figure 226: Emergency Braking on standing Obstacles

For moving object vehicles or object vehicles which had moved before, the possible speed is higher than 120km/h, because the camera is then not necessary for validation of the object type. In future, it has to be discussed and investigated, which standing objects have to be detected in which range, and what is the resulting maximum speed for automation then. If it would be necessary to also brake on generic objects, a stereo camera should be used. This special case is part of the INTERACTIVE project. The outcome could afterwards be used to enhance the safety of highly automated driving. The full spectrum of use cases relevant for ARC application has been successfully tested and described in detail in HAVEit deliverables D51.3 and D13.1.

7.1.3 Summary and conclusion

All hardware and key components for the ARC demonstrator were installed in the vehicle and integration tests where successful. The most critical task for the ARC application, the hands off driving through a road construction was tested successfully.

Figure 227: HAVEit ARC demonstrator and typical scenario with traffic jam while approaching a roadwork

With the start of the project it was obvious that the ARC demonstrator will present a very complex system consisting of many modules. The chosen hardware and software design proved its capability of such a high complexity with at the same time high computational power and minimum risk for the driver. All software components are as modular as possible. Existing functions have been integrated into the concept. Thus, it will be possible to use the outcome from the HAVEit project for the next projects inside Continental without high transfer effort.

In future, some modules that are currently running on PCs could be transferred to ECUs when they are in a high development phase. By this, the reliability of the system could be increased and the modules are shifted further into the direction of series production.

The chosen sensor concept was sufficient for the function. But it has become clear, that for critical situations, like too narrow space in road constructions, more reliable sensor signals will be necessary. This could be met by using a stereoscopic camera to the front and more accurate radar sensors to the side.

The availability of the highly automated mode depends today on the results from the driver assessment and with this from the driver monitoring system. In the future it will be necessary to get the highly automated mode to a higher reliability, so the driver is not needed anymore. At least in some defined use cases.

Summing up, all objectives related with HAVEit challenge 5.1 “Automated Assistance in Road-works and Congestion” were achieved. The full spectrum of relevant use cases and scenarios can be covered by the application. A subset – in particular driving Highly Automated through a roadword area of a motorway – is demonstrated during the HAVEit final event.

7.2 Challenge 5.2: Automated Queue Assistance

Introduction

The Automated Queue Assistance application is fundamentally intended to support the driver in monotonous traffic situations like traffic jams or monotonous long distance driving from A to B where he or she can experience work-underload which can lead to a lack of focus and increa-sed accident risk.

The AquA demonstrator is a heavy truck application which will support the driver on motorways with different levels of automation in longitudinal and lateral control of the vehicle at speeds between 0 and 30 km/h. This section gives an overview of the demonstrator setup and describes the system functionality of the complete AQuA system after having installed all sensors and components as documented in deliverables D52.1 and initially tested them as documented in D52.2.

The configuration of the demonstrator vehicle is briefly described following the commonly used layer structure consisting of perception layer, command layer and execution layer with a special attention on driver interface. Within the above mentioned layers all relevant components are described in more detail than in previous deliverables, focussing on the functional performance and contribution of each component to the overall AQuA system functionality.

Finally, the AQuA system validation method and achieved results are described. The validation was done with reference to the previously defined use cases and scenarios as defined in D11.1. The selected use cases include activation, deactivation and different driver interaction with the system. In all use cases the system performs according to the specified behaviour. The use cases involve all the described system layers and thus give a good overview about the currently achieved overall system functionality of the AQuA system in the demonstrator vehicle.

7.2.1 Demonstrator configuration

Similar to the other demonstrator vehicles, the AQuA architecture follows the 4 layers of the generic HAVEit architecture. As all vehicle modifications are described in detail in HAVEit deliverables D52.1, D52.2 and D52.3, here just a brief summary is included.

Perception layer

A high level of automation requires a high level of perception around the vehicle. To meet these requirements, numerous sensors are used to monitor the complete surroundings of the vehicle and the driver. The prototype vehicle with the installed sensors is shown in Figure 228.

The laser scanners and the object camera are used to detect objects in front of the vehicle. The camera also provides information about the lane such as the lateral position of the vehicle in the lane and lane curvature. The adjacent lanes at the sides of the vehicle are monitored for objects by two short range radars, one on each side. Moreover, e-horizon is used to receive more information about the road such as road type. The V2V sensor transmits and receives vehicle state information such as velocity and acceleration from surrounding traffic. In the cabin, a driver monitoring camera is used to observe the state of the driver and provides information whether the driver is drowsy or distracted or not. The information gathered by the different sensors is combined in the sensor data fusion algorithm) developed by partner ICCS and described in section 5.1 above) to achieve an unambiguous view of the surrounding environment.

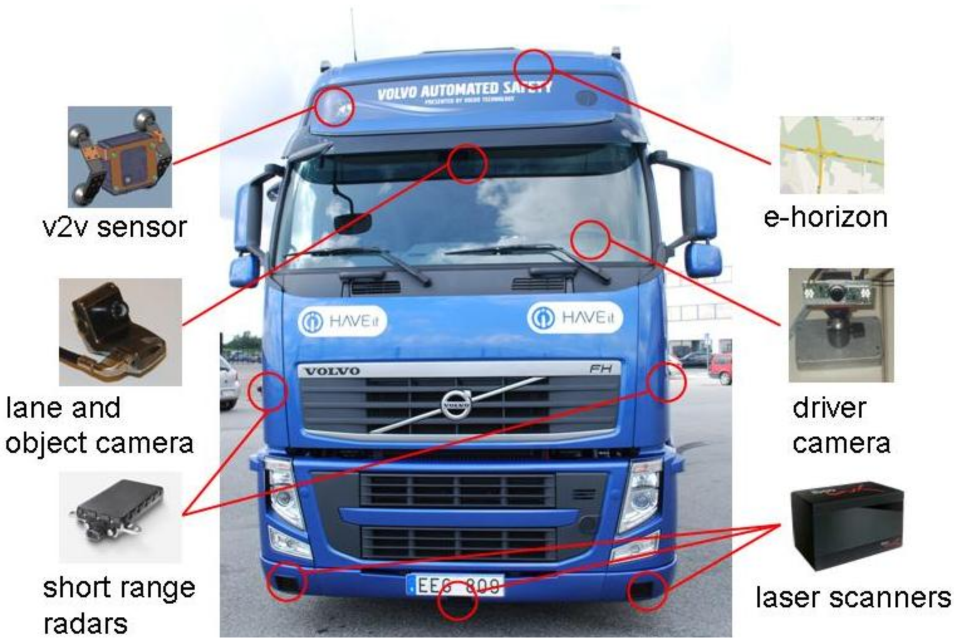


Figure 228: Prototype vehicle and sensor setup.

Command layer

Target selection unit (TSU)

To filter the list of objects received by the sensor data fusion (SDF) module, a target selection unit (TSU) was implemented. Since the target is used by the control algorithms it is highly important that any selected target from the TSU is correct. The implemented TSU uses velocity and estimated road curvature to create a funnel. Objects within the funnel are considered potential targets.

Figure 229 illustrates two different scenarios to show how estimated curvature and velocity affect the funnel. To the left in Figure 229 the host vehicle is standing still, the funnel end at approximately 50 meters. The centre of the funnel is placed on the predicted curvature. The right plot in Figure 229 illustrates the host vehicle travelling with 10 m/s and with a yaw rate at 0.018 which corresponds to a curve with radius of 550 meters. The funnel is formed around the predicted curvature. If several objects are within the funnel the algorithm uses lateral distance from centre of the predicted road and longitudinal distance from host vehicle to be able to select targets. Tracking signals such as for example track life time also affects the selection to some degree.

Naturally an object close to the centre of the predicted road and close to host vehicle is preferred over an object further away. The life time is implemented to filter out ghost object that appear for short time periods. The object that was selected as target during the last update can move small distances outside of the funnel without being dropped by the selector.

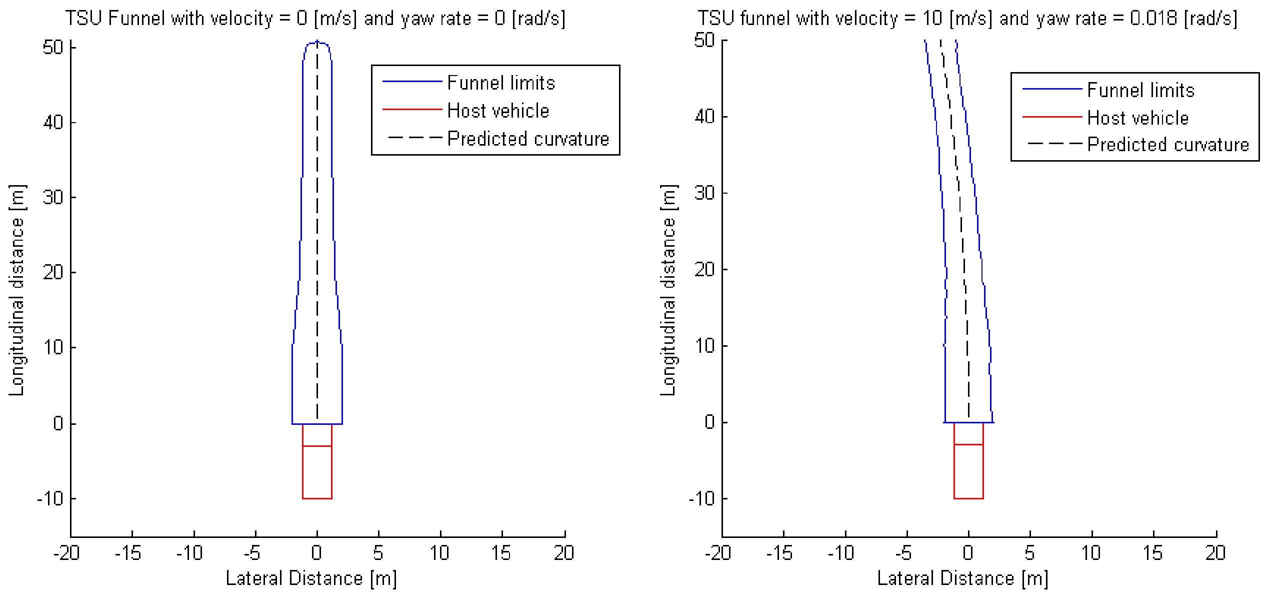


Figure 229: Funnel shape for two different scenarios

Figure 230 illustrates a series of snapshots over a period of 15 seconds. Both the target vehicle and the host vehicle speed is approximately 30 km/h. Before time 0 the host vehicle and the target vehicle are driving straight forward without interaction with other vehicles. Figure 231 illustrates the longitudinal distance to the selected target during the corresponding time period.

First in Figure 230 (a) the host vehicle follows the target in same lane. After about 5 seconds the target vehicle changes lane and target is lost, illustrated in Figure 230 (b). This can be confirmed from Figure 231 where value 255 corresponds to no target feed from TSU. Then the host vehicle changes to the same lane as the target which is shown in Figure 230 (c). The target reappears when the host vehicle and the object are in the same lane again. The TSU selects the correct target during the overtaking of the truck.

Figure 232 shows a simple test of the TSU in low velocity. There are two objects in front of the host vehicle. The SDF shows a third ghost target far out on the left side. As seen in the snapshot the TSU selects the object in the same lane as expected.

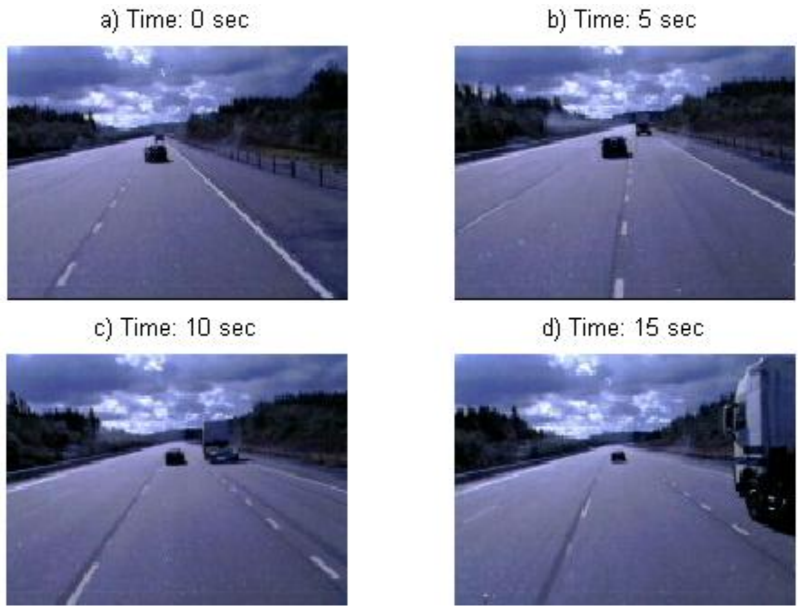


Figure 230: Snapshots from overtaking scenario

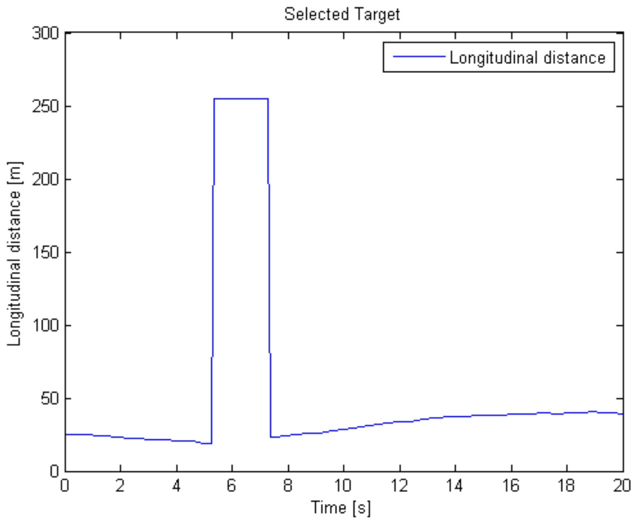


Figure 231: Longitudinal distance of selected target

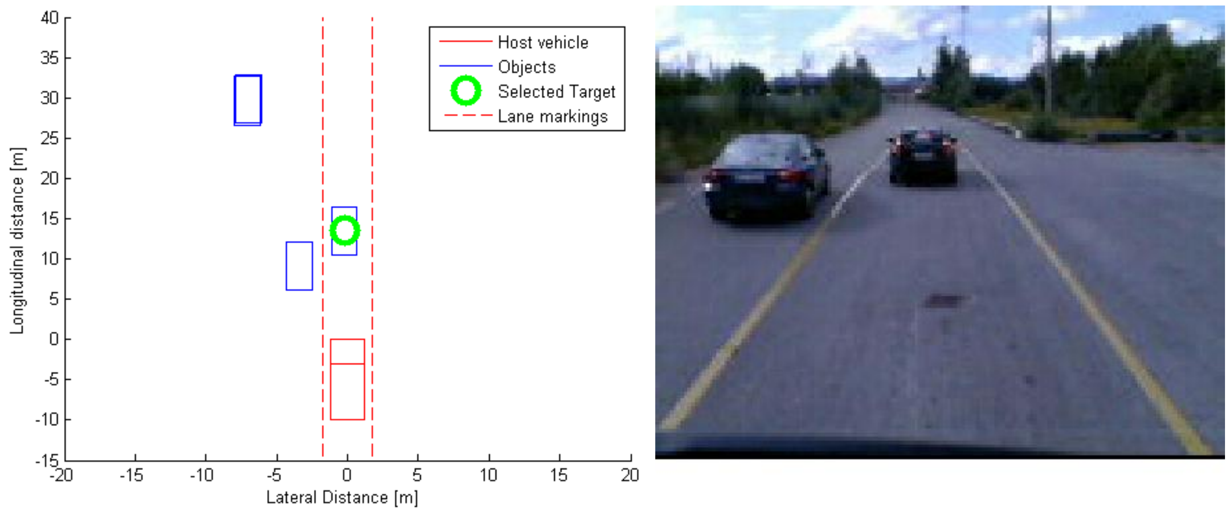


Figure 232: Simple queuing scenario with vehicles on different distance

Driver state assessment (DSA)

The DSA module assesses the driver state via direct and indirect measures. The goal of the test in the VTEC demonstration vehicle for AQuA was to validate whether the DSA component works correctly in the AQuA truck, whether all input signals are received properly, all internal calculations are done in the expected way and whether the DSA delivers meaningful output. The DSA has been described in D52.2.

The tests were performed in real traffic with a test driver following the instructions of a test leader sitting beside the test driver as a co-passenger. In addition to observing the output of the driver monitoring camera, several specific driving manoeuvres had to be performed by the test driver to simulate an “abnormal” driving behaviour that might occur if the driver gets drowsy. It had to be checked whether the DSA correctly identifies such manoeuvres and correctly calculates respective internal signals. These driving manoeuvres included executing very fast steering corrections, crossing the lanes with the wheels repeatedly within a short time interval, swerving in the lane for several minutes as getting drowsier, not steering for several seconds and driving between lanes for several seconds without using the indicator.

All instructions were followed only if the situation allowed it and if the traffic density was low enough. Only non-critical driving manoeuvres were performed on real roads. The test driver only concentrated on the driving task. All other tasks, such as recording of the test results were performed by the test leader. The driver was not distracted by the activities of the test leader.

Altogether, the tests revealed that the DSA was implemented successfully in the AQUA truck. The component provided by partners WIVW (indirect driver state assessment) and Continental (direct driver monitoring) works with full functionality. Nearly all input signals were received, internal calculations were calculated correctly and meaningful outputs were provided.

Mode selection and arbitration unit (MSU)

The Mode Selection and Arbitration Unit controls the activation and deactivation of the different parts of the AQuA function. It uses mainly the steering wheel buttons and driver actions for the control. The MSU is implemented as a state machine setting the correct state of the AQuA function with a decision unit giving the input. The decision unit uses the steering wheel buttons and sensor input to decide the wanted transitions for the AQuA system. The sensors used are e.g. vehicle sensors, environment sensors and the DSA.

Verification of the MSU function component has been carried out in simulations in PC-environment as well as with the complete system in the scenarios described in subsection 7.2.2. To activate AQuA the driver has to push the “AQuA” button on the steering wheel, AQuA function will only turn on if AQuA is available. The driver can disable the AQuA function by pushing the disable button.

Apart from pushing the disable button deactivation of the AQuA function can also be done by pressing the brake pedal or pressing the accelerator pedal and holding the steering wheel at the same time. When AQuA function has been on for a predefined time it will be deactivated as the driver presses the brake pedal.

When AQuA is on the driver can press the accelerator pedal for a certain time without deactivating the function. For deactivation the driver also needs to have the hands on the steering wheel. Similar as with the braking pedal the AQuA function can also be activated when the driver presses the accelerator pedal. If the driver presses the accelerator pedal without holding the hands on the steering wheel for a predefined time the driver will be informed to take control. If this is not done the AQuA function is deactivated.

When the AQuA function needs to be turned off, e.g. in case not all preconditions are fulfilled or in a case of system failure, the driver is requested to take over control. If the driver does not take control the system will enter minimum risk state and control the vehicle to a stop. If the driver takes over acceleration or steering of the vehicle the system will not override by entering minimum risk state, instead the full control will be handed over to the driver and a pop-up indicating "driver in control" is shown.

Command generation

a) Longitudinal controller

The purpose of the longitudinal controller of the AQuA function is to control the inter-vehicle distance between the ego vehicle and a target vehicle. The inter-vehicle distance at steady-state is derived from the ego vehicle speed by a driver set time gap. The target vehicle is the leading vehicle in the ego lane. When there is no target in front of the ego vehicle the longitudinal controller shall control the vehicle to a certain driver or system set speed. An overview of the longitudinal control is shown in Figure 233.

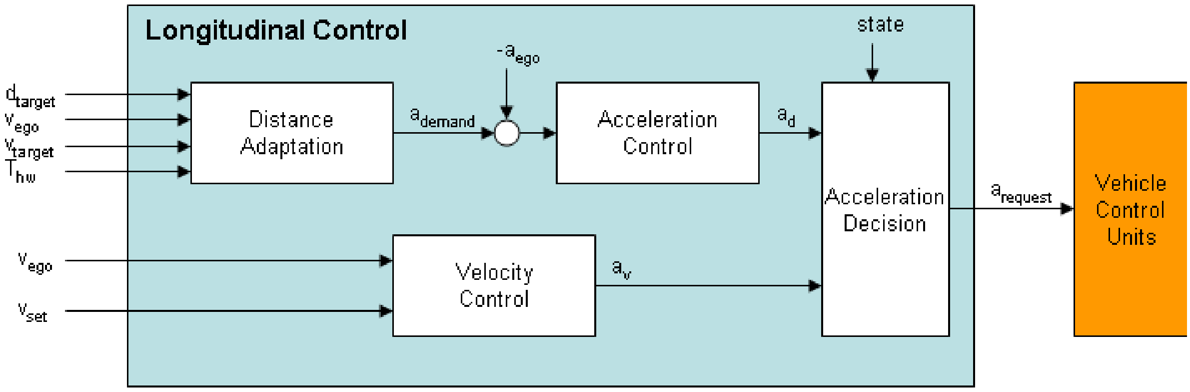


Figure 233: Overview of the longitudinal control in AQuA

The longitudinal controller consists of two parts, one to control the ego velocity when there is no relevant target vehicle and another to adapt the distance to the target vehicle. The velocity control maintains a desired set velocity of the ego vehicle, v_{set}, which is given by the driver or handled by the application in the automated driving mode. The velocity controller outputs a desired acceleration with respect to velocity, a_V.

The distance adaptation and the acceleration control adapt the distance between the ego vehicle and the target vehicle, d_{target}. The desired head-way time in the steady state conditions can be adjusted by the driver or by the application in the automated driving mode. The acceleration control outputs a desired acceleration with respect to the states of the ego and the target vehicle, a_d.

The decision is based on the state of the system and the desired accelerations provided by the different parts of the longitudinal control. The output is a requested acceleration, $a_{request}$, to the vehicle control units which handles the control of the engine and brakes.

To verify the functionality of the longitudinal controller the function was tested with a virtual target in front of the ego vehicle. The virtual target is controlled by the acceleration from a given initial distance and speed. In Figure 234 a virtual target is used to test a stop-and-go scenario for the longitudinal control. At time around 120 s the target brakes from 26.3 kph, with an acceleration of -2 m/s^2, leading to a brake of the ego vehicle. The ego vehicle manages to brake to stand-still behind the target vehicle before a collision occurs. The target vehicle then starts to drive away and the ego vehicle catches up. The speed of the target vehicle is altered through the scenario to show the dynamic behaviour of the longitudinal control of the AQuA function.

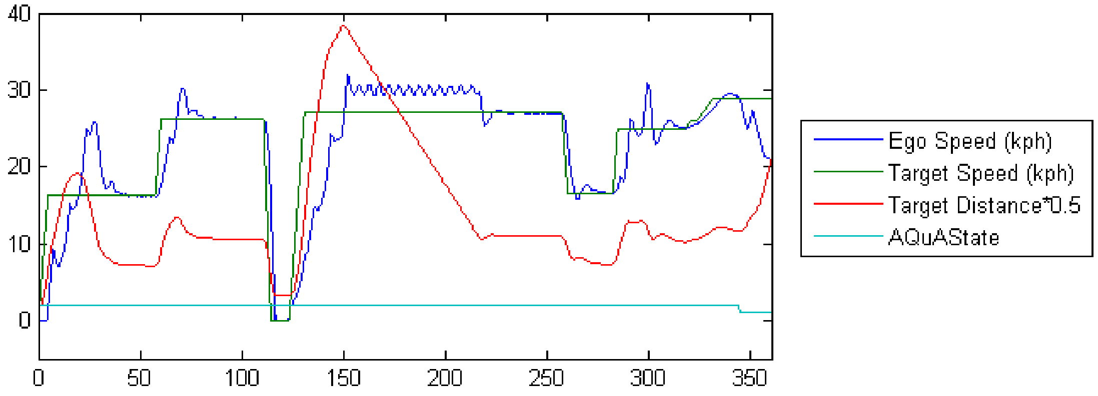


Figure 234: Longitudinal control dynamic behaviour with virtual target

Lateral controller

The lateral controller aims to keep the vehicle in the correct lateral position. It uses data from the SDF unit regarding distance to the lane markings, heading difference compared to lane and lane curvature. The input data is used to calculate the control commands to the steering actuator. An overview of the controller is shown in Figure 235. The controller utilizes the lane curvature, lane position error and heading error to calculate the desired steering angle. At the same time it takes into consideration limitations to avoid to rapid lateral movement.

The controller includes a vehicle and road model which is used for data filtering and prediction as well as for the control strategy. The filtering is needed to handle noise and spikes in the input, an example can be seen in Figure 236 (b) at time about 189 seconds. The prediction is used to compensate for delays within the system, primarily the delay originating from the steering actuator and also seen in plot (c) below.

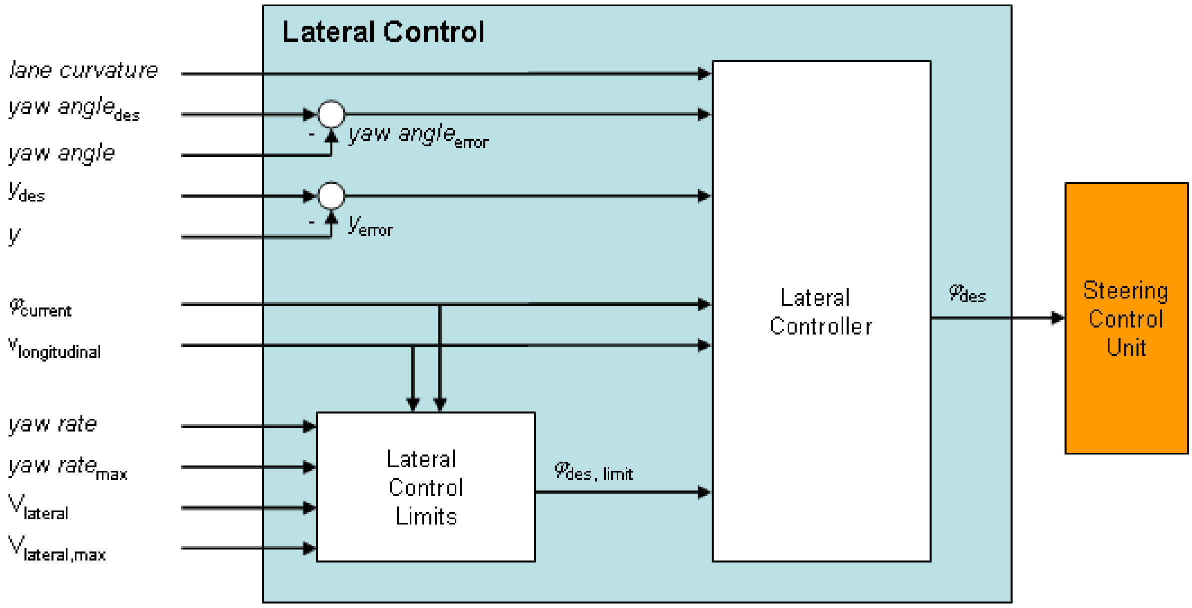


Figure 235: Overview of the lateral controller

The lateral controller moreover contains a path generating algorithm, in most cases the desired path coincides with the centre of the ego lane. However, exceptions can be made in some special cases for example in sharp curves or if there are close adjacent vehicles. The controller compares the current position and path with the desired values. Based on this control error the command to the steering actuator is calculated. The results on the lane keeping performance can be seen in Figure 236 (d). In the plot the vehicle is driving in a curve with curvature about -0.0017 m^{-1} which corresponds to a curve with radius of about 600 metres, a rather sharp curve on a motorway. It can be seen in the figure that the vehicle is clear within its lane at all times. The amplitude of the lane position is about 0.25 metres in the plot.

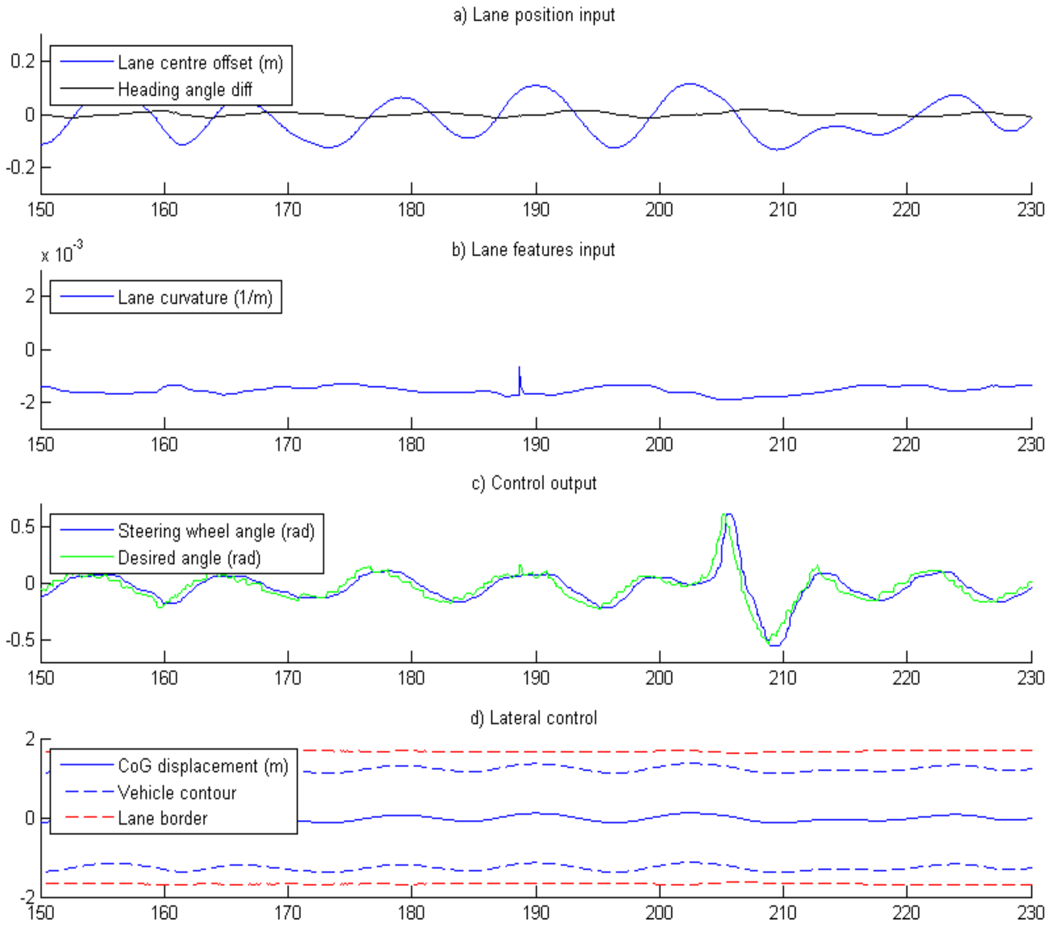


Figure 236: Lateral control behaviour

Human machine interface

Control of the Human Machine Interface (HMI) is implemented in the command layer. The HMI control triggers the pop-ups that should be shown to the driver and decides which one to show from a prioritisation of all available pop-ups. All timing of the pop-ups is situated and parameterized in the HMI control. The main input for the HMI control is the MSU output, but also signals from the longitudinal and lateral control is used, this is to trigger warnings from these function parts such as for example brake capacity warning.

The verification scenario is shown in Figure 237. AQuA button is pushed to activate the function but the system is not available and the pop-up ID 31, “AQuA not possible” is triggered. When AQuA is activated and any of the preconditions fail this is indicated by triggering pop-up ID 1 telling the driver to “Take over control”. The driver takes control and the prioritised pop-up ID 2, “Driver in control”, is triggered. The AQuA function is activated again but after some time the preconditions fail and pop-up ID 1 is triggered again. The driver does not respond and after a predefined time the system enters minimum risk state which is indicated to the driver with the prioritised pop-up ID 4.

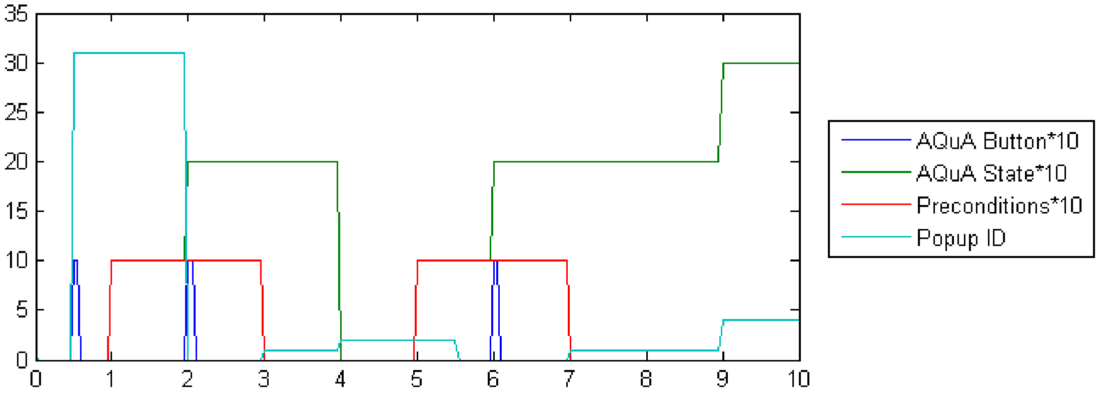


Figure 237: Results from the verification of the HMI control

Activation and deactivation of the AQuA system have been shown and also cases when the driver tries to activate the AQuA system when preconditions are not fulfilled. Moreover, entering the minimum risk state has been presented. For these different scenarios the HMI control works as desired.

To communicate with the driver an intuitive HMI including graphics and sounds is used (see detailed description of the AQuA HMI in D52.2. and D52.3). The display at all times shows the working mode of the system to the left: Manual, ACC+ or AQuA, see Figure 238. If the system is active, the symbol is lit up in blue, if it is available it is faded in light blue and when unavailable it is faded out. Urgent information to the driver such as take over requests are presented in pop ups presented in the right part of the display. The more critical information is further amplified with sound warnings.

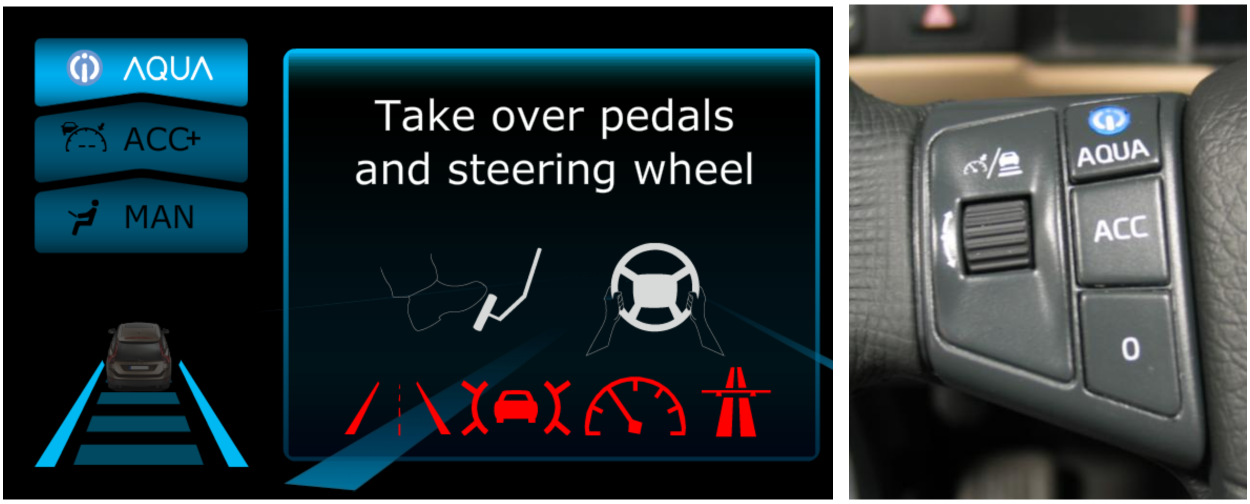


Figure 238: AQuA HMI, take over request.

Execution layer

The motion vector (longitudinal and lateral control command) is forwarded to the execution layer (see D52.2 for details). The control commands are sent to the different drivetrain controllers which directly control the engine, brake and steering systems of the vehicle: The longitudinal control command is implemented by applying engine torque or braking force. The lateral control command is executed by an appropriate steering torque.

7.2.2 System validation

In the following the AQuA behavior in dedicated use cases is presented. All use case tests were performed on the test track in controlled environment. The system is working as intended in all use cases. In each use case described below the actors are:

- Host vehicle (blue)
- Target vehicle (white)
- Driver
- Environment

Use case "AQuA activation"

The subject vehicle approaches the target vehicle. The driver activates the AQuA function. AQuA adjusts the longitudinal speed of the subject vehicle in a comfortable way to achieve and maintain the same speed as the target vehicle at the distance derived from the speed of the subject vehicle and desired time gap. The AQuA controls the subject vehicle laterally to stay in the own lane by actively steering the vehicle. Below two different alternatives of the use case are shown.

Preconditions (alternative 1)

- All preconditions fulfilled (target available, subject speed below maximum AQuA speed (30 km/h), driving on highway and lane markings available).

Triggers (alternative 1)

- The driver pushes the AQuA button to turn on the system

Reaction (alternative 1: all preconditions fulfilled)

- The subject vehicle is controlled to follow the target vehicle and to stay inside the lane

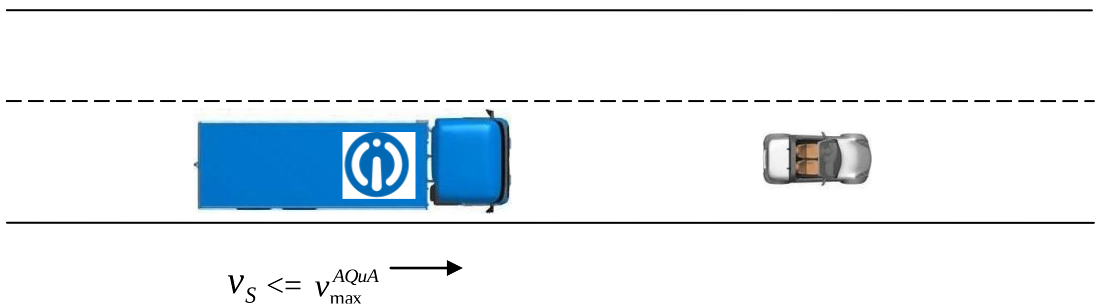


Figure 239: AQuA activation.

Preconditions (alternative 2)

- One or more preconditions not fulfilled.

Triggers (alternative 2)

- The driver pushes the AQuA button to turn on the system

Reaction (alternative 2: not all preconditions fulfilled)

- AQuA is not activated. The driver is informed that not all preconditions are fulfilled.

The results from the first use case are shown in the figures below. In Figure 240 the activation of the AQuA system is shown. In a) the AQuA state is “1-available” before the AQuA button is pressed. Since AQuA is available the AQuA state changes to “2-active” and the vehicle is controlled to follow a target and to stay in the lane. In b) the state is “0-not available” due to preconditions not fulfilled. When the button is pushed the driver is informed that the system is not available and AQuA is not activated. The message sent to the driver through the display is shown in Figure 241. The preconditions that are not fulfilled are marked in red. In the shown figure none of the preconditions are met.

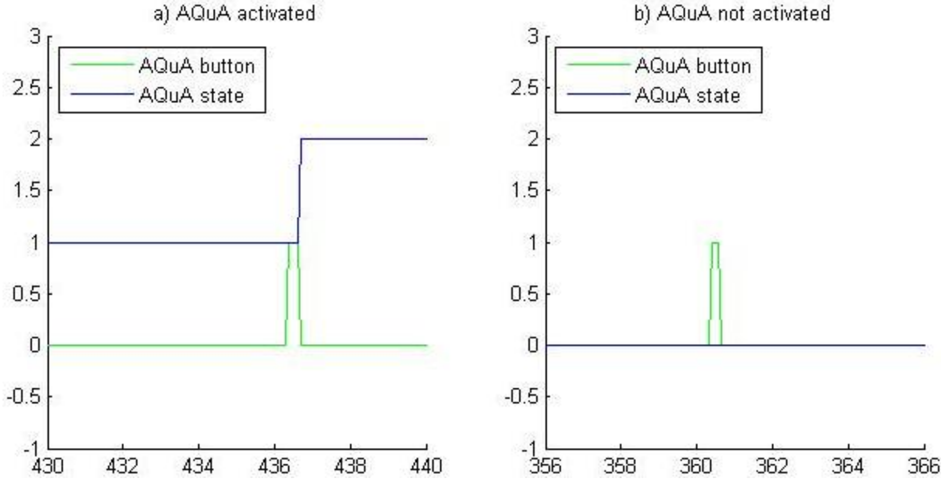


Figure 240: AQuA activation.

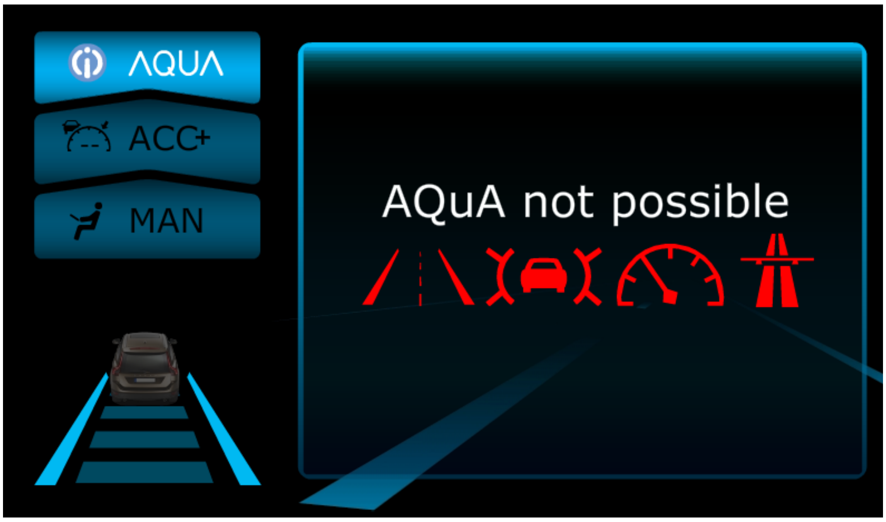


Figure 241: Driver display, AQuA not available.

Figure 242 shows results from driving with AQuA active. The data is from a driving in a curve on Hällered test track. (a) shows the lateral offset and heading deviation between lane and vehicle which is the result of the controller. (b) shows the corresponding lateral acceleration and yaw rate. In (c) the inter vehicle distance is shown together with ego and target speed. (d) shows the acceleration of the AQuA vehicle. The algorithm manages to keep the vehicle in the intended lane and controls the speed to keep a safe distance to the vehicle in front.

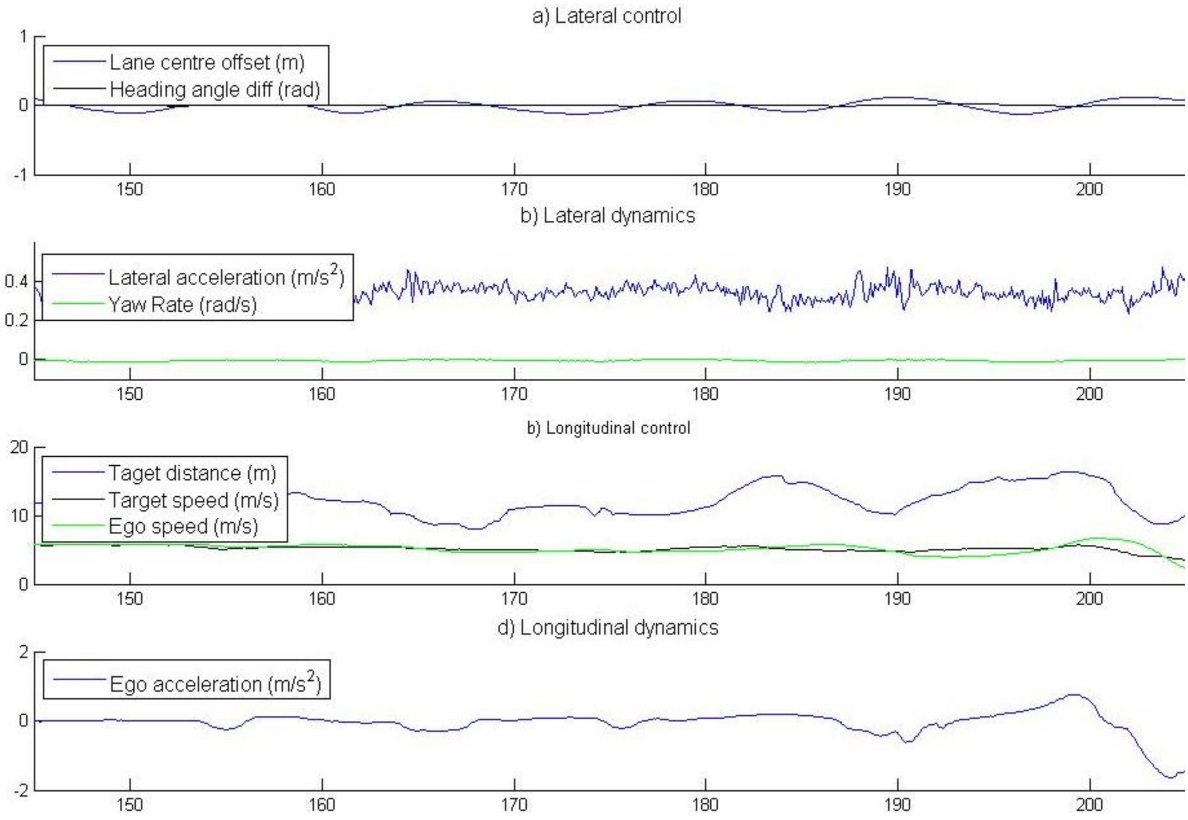


Figure 242: AQuA active.

<u>Use case “AQuA deactivation”</u>

The driver can actively turn AQuA off by pressing the brake pedal, steering against the system, accelerating or pushing the off button. When AQuA is active the surrounding conditions can change in a way so that the AQuA function becomes unavailable. When this occurs the control needs to be handed over to the driver. The system then informs the driver to take over the control of the vehicle. Changes that will make AQuA unavailable if occurring are loss of target, loss of lane markings, end of highway or driver out of the loop. Two different deactivation scenarios are described below, one which is driver initiated and one which is forced by the system due to preconditions not fulfilled.

Preconditions (alternative 1)

- AQuA is active

Triggers (alternative 1)

- The driver presses the AQuA button, brake pedal, accelerates above threshold or applies steering torque above threshold

Reaction (alternative 1)

- The AQuA system is deactivated and the control is handed over to the driver

Preconditions (alternative 2)

- AQuA is active

Triggers (alternative 2)

- One or more preconditions are not fulfilled

Reaction (alternative 2)

- The system informs the driver to take over control
- The AQuA system is deactivated and the control is handed over to the driver

Figure 243 (a) shows a driver initiated transition from AQuA to Manual. At t = 43 s the driver presses the brake pedal. When the defined threshold is reached the AQuA state goes from "2-active" to "1-available". In Figure 243 (b) a transition from AQuA to Manual initiated by preconditions is shown. At t = 532 s the AQuA preconditions signal goes to zero which indicates that not all preconditions are fulfilled. At that time a warning is issued to the driver urging him to take over control of the vehicle. After a few seconds the driver responds to the warning by pushing the accelerator pedal and the AQuA state goes from "2-Active" to "0-Not available". The reason for going to "0-Not available" instead of available is that the AQuA preconditions are not fulfilled as seen in (b).

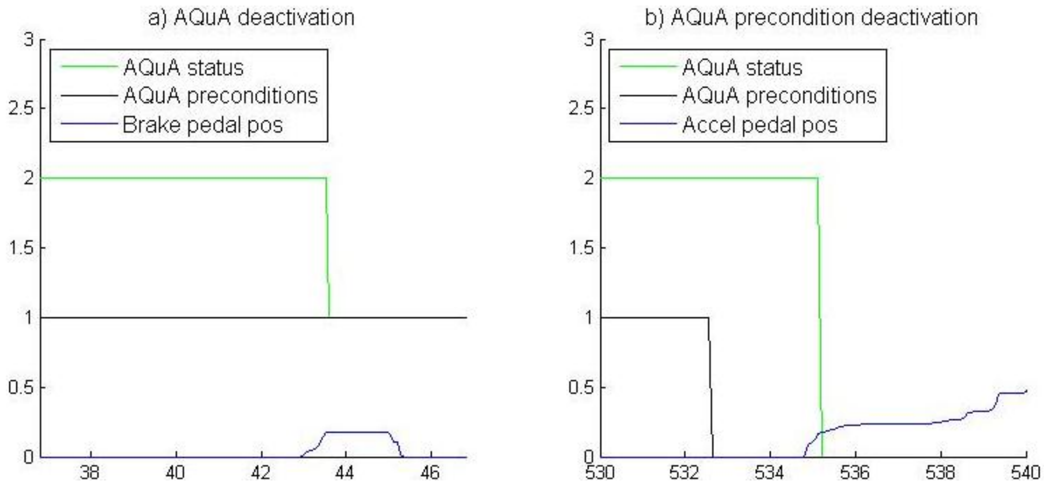
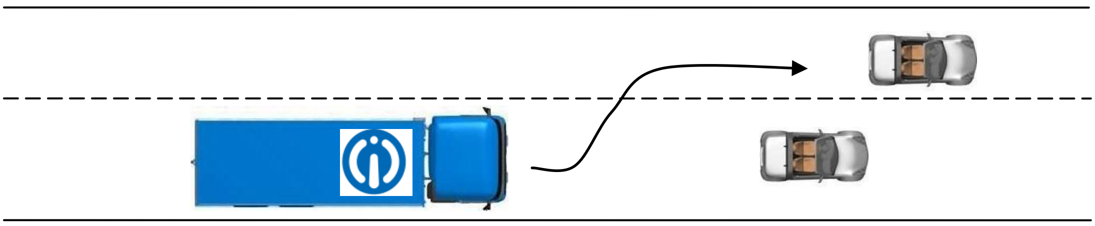


Figure 243: AQuA deactivation.

<u>*Use case "AQuA lane change"*</u>

When AQuA is activated and the driver initiates a lane change, the control will temporarily be handed over to the driver until the lane change is completed. When the subject vehicle is in the new lane and a new target vehicle is detected, the AQuA system automatically adapts to the longitudinal speed and regains the lateral control. The driver is informed that the AQuA is fully activated again. This scenario is outlined in the following figure. In case a vehicle is present in the adjacent lane parallel to the Subject vehicle, a lane change warning will inform the driver that a lane change should not be performed. The AQuA will continue to be active.



$v_S <= v_{\max}^{AQuA}$

Figure 244: AQuA lane change.

Preconditions (alternative 1)

- AQuA is active

Triggers (alternative 1)

- The driver indicates a lane change by turning on the left indicator when no vehicle in the adjacent lane

Reaction (alternative 1)

- The AQuA system gives the control to the driver
- When the driver has performed the lane change AQuA is reactivated

Preconditions (alternative 2)

- AQuA is active

Triggers (alternative 2)

- The driver indicates a lane change by turning on the left indicator when a vehicle is present in the adjacent lane

Reaction (alternative 2)

- A warning informs the driver that there is a vehicle in the adjacent lane

Figure 245 shows a lane change manoeuvre when there is no vehicle in the left lane and the lane change is aloud by the system.

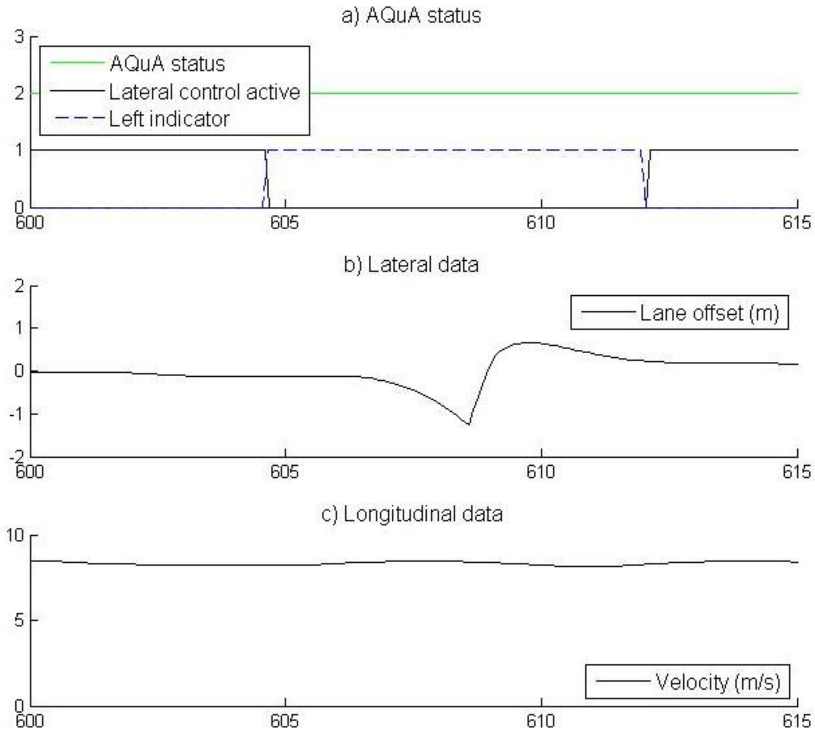


Figure 245: AQuA lane change.

(a) shows that the AQuA status is “2-active” at all times. At t=604.5s the driver indicates a lane change by turning on the left indicator. At that time the lateral control is paused as can be seen on the lateral control active signal. The driver performs the lane change and when he turns of the indicator the lateral control is reactivated and the steering handled autonomously. The lane offset during the manoeuvre is seen in (b) and shows that the offset to lane centre increases and then at t=608s starts to decrease when the system changes the chosen lane to the new lane. During the lane change the velocity is kept on a constant level and when the lane change is performed the speed is adapted to the new target vehicle.

Figure 246 shows when the driver indicates a lane change when a vehicle is present in the adjacent lane. It can be seen that when the indicator is turned on in this case the lateral control is not deactivated. Instead the warning in Figure 247 is shown to the driver and a possible accident can be avoided.

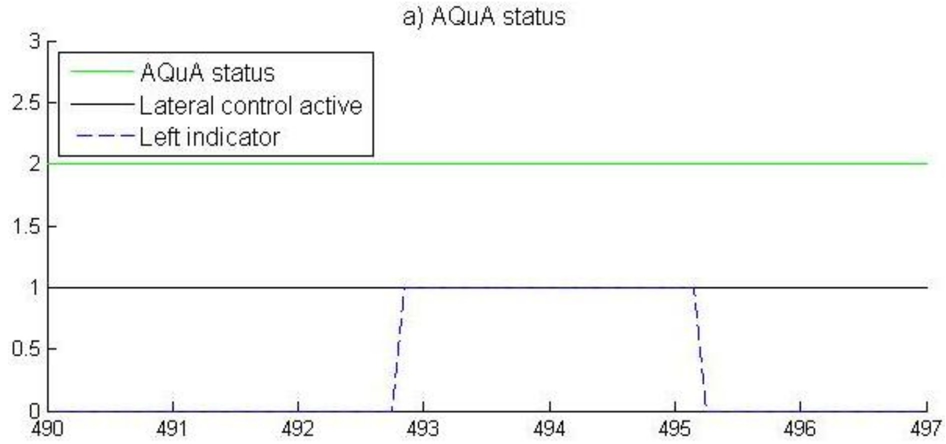


Figure 246: AQuA lane change when vehicle in adjacent lane.

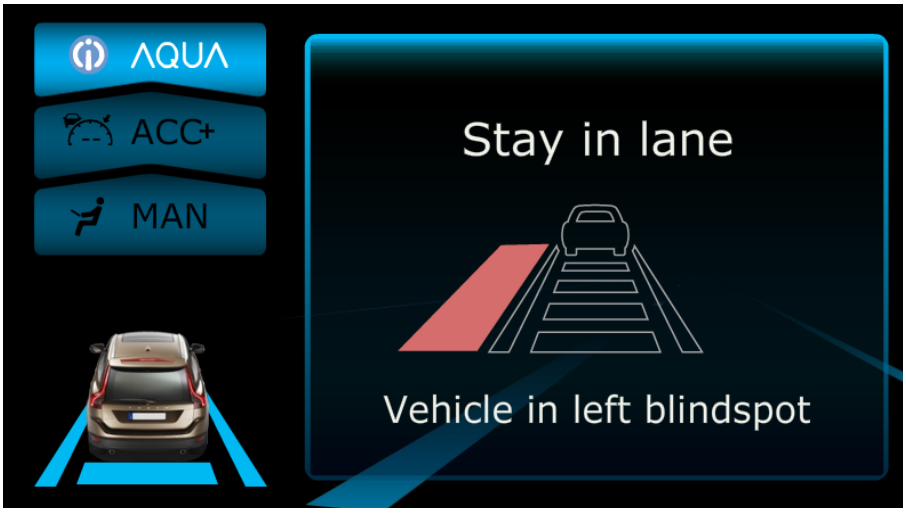


Figure 247: Warning, vehicle in blind spot.

Use case "AQuA minimum risk state"

AQuA will activate the minimum risk state manoeuvre if the driver is not in the loop or does not respond properly to the AQuA requests. In the minimum risk state, the AQuA system will slow down and stop the subject vehicle in a safe, controlled and smooth way.

Preconditions

- AQuA is active

Triggers

- The preconditions is no longer fulfilled, for example the target is lost
- AQuA informs the driver to take over the control of the vehicle
- The driver does not respond to the AQuA request

Reaction

- The AQuA system keeps the control of the longitudinal and lateral support and gives a significant warning to the driver to take over the control
- When the driver does not react to the AQuA system requests and warnings the system will go into minimum risk state and bring the vehicle to a complete stop
- The vehicle is kept stationary until the driver takes over control

In Figure 248 the results from tests of the minimum risk state is showed. In (a) AQuA state is “2-active”, (b) shows that the target disappears at time t = 394 s. At this time a message is sent to the driver to take over control. After about six seconds (when the driver has not responded to the warning), the system enters “3-Minimum Risk State” (MRS). The vehicle is brought down to a full stop, when the vehicle is stationary the driver presses the acceleration pedal to regain control of the vehicle and AQuA state changes to “0-not available”. The information showed to the driver in MRS is shown in Figure 249.

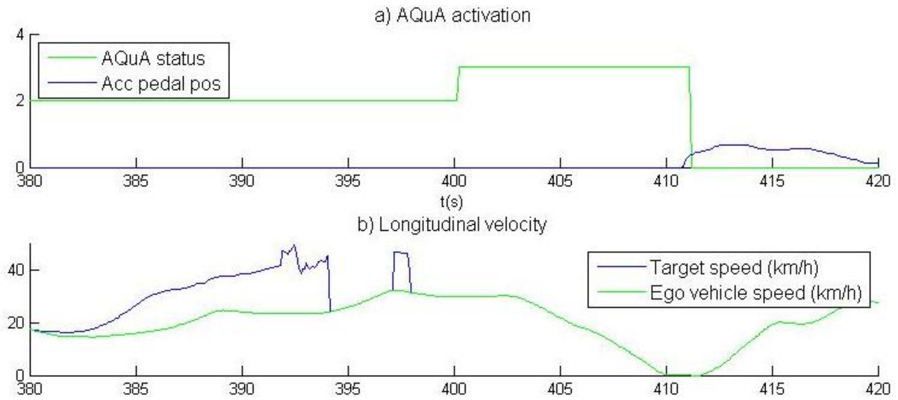


Figure 248: AQuA minimum risk state.

Figure 249: AQuA HMI minimum risk state.

7.2.3 Summary and conclusion

The highly automated vehicle application AQuA (Automated Queue Assistance) has been successfully integrated on the Volvo truck. The fully integrated system works as intended. In all scenarios derived from the AQuA use cases the algorithm reacted to the triggers in a satisfying way.

The successful testing proves that the technical concept and the designed architecture are valid. All units are implemented according to the HAVEit generic standard with just minor AQuA specific adaptations. The communication and interaction between all the components work both inside and between the different ECU's.

Having successfully passed the validation tests in relevant use cases, the objectives related with the HAVEit challenge 5.2 have been achieved.

7.3 Challenge 5.3: Temporary Auto-Pilot

The Temporary Auto Pilot (TAP) is fundamentally intended to support the driver in monotonous traffic situations like traffic jams or monotonous long distance driving from A to B where the driver can experience work underload which can lead to a lack of focus and increased accident risk. The TAP is a passenger car application which supports the driver on motorways and motorway similar roads with different levels of automation in longitudinal and lateral control of the vehicle at speeds between 0 and 130 km/h. This guarantees that the driver gets the best possible support available and contributes to traffic safety.

This section summarizes the overall system functionality of the integrated TAP system after previously installing all sensors and components (see deliverable D53.1) and basically testing them separately as documented in D53.2. Focus is laid on the validation of the TAP system in relevant use cases defined in D11.1. First system validation tests are described in detail in D53.3.

7.3.1 Demonstrator configuration

The TAP system has the following functionalities:

- Highly-Automated: hands-off driving, automated longitudinal and lateral control (Pilot)
- Semi-Automated: hands-on driving, automated longitudinal control (ACC)
- Assisted driving: hands-on driving, assisted lateral control like Lane Keeping Systems
- Minimum Risk Maneuver: emergency braking if the driver does not respond to Take-Over-Request (TOR)

The Pilot function can only be maintained under the given preconditions as follows:

a) subject vehicle is driving on a motorway with the maximum velocity of 130km/h, in the correct direction and between the lane markings

b) driver is not drowsy, is not sleeping, passed out or dead, is sitting on his seat, has his seat belt fastened, has not opened any door of the vehicle and does not press the brake pedal

c) lane markings are detected by the environment sensors

d) motorway (in the next few kilometers): motorway does not terminate, lane does not terminate, no construction sites and no exit proposed by the navigation system

e) no system failure

The TAP demonstrator vehicle (Figure 250) is a Volkswagen Passat 2.0 TDI passenger car with series environmental sensors, which are used for performing the following functions:

- ACC (77GHz Long Range Radar Sensor),
- Lane Keeping System (Monochrom Camera) and
- Park-Assist (Ultrasonic sensors).

For HAVEit purposes, the vehicle was modified according to the generic HAVEit architecture.

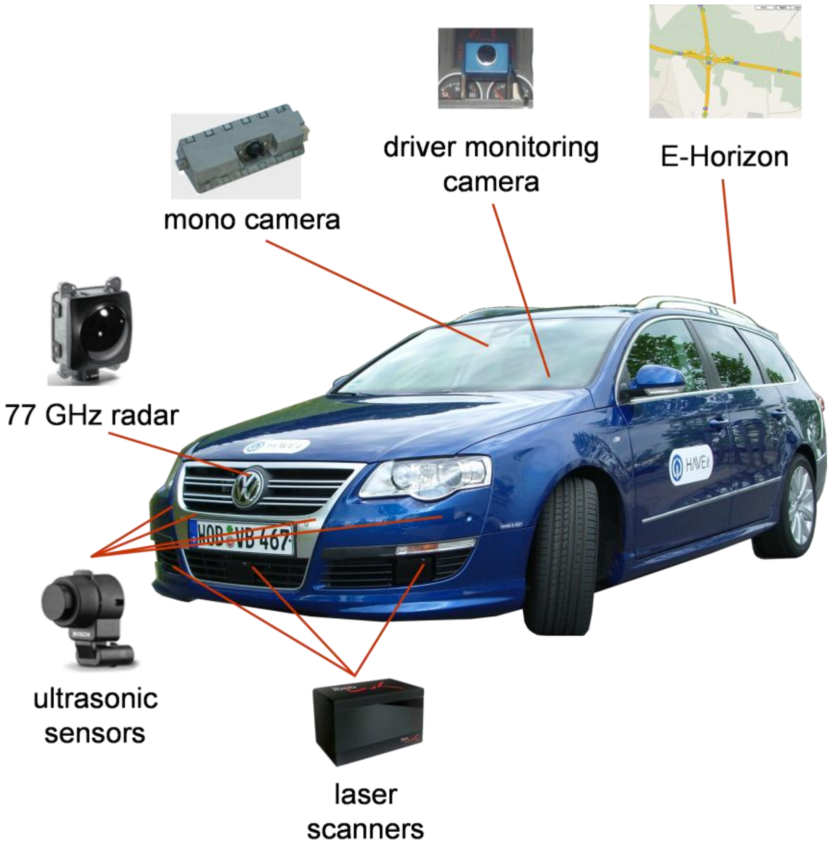


Figure 250: TAP demonstrator

Perception layer

The environmental perception sensors are extended compared to the serial vehicle:

Laser scanners

Three laser scanners from SICK were mounted in the front bumper (left, middle and right laser scanner). In the first phase of our software development we used all three laser sensors for the object detection and the free area detection at the environmental perception. In the final version, the left and right sensors are optional (as the TAP relevant scanarios can be covered with only one laser scanner as well).

An example of the laser scanner performance can be shown in a driving situation, where the host vehicle is stopped in the middle lane of a three-lane country road in Hällered test track. A preceding vehicle is coming in the same ego lane at the velocity of 70km/h. This moving object is early detected at the distance of about 130m and well tracked. As depicted in Figure 251 the plot for distance and velocity value over time is smooth, and accurately represents the object dynamical behaviour. The lateral velocity just after the first detection is a little high due to the fact that the object geometry and the lateral distance can't be well estimated in the far area. But it doesn't cause any bad effect on the whole performance of the TAP demonstrator.

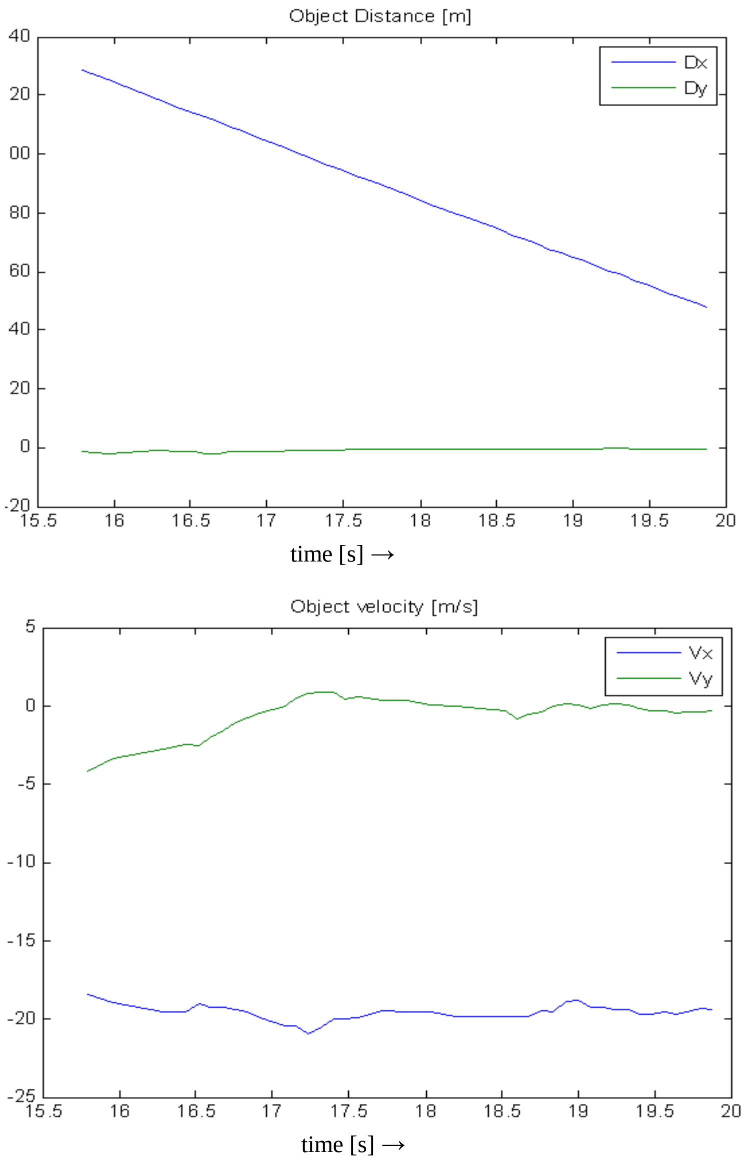


Figure 251: Object detection by SICK laser scanner

The following driving situation is selected for testing the performance of the free area detection with laser scanners: the ego vehicle is following a car on the most right lane of a two-lane motorway bounded by guard rails on the left and by green grass / trees on the right (see Figure 252). This figure illustrates the ego trajectory as a red line and the current boundary of the free area as a blue polygon. The left guard rail is robustly detected as a boundary in the near area until 30m (green marked box).

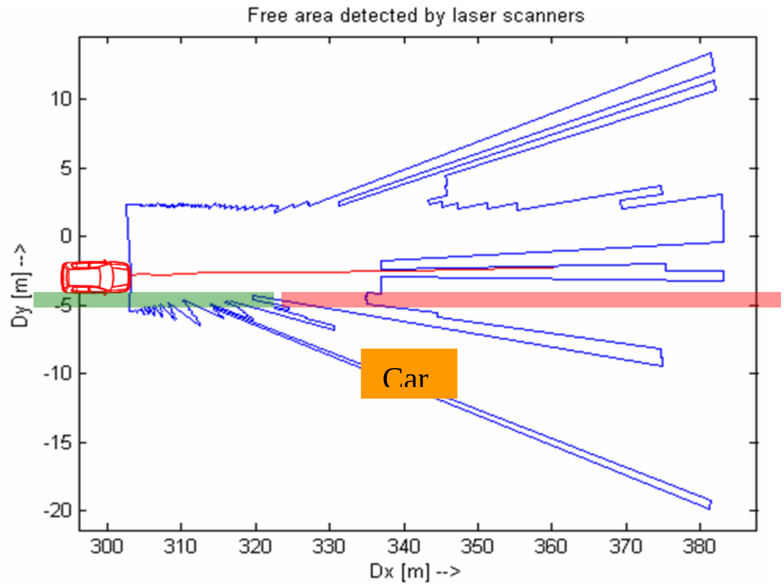


Figure 252: Free area detection by laser scanners

e-Horizon module for digital maps

The main idea for using eHorizon is to improve the whole performance of the TAP system by providing additional map data like lane curvature, lane curvature change in the lane segments, number of lanes, and speed limits. Those data are used to make the lane information available in the far area, which will improve the TAP functionality at least in two aspects:

- For the longitudinal control by a better lane assignment for objects: the relevant object will not be released on the sharp curve.
- For the lateral control by a better path planning and remaining the comfort lateral acceleration.

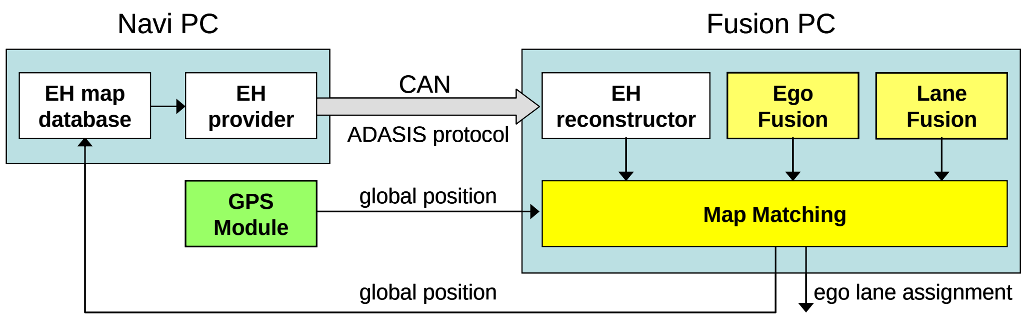


Figure 253: eHorizon and Map Matching

The eHorizon is implemented in NaviPC, which receives the current global position of the ego vehicle at the input and delivers map data over CAN into FusionPC (see D53.2 and Figure 253). But the final effect of the eHorizon can only be shown by working together with Map Matching, Ego Fusion and Lane Data Fusion, whose performance will be analysed more in detail below.

Sensors for driver state assessment and driver monitoring

First a comera for direct driver monitoring and drowsiness and distraction detection was integrated into the central instrument cluster (see description of the DSA module developed by partners WIVW and Continental in section 5.2 of this report).

Second, Volkswagen developed a hands-on detection module based on a capacitive sensor mounted in the steering wheel, the corresponding ECU and CAN module. This module together with modified steering wheels was provided by VW also to other partners, e.g. Continental, VTEC and DLR (for use in the JSD, AMD, ARC and AQuA demonstrators).

The hands-on detection developed by VW uses a prototype capacitive sensor and transmits raw data over CAN communication. In the first hardware-based implementation, the raw data are directly compared to the fixed threshold values (each one for a hands-on level by tuning corresponding potentiometers in the hardware) and the final results (hands-on, hands-close, hands-far and hands-off) are generated and delivered over CAN. The hardware-based implementation had difficulties with its calibration and its adaptation to the changing environment (changing pressure, humidity, different driver behaviours etc.). For example, in case of high humidity, the hands-close action was often detected as hands-on.

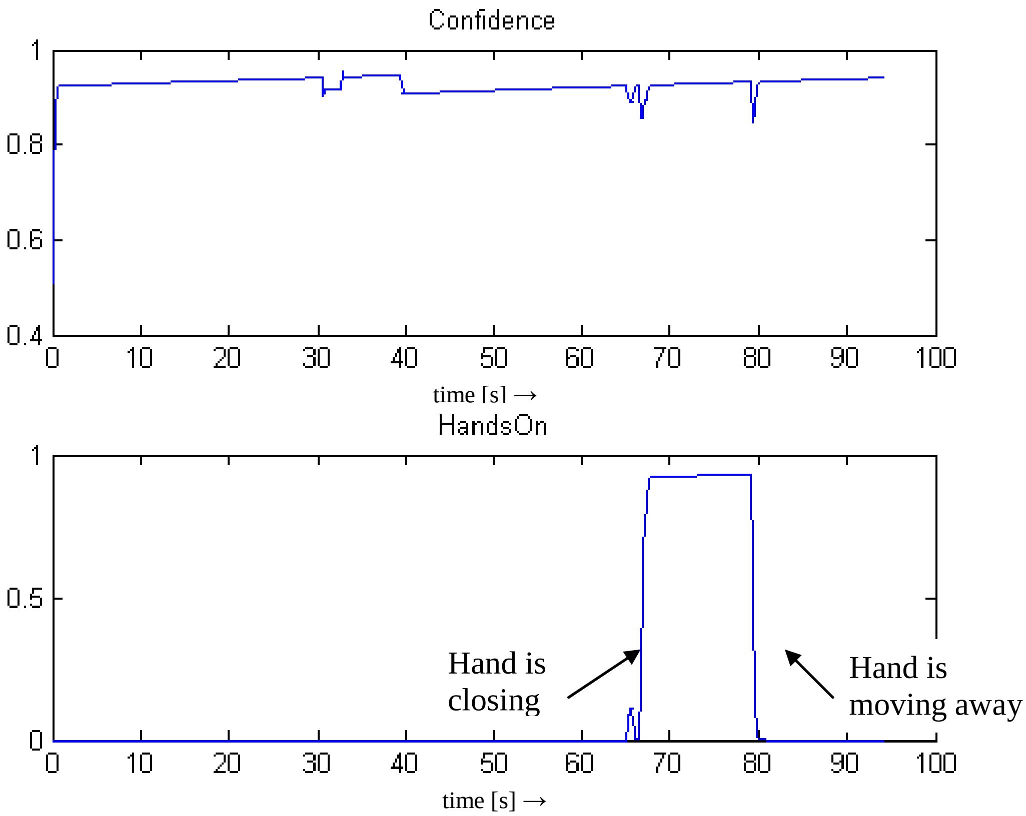


Figure 254: HandsOn Detection and its confidence

To avoid those problems, a second so-called software-based implementation was developed, where the raw data are further sent to and processed in the Sensor Data Fusion, and the threshold values are automatically calculated based on other additional available information. The system now runs very stable. In difference to the previous version, the new one delivers the hands-on signal and its confidence. Figure 254 depicts the belief on the hands-on process and its confidence over time. The belief for “hands-on” increases significantly from zero to 0.97 when the hand is moving closer to the steering wheel. It increases slightly when the hand stays on the wheel. It slows down dramatically, when the hand is releasing the steering wheel.

Sensor data fusion

Before explaining the TAP key modifications in the command layer, a brief overview is given on the Sensor Data Fusion concept (see fusion architecture in Figure 255) employed for the TAP, as the outcome of the data fusion has direct impact on the longitudinal and lateral control behaviours in particular and on the overall TAP functionalities in general. The Sensor Data Fusion is intended to receive data from all the sensors, to perform processing on the data, and finally to send the processed data to the command layer.

According to the TAP functionalities, the Sensor Data Fusion module must robustly make available the following information to the applications:

- a list of objects that are in the driving environment around the TAP vehicle in order to select a relevant target for longitudinal control (e.g. to keep a safe distance to the vehicle in front of the TAP vehicle or to avoid TAP vehicle to overtaking on the right side)
- information about the TAP vehicle's dynamics within the current lane for lateral control (to keep the vehicle in the middle of lane), and
- the free drivable area around the TAP demonstrator vehicle, which will be used e.g. for supervising automatic GO manoeuvres, especially in a traffic jam situation.

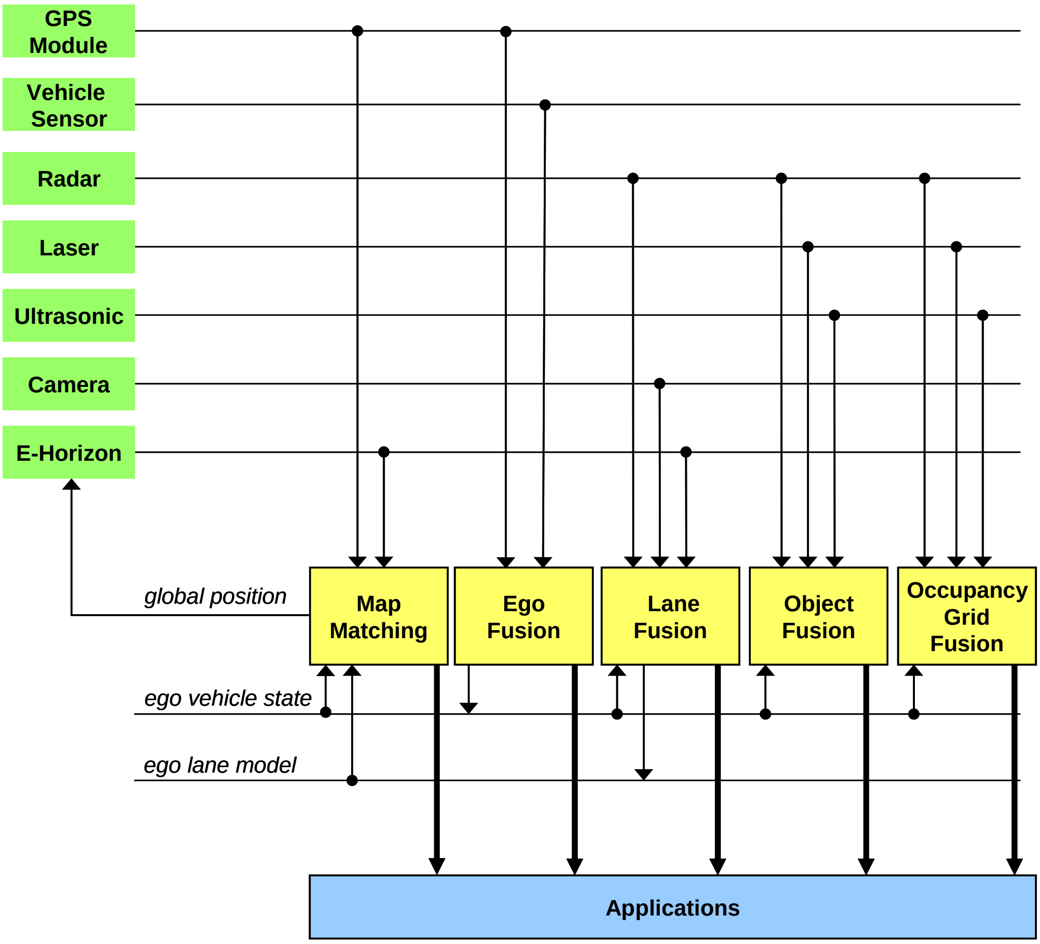


Figure 255: Sensor Data Fusion architecture

The SDF framework is implemented in the Automotive Data and Time-Triggered Framework (ADTF) and has therefore a good modular architecture. It all runs on an EKF-PC. The fusion architecture is shown in Figure 255. For a more detailed description of the SDF system and its algorithms, refer to D53.2. The SDF system has been successfully implemented and thoroughly tested in Online and Offline:

1. Offline testing: Sensor data has been logged from the TAP demonstrator and replayed offline to feed the SDF with data in a development framework. This has enabled a detailed analysis, debugging of many aspects and developing new algorithms.
2. Online (Real-time) testing in the TAP demonstrator: This involves driving in a variety of scenarios and observing the output in a visualisation framework. This can be a quick way to evaluate overall SDF system performance in real time.

<u>Command layer</u>

The command layer is structured according to the HAVEit architecture.

Target selection unit (TSU)

The Target Selection Unit is implemented to select the relevant (control) target object from the objects received from the Sensor Data Fusion. The relevant target object is a target, which is in the ego lane or in the ego driving tube if the lane information is available or not respectively.

For driving comfort purposes, the relevant target object should be recognized as early as possible:

- Even if an object at the current time point still isn't in the ego lane, but it is going to cut in the ego lane, just in the front of ego-vehicle, it should be identified as the relevant object and the ego vehicle will slow down its velocity.
- For the cut-out situation, even the relevant object is also in the ego lane, but plans to change the lane, it should be not longer considered as relevant. As effect, the ego vehicle should plan to accelerate on-time.
- If the ego vehicle intends to change the lane, the current relevant object should be released immediately.

For making a correct decision if an object is relevant or not, the lane information, the lane assignment for ego and objects and the manoeuvre prediction are needed. This information is provided by the Sensor Data Fusion and available in the TAP system.

For illustration, Figure 256 depicts the target selection for a driving situation from a straight road through a sharp left curve with curve radius about 200m, through a straight road and to a left curve with curve radius about 300m (see the top plot for lane curvature).

The ego vehicle is driving on the straight road and detects a little slower moving object (with Object-ID 9 depicted in the Object-ID plot) entering the 1st curve at the time t=106sec and at the distance of 60m (see the 2nd plot from the bottom). The ego vehicle slows down its velocity to approach the relevant object and after that it follows the relevant object on the curve. At time t=140sec, just before the curve exit, the relevant object changes the lane to the left lane (identified by a positive lateral offset of the object to the lane middle about 2m at the bottom plot), and the ego vehicle drives in the free mode on the straight-forward road (no relevant object). At the time t=195sec, it detects a very slowly moving object which is cutting in its lane from the right lane in a traffic jam situation, and considers it as the relevant object while following it until the still stand.

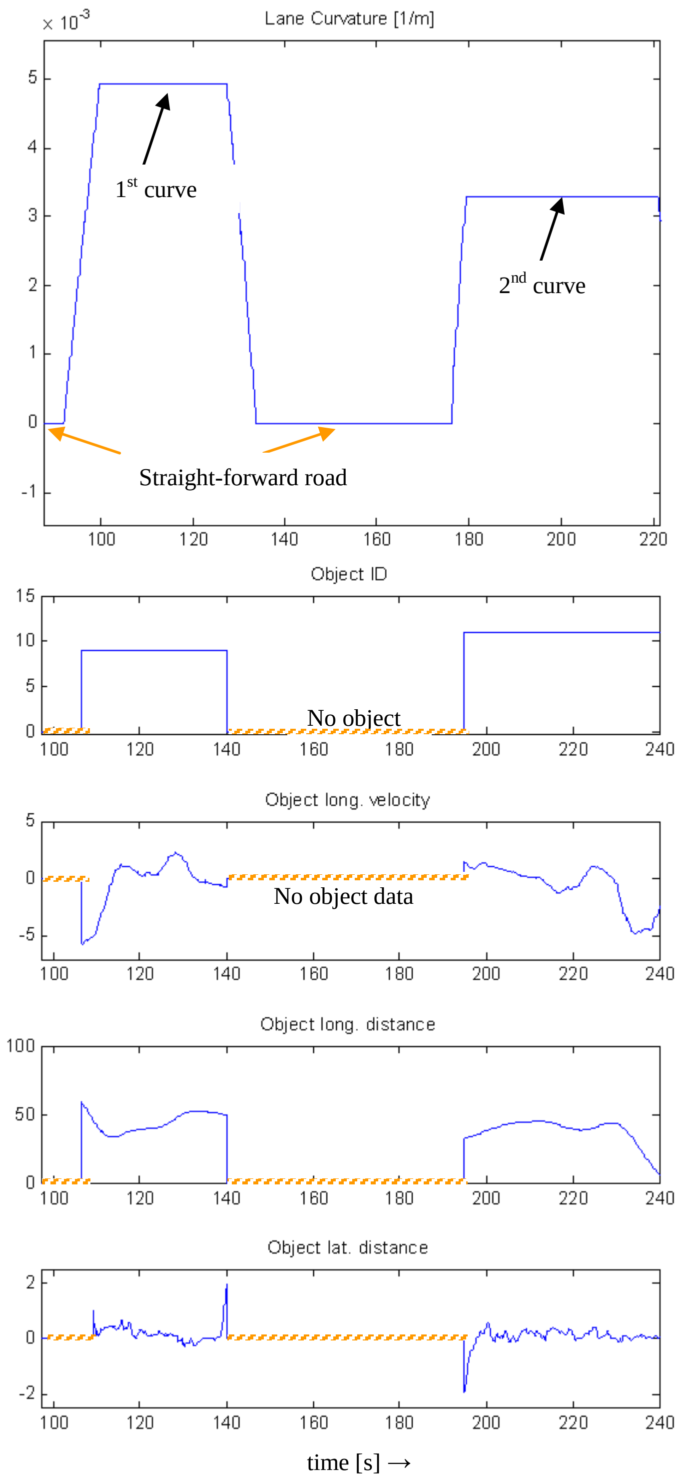


Figure 256: Target selection in the cut-in and cut-out situation

Mode Selection and Arbitration Unit (MSU)

The TAP demonstrator has the following automation levels:

- Driver only / Driver-Assisted (Assistance Function)
- Semi-Automated (ACC - Adaptive Cruise Control)
- Highly Automated (Pilot Function)
- Minimum Risk Manoeuvre (Safety Function)

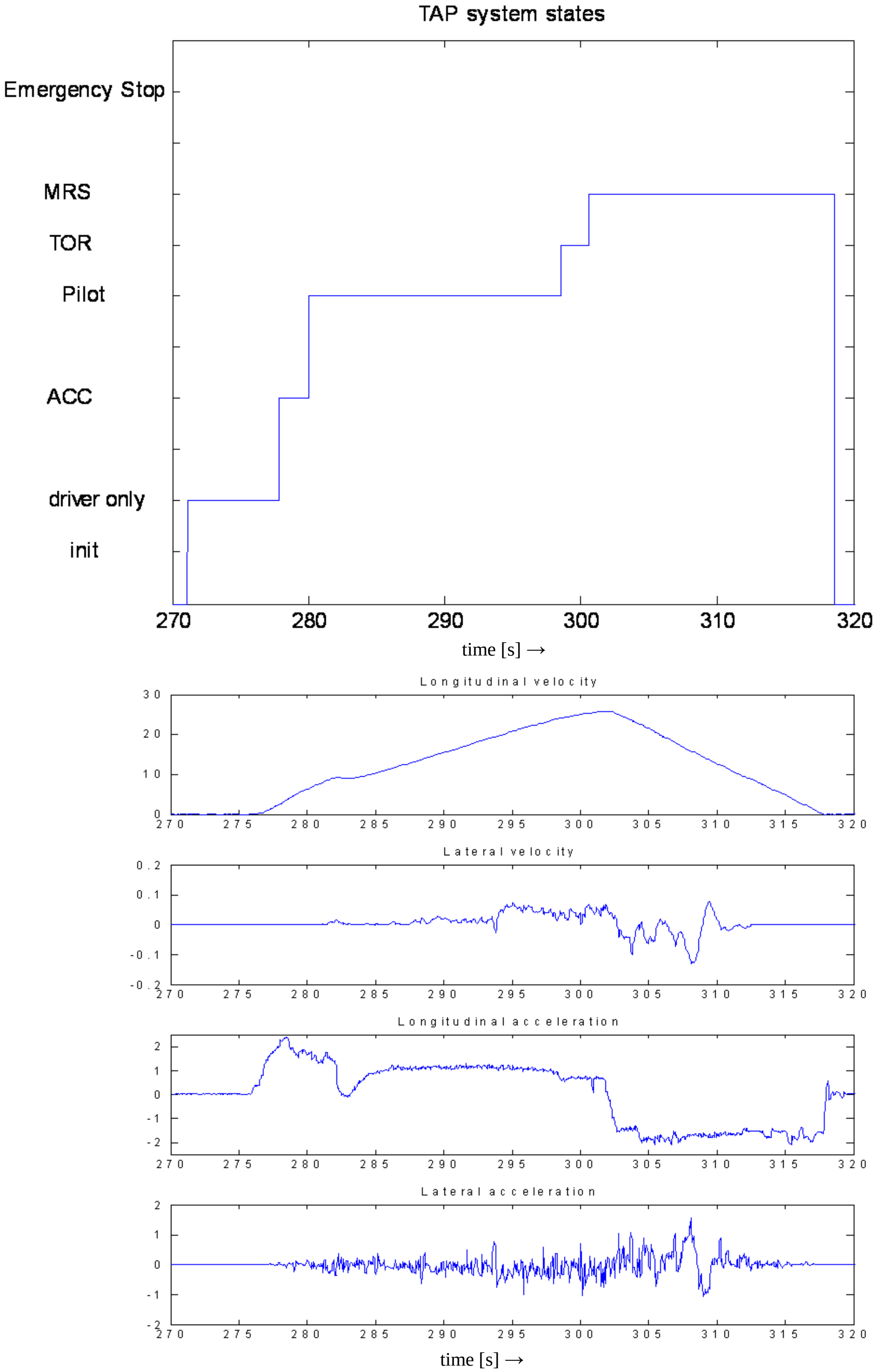


Figure 257: Mode Selection Unit

For implementing those functionalities, a state machine is used to organize the mode selection and transitions between different states like *Off, Initialize, Driver only, ACC, Pilot, Take-Over-Request, Minimum Risk State (MRS), Emergency Stop.*

Figure 257 shows a test for the Mode Selection and Arbitration Unit in a driving situation, where the ego vehicle drives from standstill to a given desired velocity (about 100km/h) through different modes *Initialize, Driver only, ACC,* and reaches *Pilot* state after 10 sec. At the time instant t=298sec. the system detects that driver is “un-responding” (here, the driver’s face is covered by a book so that DMS does not detect him) and generates a TOR (take-over request) message to warn the driver to take over. Because the driver does not respond to the TOR message after a given time interval (e.g. here 2sec.) the TAP demonstrator starts Minimum Risk Manoeuvre to stop the ego vehicle on the current lane. Here, the acceleration and deceleration of the ego vehicle seems to be quite comfortable and have acceptable values in the comfortable interval [-3.0m/s^2, +3.0m/s^2] (see the plot Longitudinal acceleration).

Longitudinal controller

The longitudinal controller (see Figure 258 and D53.2) is aimed to maintain a desired velocity in the free mode and a desired time gap to a target vehicle in a suitable way. The time gap is defined as the time the ego vehicle needs to reach a target vehicle by the current velocity. In other words, the time gap is a ratio between the current distance to a target object and the current ego velocity. In the TAP demonstrator, the desired time gap can be set and adjusted by the driver using the ACC lever on the left side of the steering wheel. Figure 259 depicts the time gap over time (in the top plot) and the ego longitudinal velocity over time (in the bottom plot) for the Follow-To-Stop situation.

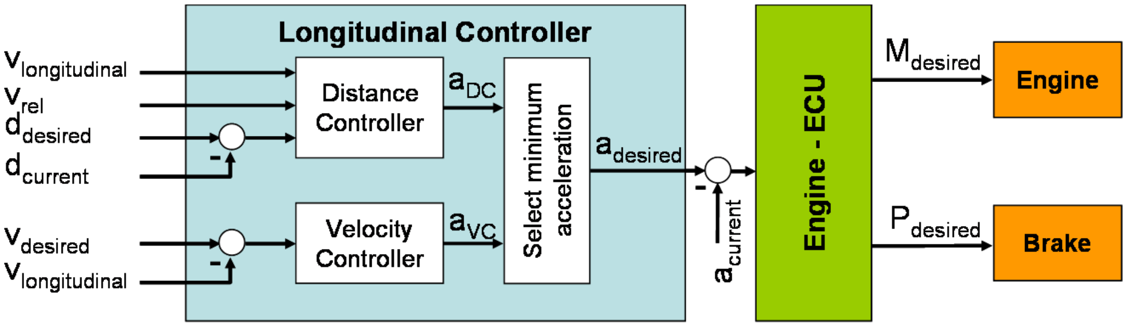


Figure 258: Longitudinal Controller

Lateral controller

The Lateral controller aims to keep the vehicle in the center of the current lane. An overview of the controller is shown in Figure 260 (see D53.2). The Lateral Controller consists of Lateral Offset Controller and Steering Angle Controller. The first one utilizes the lane curvature, lateral offset error and heading error to calculate the desired steering angle. The angle error between the desired and current steering angle is sent further to the Steering Angle Controller to calculate the desired steering torque. At the same time it takes in consideration limitations (e.g. maximum lateral acceleration, maximum steering torque) to avoid too rapid lateral steering movement.

The functionality of the Lateral Controller can be improved if the TAP is not only kept in the middle of a lane, but at the same time the best driving comfort is maintained in the curve. This requirement can be fulfilled by implementing an integrated longitudinal and lateral control, e.g. by reduction of the longitudinal velocity in the curve to the optimum value. Normally, the maximum allowed velocity depends on the curve radius: the higher the curvature of a curve is, the slower the TAP demonstrator should drive.

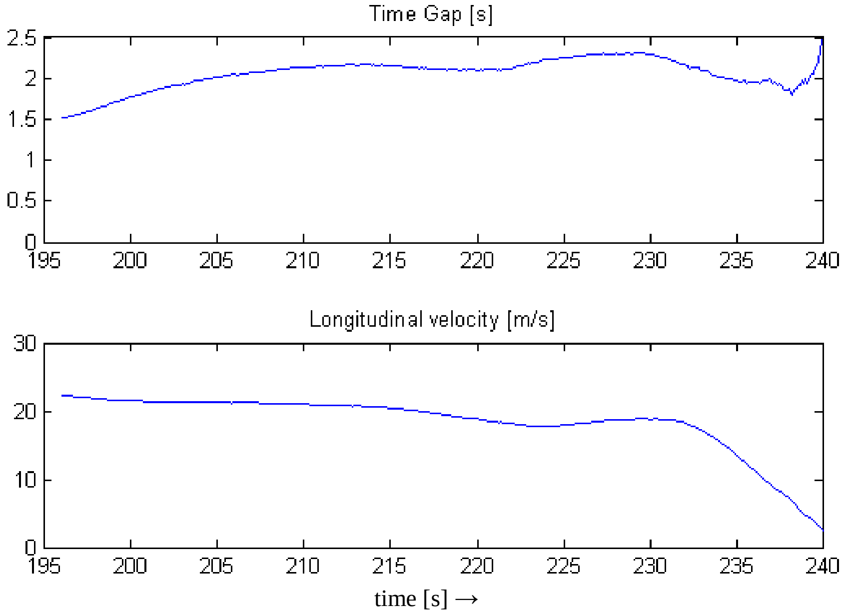


Figure 259: Time gap in the follow to stop situation

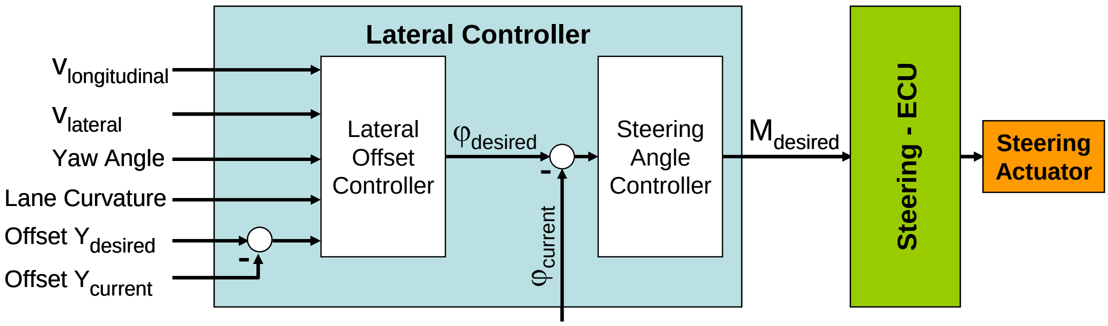


Figure 260: Lateral Controller

Figure 261 describes a driving situation from a straight road through a sharp curve with the curve radius about 200m and to another straight road (see the top plot *Lane Curvature*).

It is very interesting to show here the excellent ability of the TAP demonstrator by very early detecting a curve (t=365sec.) using intelligent Lane Data Fusion (camera data + digital map). After detecting a curve in front of the vehicle, the ego longitudinal velocity is slightly decreased from 28m/s to 20m/s as depicted in the plot *Longitudinal Velocity*, while significantly increasing the absolute value of the ego lateral velocity from zero to 0.3m/s on the curve (see plot *Lateral Velocity*). Just before the curve exit, the ego vehicle changes to the left lane at the time t=410sec. (identified by dramatical changes of the lateral offset in the plot *Ego lateral offset*), accelerates slightly and reaches to the velocity (23m/s) of a new detected moving car in the front, which is considered here as a relevant object.

The plot *Ego lateral offset* shows that the ego vehicle is kept in the middle of the lane for almost the time in this driving situation. But there is a big deviation of the lateral offset just before entering the curve (marked by an orange circle in the time interval from 373sec. to 378sec.). It is an ambiguity problem of the physical lane markings before the curves in the Volkswagen proving ground that causes a big difficulty even for the human perception.

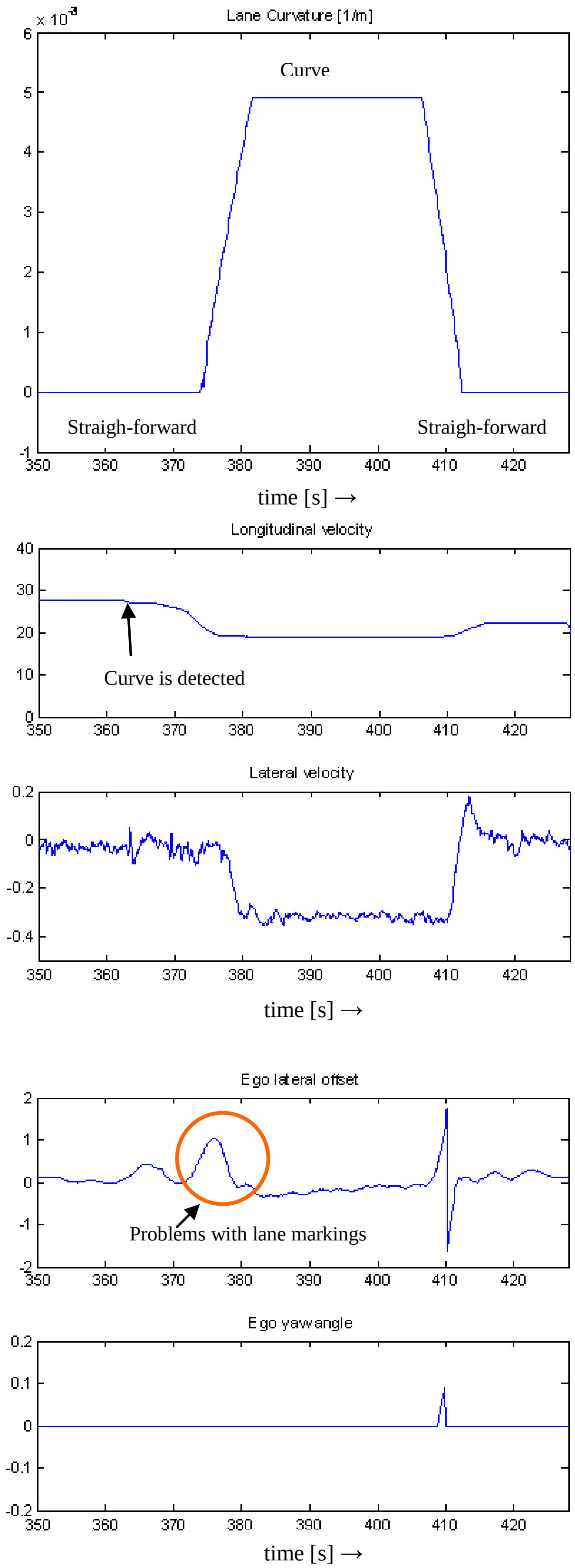


Figure 261: Lateral behavior on the curve

Execution layer

Following modifications were implemented at the execution layer (see D52.2 for details):

- Software modification for steering ECU to handle with the bigger steering moment,
- Software modification for braking ECU to handle with the stronger (e.g. emergency) braking,
- PCs and ECUs used for the implementation of the HAVEit systems have been installed on a rack in the trunk.

Automation spectrum and HMI concept

In Figure 262 the different automation levels of the TAP demonstrator vehicle (blue ovals) and the transitions between these levels (dark-blue arrows) are depicted.

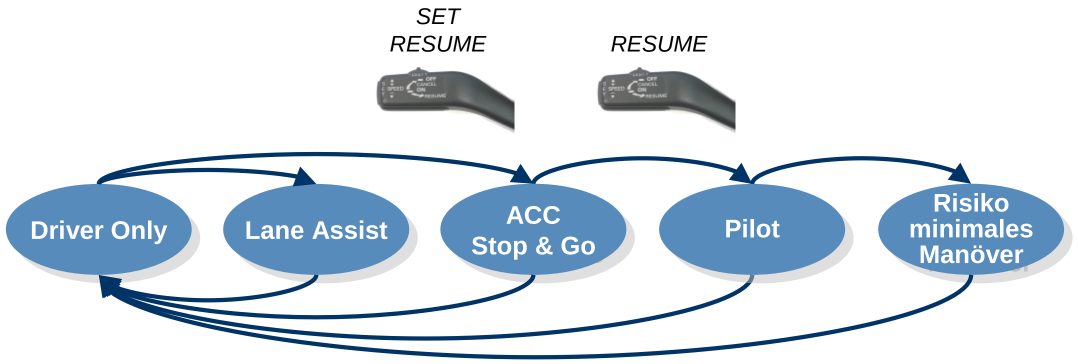


Figure 262: The TAP automation spectrum

The automation levels and the system states of the TAP demonstrator are shortly described in the Table 36. The transition of control is defined as the changing of an automation level to another whereby the driver or the automation gets more control over the vehicle depending on the automation levels.

System states	Description
Driver only	This is the lowest automation level in which the driver has to control the vehicle 100% manually. Only driver assistant systems like ESP, ABS, etc. are active.
Assisted	A level in the automation spectrum between “Driver only” and “Semi-automated”. In the TAP demonstrator vehicle this automation level is covered by a lane keeping system (LKS).
Semi-automated	It is the full-speed Adaptive Cruise Control (ACC) including stop and go (S&G), where the complete longitudinal control is done by the automation.
Highly-automated	The most automated level of the TAP demonstrator vehicle where automation for longitudinal and lateral vehicle guidance is combined (Pilot), but where the driver is still involved in the driving task most of the time.
Minimum risk manoeuvre	A manoeuvre of the vehicle that the automation executes if the driver does not react to a take-over request fast enough. The automation generates this take-over request e.g. if the automation is no longer capable of performing a certain level of automation or the driver does not meet his duties for system surveillance. The minimum risk manoeuvre is a safe stop of the vehicle.

Table 36: Description of the automation levels and system states of the TAP vehicle

The assisted and semi-automated mode can be activated and deactivated in the familiar way, i.e. by pulling the ACC-lever from the ON position towards the driver. The transition of ACC mode to the Pilot mode can be done by anew pulling the ACC-lever. The transition between the automation levels and their preconditions can be summarized in the Table 37.

No.	Transition	Precondition
1	*Driver Only* ➔ *Assisted*	LKS ready *and* manual activation by the driver
2	*Assisted* ➔ *Driver Only*	LKS not ready *or* deactivation by the driver
3	*Driver Only* ➔ *Semi-Automated*	ACC ready *and* manual activation by the driver *and* driver does not brake
4	*Semi-Automated* ➔ *Driver Only*	ACC not ready *or* deactivation by the driver (switching, braking)
5	*Semi-Automated* ➔ *Highly Automated*	Pilot ready* *and* manual activation by the driver *and* driver does not brake *and* driver does not steer heavily *Preconditions for "Pilot ready" are: ego vehicle is driving on a motorway, maximum speed is not exceeded, driver is attentive, driver is not drowsy, lane markings are clearly visible, etc.
6	*Highly Automated* ➔ *Driver Only*	Deactivation by the driver (switching, braking, heavy steering) *or* take-over request *and* response of the driver to take-over request (switching, braking, steering)
7	*Highly Automated* ➔ *Minimum Risk Manoeuvre*	No response of the driver to take-over request (switching, braking, steering)
8	*Minimum Risk Manoeuvre* ➔ *Driver Only*	Driver intervention (accelerating, braking, steering)

Table 37: Transition between the automation levels and their preconditions

For the transparent communication between the TAP system and driver, a new easy and transparent HMI concept has been developed (see Figure 263). To achieve this transparent HMI concept and comprehensible transition between the automation levels, a lot of studies on the different simulators and also with test drivers on the real driving have been successfully done for validation and optimization.

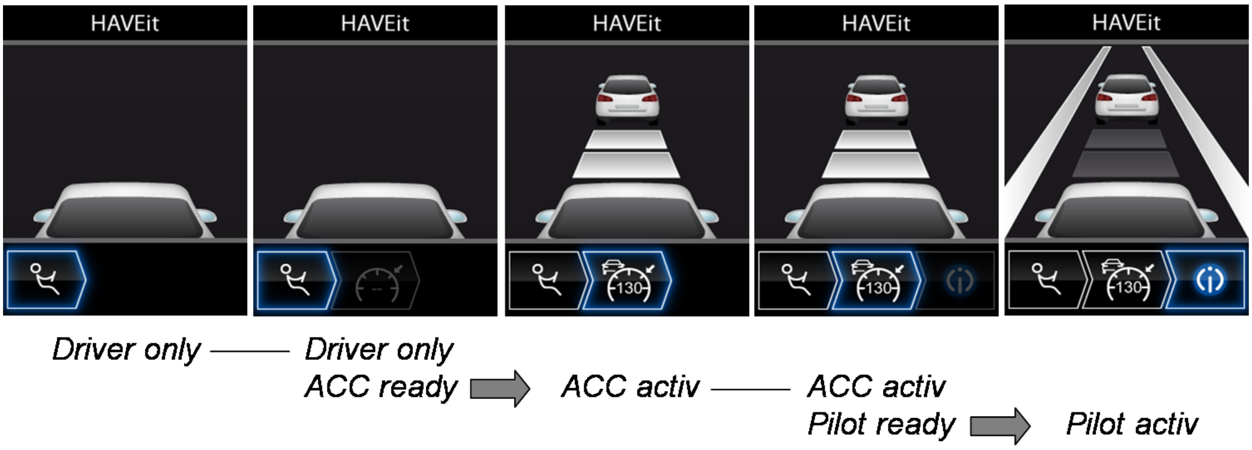


Figure 263: HMI concept

In case, that for some reasons the Pilot mode can not be maintained anyhow by the system, a Take-over-Request message will be immediately displayed in the central instrumental display and an acoustical signal will be generated to inform the driver to take over control (see Figure 264).

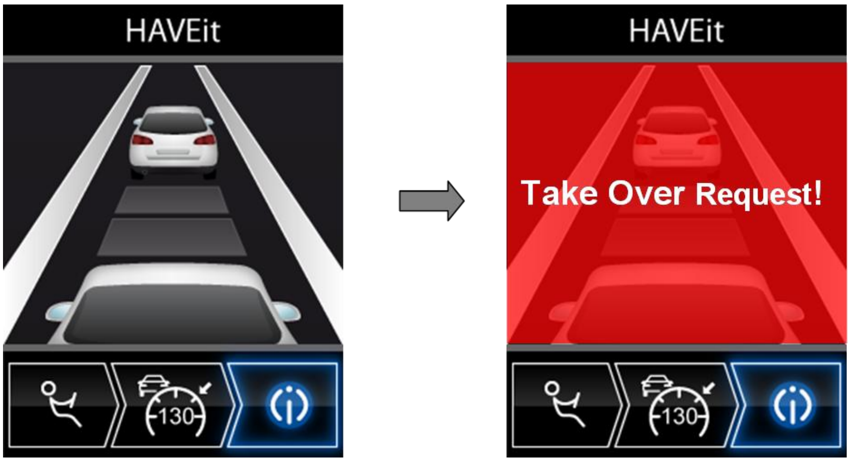


Figure 264: Take-over-Request Message

7.3.2 System validation

The TAP system development consists of many working packages in the different development frameworks, from HMI concept and DMS through Sensor Data Fusion in the ADTF to the implementation of the longitudinal and lateral controller in the Autobox. The validation process can be divided into several steps:

- First, the algorithms of each working package were implemented and validated in their own frameworks using simulation data or measuring data recorded last time. At the end of this stage, a code review has to be done to make sure that the source codes are understandable and bug-free.
- Second, the different components were installed and integrated in the vehicle. All components were tested separately and the results are found in D53.2.
- In the third step the fully integrated system was tested based on the defined use cases described in D53.3.
- In parallel to the 3^{rd} stage, the software in the loop (SiL) test was performed in the laboratory using the same defined use cases (see Figure 265). The SiL-Tests are especially helpful for some critical use cases, that are very difficult or with a big time consuming to test with on the real driving.

Some typical driving scenarios for the TAP functionalities, which are derived from the use cases defined in D11.1 and D53.2, can be categorized in:

- Scenarios for (de)activation of the TAP system in the different driving situations, e.g. activation of ACC by the driver, activation of the Pilot function by the driver, deactivation by the driver or automatic deactivation of the Pilot function caused by the lane markings or automatic deactivation of pilot function caused by driver in the loop assessment.
- Scenarios for validating the longitudinal and lateral controllers in automated driving modes, e.g. free mode, follow mode, stop & go, Minimum Risk State and emergency stop.

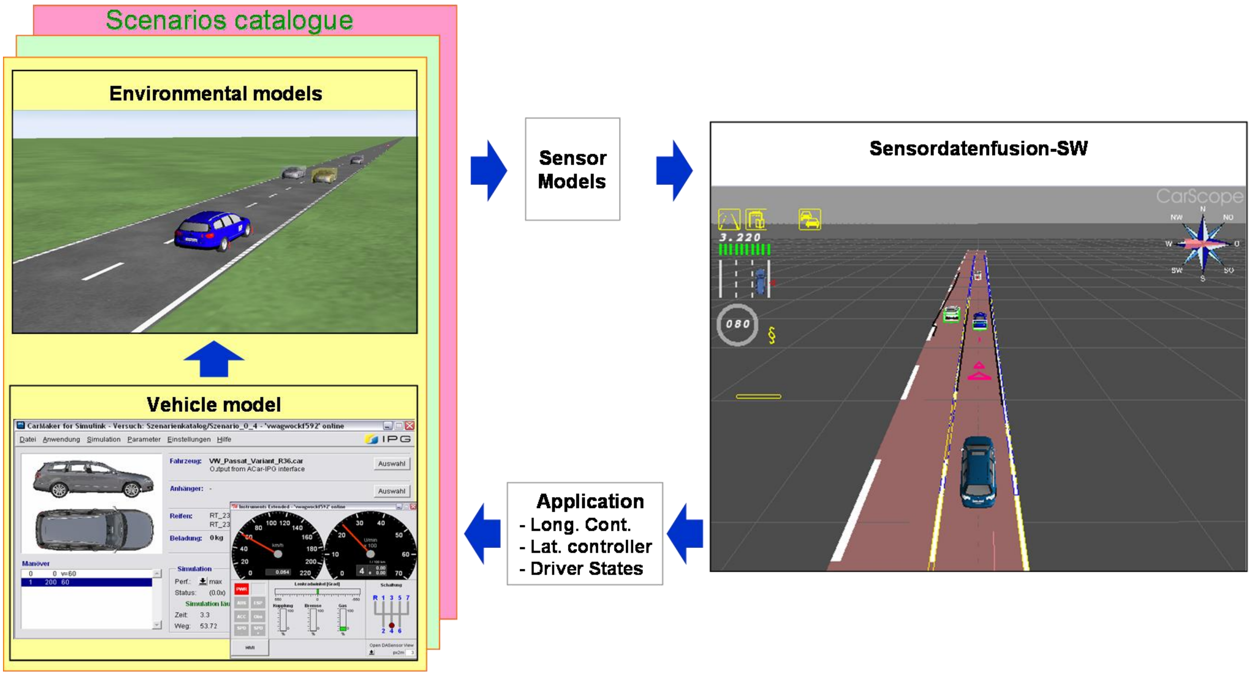


Figure 265: Software in the Loop Framework

Test results for these use cases are described in D53.3. As similar tests were already described for the AQuA demonstrator in the previous section, here a full driving test in complex scenarios shall be described to validate the TAP application. The driving test with all TAP scenarios has been done in the two-lane motorway at the Volkswagen Proving Ground in Ehra, Lessing (see Figure 266).

Figure 266: Driving scenarios for a full TAP test

The test track consists of two straight roads with the length about 800m, and two sharp curves with the curve radius about 300m (on the left side) and 200m (on the right side), respectively. In difference to the normal motorways, those curves are extremely sharp, can be found only for motorway exit or highways. Our full test begins with the free driving (*Free mode*), then *Follow*

Mode with a moving car, then *Stop&Go* and through a short driving in *Free Mode* to *Minimum Risk Manoeuvre* and ends with *Emergency Stop* before a stationary obstacle. The corresponding measuring data for the ego vehicle, objects and lane are summarized and depicted in Figure 267 and Figure 268, respectively. The next paragraphs describe each scenario in detail.

Free mode

At the beginning at t=158sec., the TAP demonstrator is driving alone in a two-lane motorway just before a sharp left curve. It moderately accelerates to reach the desired velocity v=27.78m/s at the time t=168sec. Because it is going to enter the left curve thereafter, its velocity should be slightly reduced for keeping the driving comfort on the curve (see also the time interval marked by the green colour or marked by number 1 in Figure 267).

Follow-to-Stop and Stop&Go mode

At the time t=187sec., a moving car moves from the left lane to the ego lane and is detected just in time as the relevant object, the TAP demonstrator slightly slows down and follows then the target object until stop (see also the time interval marked by the aqua colour or marked by number 2 in Figure 267). Just after Follow to Stop, Stop & Go situation can be demonstrated for a short time duration until this object changed the lane to the left lane at t=275sec.

At the end of Stop & Go situation, the driver turns off the system for preparing the next test with Minimum Risk State. It can be easily remarked by suddenly dropping of the TAP state from Pilot mode to zero (see the top plot in Figure 267). In principle, turning off the system is not needed, but here intended for checking again, if the state machine for TAP will work correctly any time.

Minimum Risk State (MRS)

In the preparing phase for testing the Minimum Risk State, the ego vehicle accelerates from still stand through different modes: *Driver Only, ACC,* and *Pilot* and reaches the new desired velocity of 18.5m/s. At the time t=298sec. the driver mimics the “un-responding” situation by covering his face by a book so that the camera system does not detect the driver, the TAP system informs the driver with a TOR message. Because the driver doesn’t respond to this message within a given time interval, the Minimum Risk Manoeuvre is released at the time t=305sec. to comfortably stop at the same lane with the deceleration of -2m/s^2. After that, the TAP state becomes Off (see also the time interval marked by the magenta colour or marked by number 4 in Figure 267).

Emergency Stop

This scenario can be met while driving into traffic jams or for the collision avoidance with any obstacle detected by on-board sensors. We test this capability of the TAP system with the velocity of 21m/s by driving directly against a stationary obstacle in the ego lane. Emergency Stop will be executed in two phases:

1. First, TAP detects this object at the distance of 100m at t=386sec., slows down the velocity with the ACC deceleration until t=389sec. Because the comfort deceleration is not strong enough to stop the ego vehicle without any collision, the 2nd phase will be initiated.
2. In the 2nd phase, the deceleration will be dramatically increased to the maximum deceleration to bring the ego vehicle to the still stand. In our test case, a short pulse of the longitudinal deceleration with the amplitude of -10m/s^2 is generated and the ego vehicle stops at the distance of about 9m before the stationary obstacle (as depicted in the Figure 267 at the time interval marked by the red colour or marked by number 5).

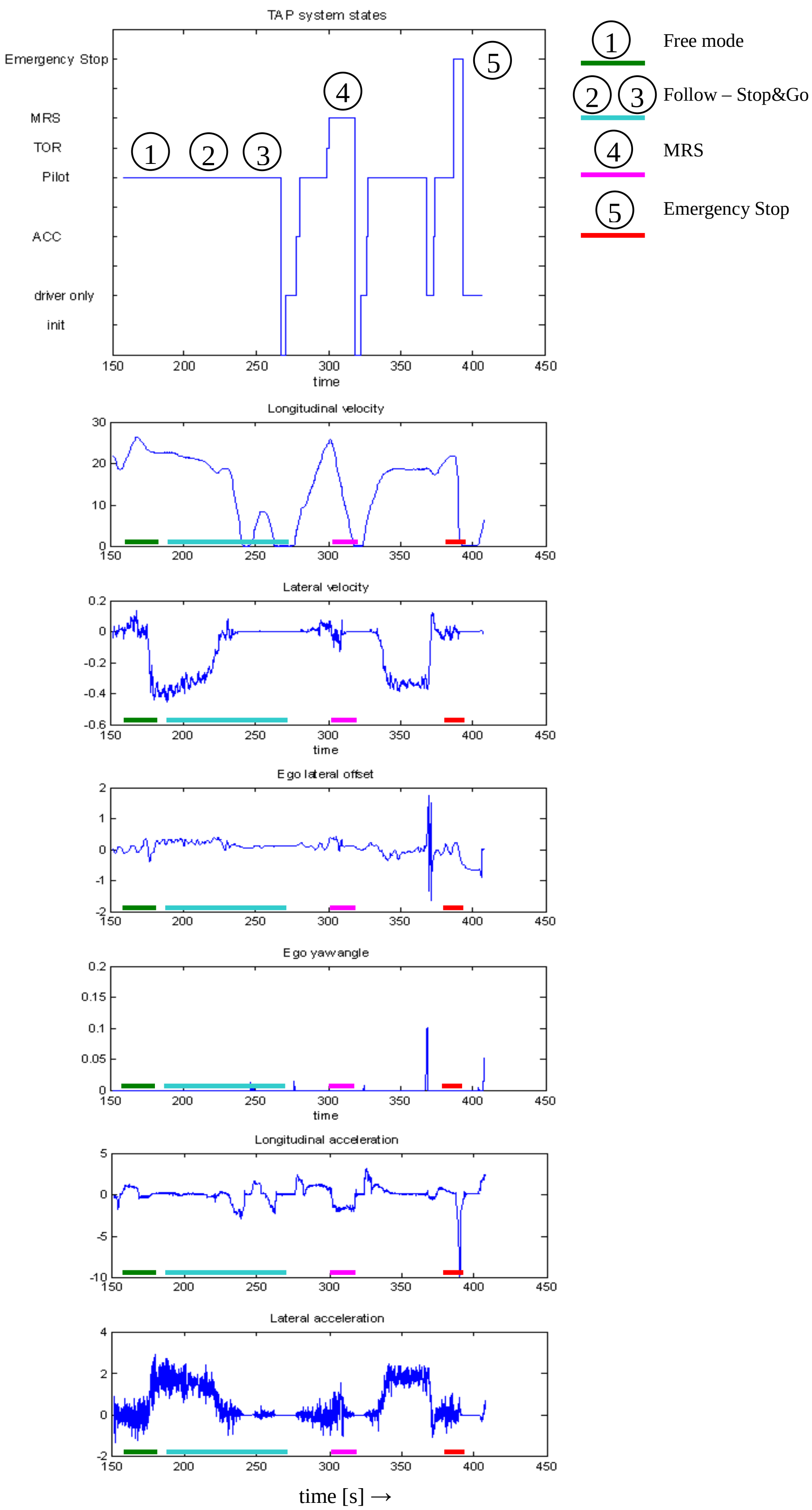


Figure 267: TAP states and ego vehicle data for a full test

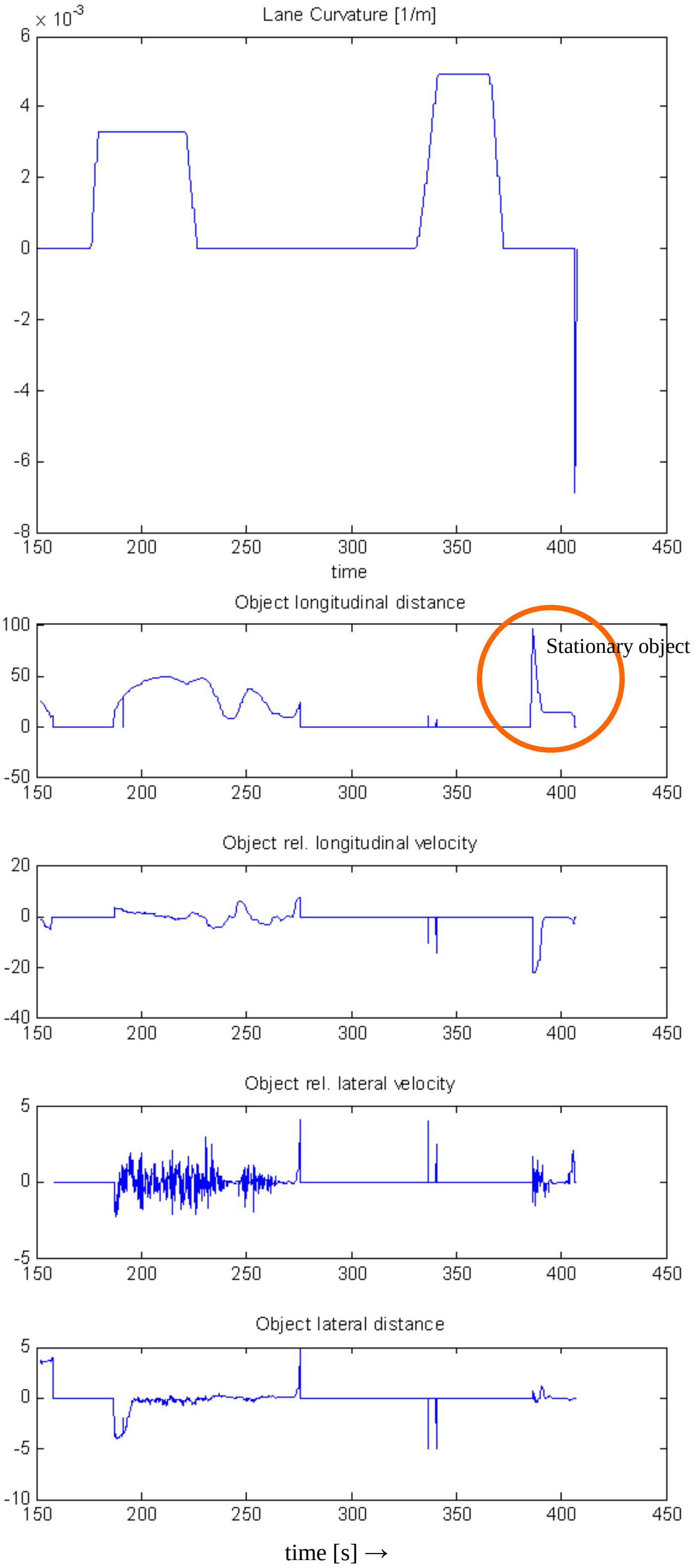


Figure 268: Object and lane data for a full test

7.3.3 Summary and conclusion

The validation measurement described in the previous section illustrates how the TAP application works. All target scenarios for the TAP that were defined at project start (see D11.1) are covered. We can conlcude that the highly automated Temporary Auto-Pilot algorithms and the fully integrated system work as intended. All units are implemented according to the HAVEit generic standard with just minor TAP specific adaptations. The communication and interaction between all the components work together adequately.

In all scenarios derived from the TAP use cases the system reacted to the triggers in a satisfying way. The technical concept and the designed architecture are proved as valid. All objectives related with HAVEit challenge 5.3 were achieved.

7.4 Challenge 5.4: Active Green Driving

Introduction

This section reports on the first results of the Active Green Driving (AGD) application. In all parts of the application, simulation on desktop PC and on more elaborated driving simulators with drivers in the loop, have been planned steps in the development.

In the AGD application, information obtained by sensors installed on the demonstration vehicle to monitor the external traffic environment, are used to predict the future driving horizon. By predicting near future changes (within the horizon), as for example speed and elevation, the, in the best case optimal, use of the demonstration vehicle's hybrid powertrain can be found and executed automatically without intervention from the driver. The key difference to a conventional powertrain is the brake regeneration and temporary storage of electrical energy. Furthermore, without restrictions to the kind of powertrain deployed, information within the prediction horizon is also used in a Driver Coaching System to help the driver handle the vehicle in a more fuel efficient manner by advice via graphical display and haptic accelerator pedal. These are the functionalities that are evaluated here.

The AGD demonstration vehicle is up and running as a test bench for speed and load prediction and subsequent optimal control of the powertrain and for driver coaching. The prediction is based both on static route information taken from the in-vehicle database, using GPS to acquire the vehicle's position such as maximum likely speed and bus stops, but it is also based on dynamic traffic ahead taken from the front mounted laser scanners. The first vehicle tests show that the estimation of future vehicle speed works well.

Currently, there is work in progress to make the prediction more robust. The Energy Management works well in simulation and shows that fuel actually can be saved. The Driver Coaching System has been installed in the vehicle.

7.4.1 Demonstrator configuration

The AGD demonstrator architecture follows the general HAVEit architecture. However, as the ADG application is directed towards optimum control of the vehicle's hybrid and to driver coaching, activities in HAVEit mainly concerned perception, HMI and execution layer of the generic architecture.

Perception layer

Laser scanners

Two SICK (former IBEO) LUX laser scanners are installed and operating on the demonstration vehicle. This setup consists of the two laser scanners plus an ECU for processing of the laser scanner data. These sensors provide forward object detection as well as a basic form of lane and road barrier detection. Since their original installation in early 2009, these sensors have received many software updates from SICK based on VTEC feedback, aimed at improving the object tracking behaviour, and therefore bringing their performance now to that which is required by the perception layer. The laser scanners are the primary source of forward object detection, and provide the most accurate estimates of object positions.

An example of the laser scanner system performance is shown in Figure 269 and Figure 270, where the host vehicle is driving on an open freeway. Here, two forward vehicles are tracked. Both vehicles have just passed the host vehicle, and the leftmost vehicle is travelling faster than the right vehicle. Figure 269 visualises the scene in the SICK visualisation software, where one can see the object tracks from a top view, as well as the individual laser scanner scan points. The road barrier on the far left is also visible. The two pairs of rays, green and blue, originating from the bottom where the ego vehicle imagined, depict the field of view of the left and right scanner. The host vehicle is travelling at 72.1 km/h. Figure 270 is a time plot of the scenario, showing the measured objects' longitudinal distances as transmitted from the laser scanners via CAN. The plots are smooth, and accurately represent the object behaviour. One can also see that at approximately 24000 ms, the left object track is lost. This is due to it becoming occluded by the right vehicle.

V2V communication

Vehicle-to-vehicle (V2V) communications devices, developed by EFKON, are used to transmit CAN data between vehicles. The V2V modules in HAVEit use infrared light as a transmission medium. This requires direct line of sight and optical alignment of the units before communication can occur. The AQuA truck is equipped with a front and rear V2V transceiver, which can communicate with the similarly equipped AGD bus.

After some integration work, the V2V modules are now capable of transmitting CAN data between each other. Transmitted information includes parameters such as vehicle speed, yaw rate, and steering angle. The transmission capability of the units was tested both statically and dynamically:

- The static tests involved arranging both vehicles in a range of positions to examine the extreme values of the operating capability, for example, max/min range and lateral offset. The performance of the modules in these tests met their original specifications.
- The dynamic tests involved driving both vehicles together on public roads with a lateral distance of approx 30-50m. This enabled the verification of connections reacquisition after a connection is lost when manoeuvring through sharp corners, and robustness to general driving vibrations etc. The performance during the dynamic test was found to be sufficient.

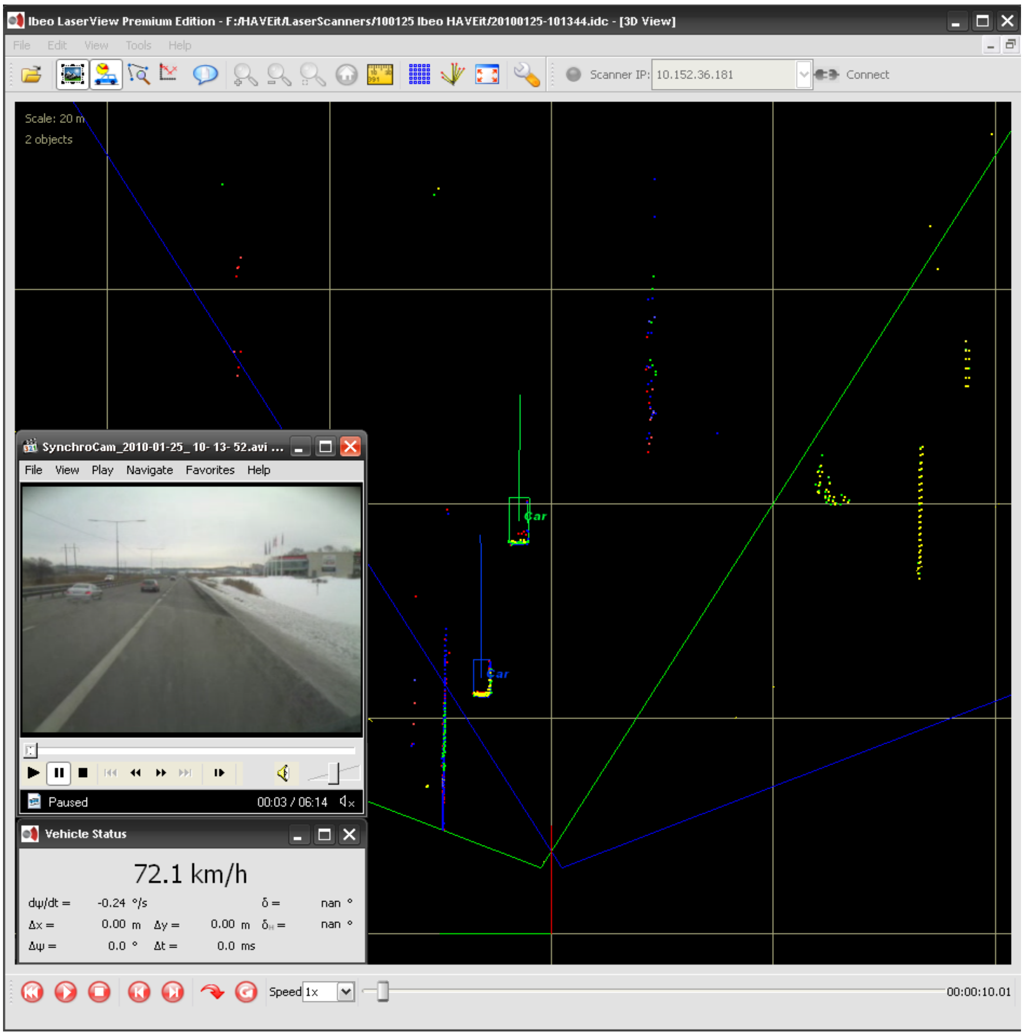


Figure 269: Top view laser scanner visualization, with accompanying video

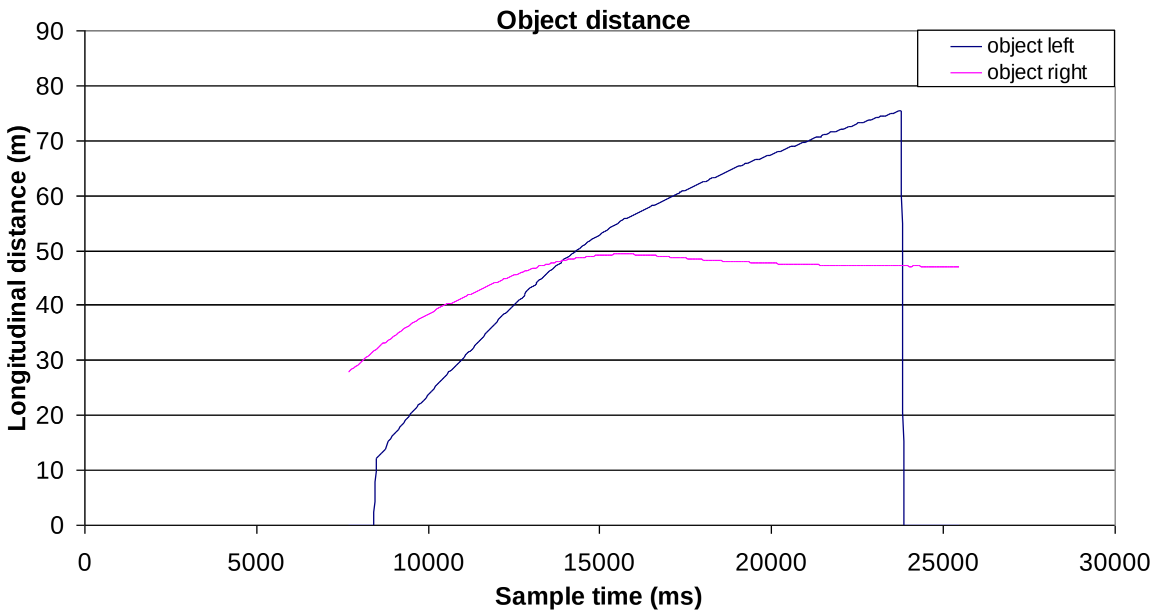


Figure 270: Object distance measurements for the scene in Figure 269

Figure 271: The vehicles used for V2V system verification

Some results from the dynamic testing are shown in Figure 272 and Figure 273. In this test, the WP5200 truck lead the WP5400 bus and the data sent from the truck was logged on the bus. The route driven during this test is shown in Figure 272, where several sharp turns, roundabouts and shallow curves are connected by straight roads. Each straight road is designated a letter (A through E).

In Figure 273, the number of CAN messages received by the bus was plotted, along with the correspondence to the straight sections of road. As can be seen, CAN messages are sent fairly consistently on the straights when the V2V modules are well aligned, and drop to zero when travelling through 90 degree turns. The different coloured lines represent different CAN messages (0x32C, 0x32D, 0x32E) as blue, red, and green, respectively.

Sensor Data Fusion (SDF)

The SDF system is intended to receive data from all the sensors, to perform processing of the data, and finally to send the processed data (perception model) to the relevant vehicle control systems for the AGD functions. The sensor fusion platform developed for AGD has been developed in close cooperation between ICCS and VTEC. ICCS has developed the detailed fusion algorithms, while VTEC has integrated those algorithms into the AGD demonstration vehicle. The Sensor Data Fusion is a mostly common platform shared with the AQuA truck.

The SDF framework is implemented in Simulink and the detailed fusion algorithm modules are implemented in C++. It all runs on a PIP11 xPC (robust automotive PC). The fusion architecture is shown in Figure 274. For a more detailed description of the SDF system and its algorithms, refer to D54.2. The input sensor configuration is described in D54.1.

Figure 272: Test route driven during part of the V2V communications testing

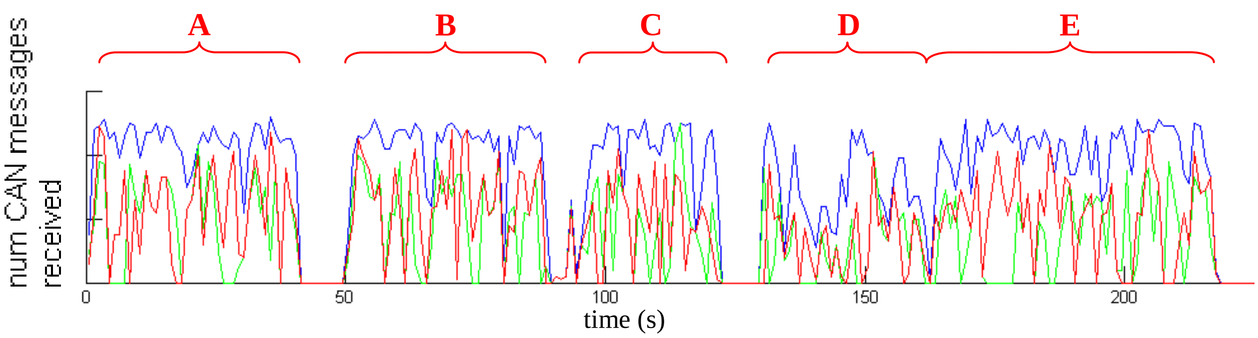


Figure 273: Graphs indicating the transmission of CAN messages transmitted in V2V testing

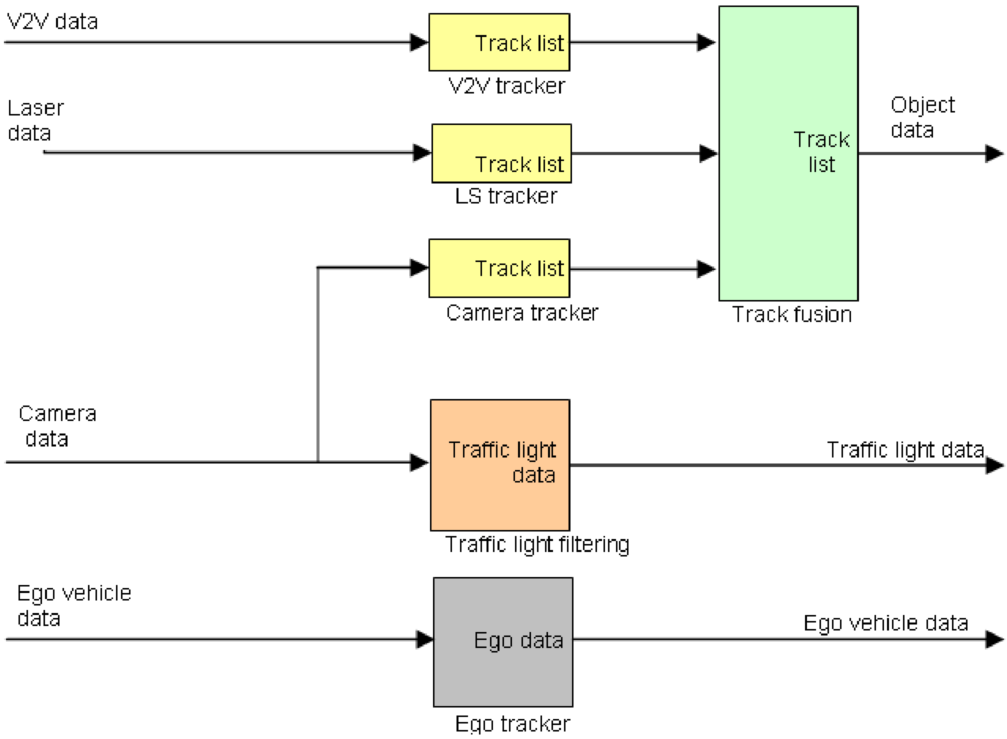


Figure 274: Sensor data fusion architecture

Table 38 provides a summary of the output information derived from the SDF system. The SDF system has been successfully implemented and thoroughly tested in two steps:

1. Offline testing by ICCS and VTEC. Sensor data has been logged from the AGD bus and replayed offline to feed the SDF with data in a development framework. This has enabled a detailed analysis and debugging of many aspects.
2. Real-time testing by VTEC in the bus. This involves driving in a variety of scenarios and observing the output in a visualisation framework. This can be a quick way to evaluate overall SDF system performance.

Object Data	Traffic Light Data	Ego Data
x, y position and std dev	traffic light position	yaw rate
x, y velocity and std dev	traffic light state	velocity
x, y acceleration and std dev		acceleration
class		stand still flag
lane assignment		
track time		

Table 38: Summary of output data from the SDF system

Command layer: Joint System

The speed and load prediction is a vital supplier of information to the Energy Management. Optimization of future energy consumption only becomes possible with this prediction. Here, a couple of logs from the vehicle will be shown. The actual (red) and predicted (blue) speed profiles of a section of the Mölndal route are compared in Figure 275 and Figure 276. The Mölndal route is a short and closed circuit of city bus suburban traffic nature and with hills representative for the Göteborg area. The first figure shows the speed (m/s) as a function of distance (m) and the second figure shows the speed (m/s) as a function of time (s). The surroundding traffic affects the prediction.

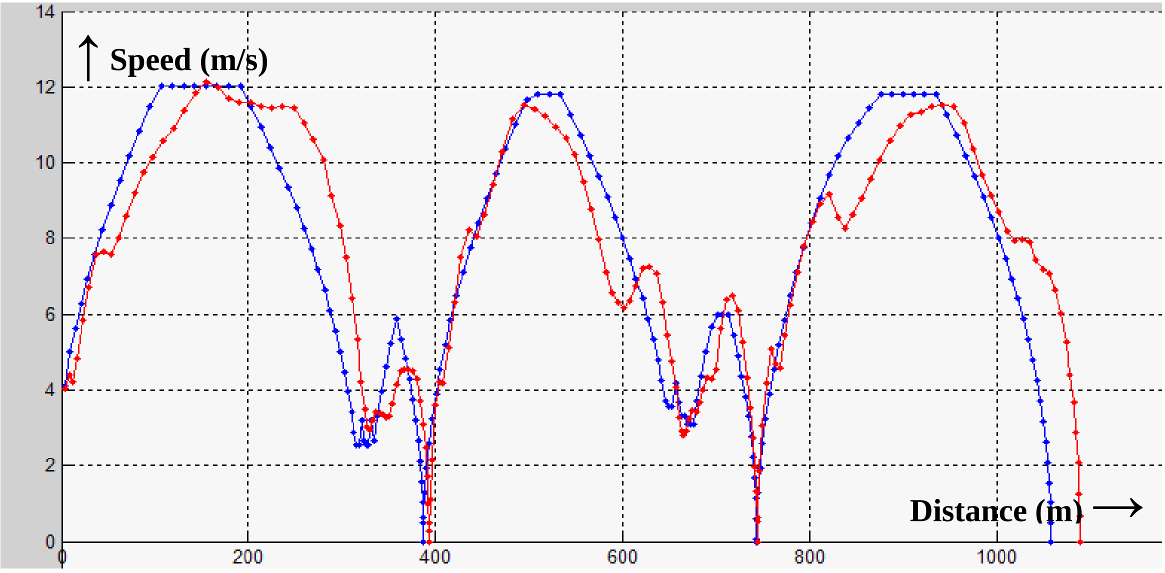


Figure 275: Speed prediction as a function of distance. Blue line is predicted and red is actual.

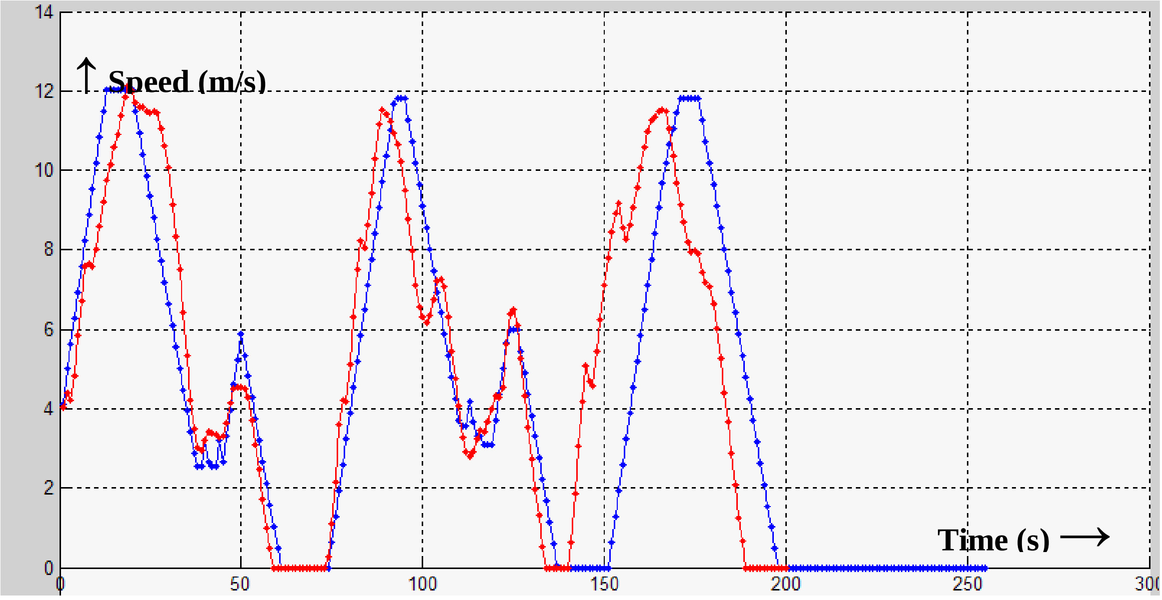


Figure 276: Speed prediction as a function of time. Blue line is predicted and red is actual.

Figure 277 shows the predicted required torque (Nm) for the same instant as in previous figures. Positive torque means acceleration request and negative torque means a brake request, the request is an interpretation of the drivers demand exercised by the gas and brake pedals. The retardation request can be effectuated by the combustion engine, the electrical generator and finally, if necessary, the service brakes.

The prediction horizon is 200 seconds, which is noticeable in the second and third figure. A new profile is generated every second. In this case it means that the horizon is about a kilometer ahead. There are two bus stops along the depicted section, the first one at 400 m ahead. Here, the predicted speed is also zero (and the torque request on the powertrain is zero).

Since the speed as a function of distance is known from the route database, the accuracy does not deteriorate with prediction horizon. For the second plot distance is translated into time and the estimation is further improved by the driver model. To the left in the figure the prediction is close and should be quite accurate, and to the right, the prediction is far in future time which means that the accuracy is expected to be worse because of unexpected delays and varying vehicle speed. However, the speed profile for the Energy Management is continuously updated. There is a noticeable shift in time due to shorter than expected bus stop around 140 seconds.

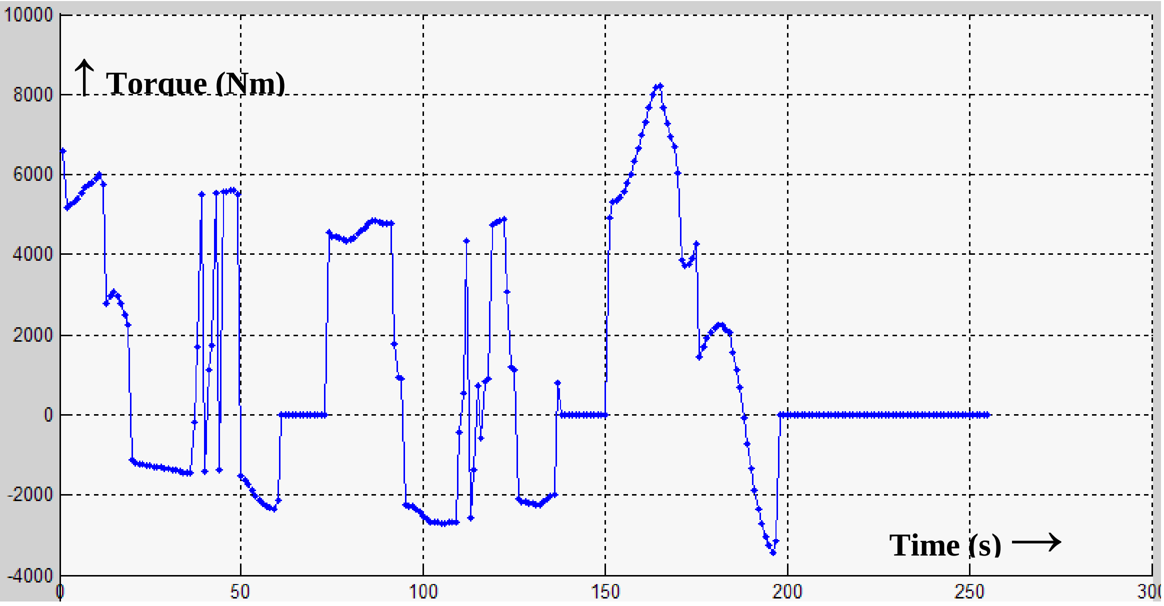


Figure 277: Torque prediction as a function of distance.

Execution layer: actuators

The existing powertrain production software has been modified to enable AGD, essentially overriding default Energy Management functionality. However the actuator control, responsible for example for the actual gear shift, is still active as well as the functionality that aims at protecting the machine from damage.

In Figure 278, Figure 279 and Figure 280 data logs taken from the AGD demonstration vehicle show the operation of its hybrid powertrain. In the first figure the vehicle speed and driver's demand are plotted; there are two accelerations, both starting from an initial velocity of about 5 m/s and ending at a velocity of about 15 meters per second. Figure 279 shows the corresponding torque (the range has been obscured) delivered by the internal combustion engine and the needed flow of Diesel fuel (the range starts at zero flow and the scale has been obscured by purpose since these are preliminary figures). From the indicated torque, the friction torque has been subtracted, i.e. when no fuel is injected the combustion engine will retard the vehicle. The third figure, Figure 280, shows the torque of the motor/generator and the battery's state of charge (SoC). When the torque is negative the electrical machine operates as a generator, braking the vehicle with a torque of opposite sign to the rotational velocity and the SoC increases. The gear shifts are clearly visible. The figures show that the system is working, but there remains much tuning and testing before validated estimates of fuel savings can be reported.

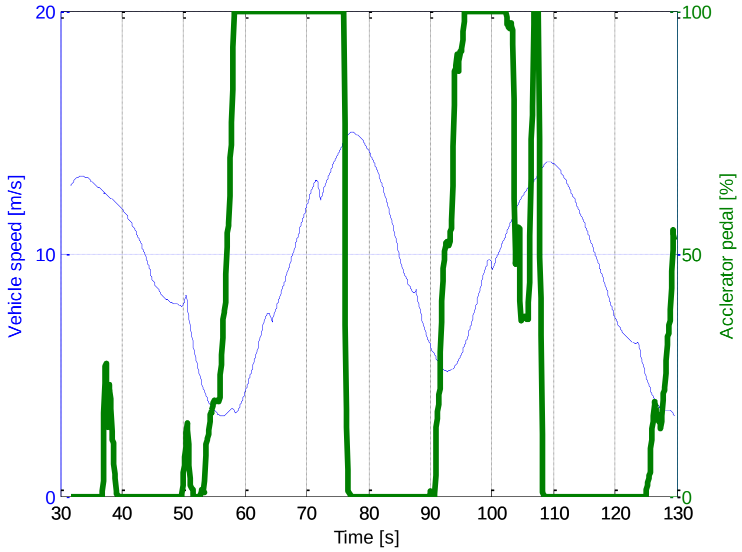


Figure 278: Vehicle speed (left axis, blue line) and gas pedal (right axis, thick green line).

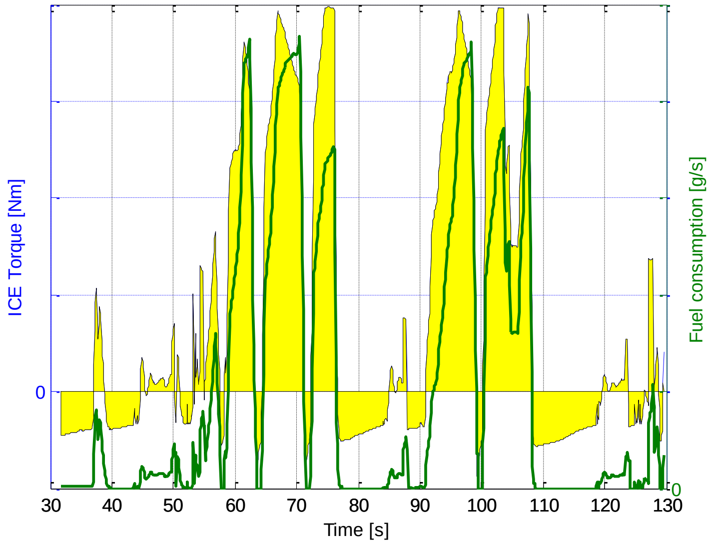


Figure 279: Generated torque by combustion engine (left, yellow) and mass flow of fuel (right, green).

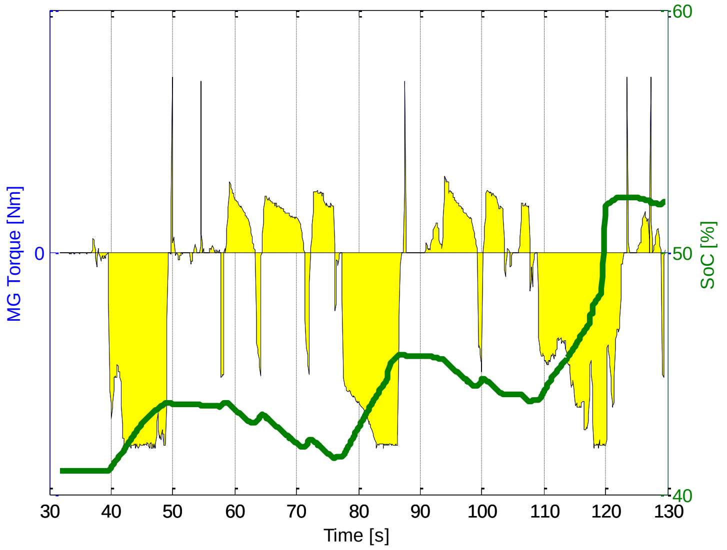


Figure 280: Electrical machine torque (left, yellow) and state of charge (right, green).

Driver interface

This part describes the graphical displays developed for the AGD system. The graphical expression is by purpose similar to that of the AQuA system, although the information content is different. The graphical user interface, GUI, is developed using an iterative design process with several user clinics and expert evaluations that provide feedback to two redesigns of the GUI.

Driver's display

The driver's display is divided into left and right see Figure 281 and Figure 282. The left part of the display consists of an Eco value represented as a plant or branch with leaves that turn from neutral beige to green. The amount of green leafs indicates the Eco value, which is a combination of the different KPIs as calculated by the Driver Coaching System.

Figure 281: Examples of GUI's of the second version.

At the lower left corner is a speed gauge representing current speed with a red needle. There is also a green circle sector with fuzzy edges indicating the recommended (or target) speed, that renders the highest Eco value. The right side of the display is used for information pop-up windows. The pop-ups are divided in two types, as well as they got two positions on the display. Pop-ups placed at the top, called tip pop-ups, contain information that requests the driver to act; “avoid hard braking”, “down hill ahead, please coast” etc. Pop-ups presented at the bottom, called feedback pop-ups, contain information that gives feedback to the driver; “well done, good coasting” etc, see Figure 281 or Figure 282.

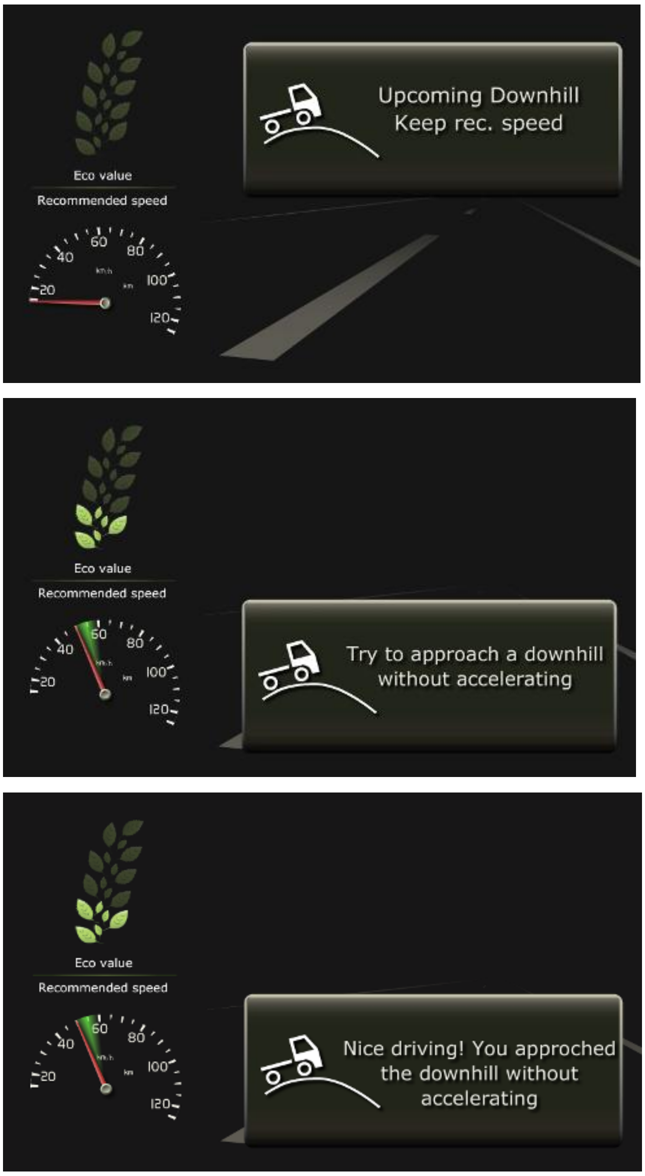


Figure 282: Example of the popup strategies.

The reason for dividing the pop-ups in two different presentation areas and making them smaller than the AQuA is mainly that they should be regarded as information and not too urgent – they should not interfere too much with the driver. The placement is suggested to emphasize that information presented at the top is more prioritized and urgent. With one glance at the display the driver can acknowledge whether it is contemplative feedback or a request to react immediately. The size is also a way of not stealing as much attention from the driver and both the size and the shape should indicate: “Here is information, if you want it”.

A new feature of the second version of the AGD GUI is the Post trip feedback. The post trip DCS feedback screen appears after a trip and the driver can receive feedback of his driving performance according to eco driving coaching functionality.

At the upper part in the left corner the average eco value is presented in the same form as the instant Eco value previously presented while driving. Below the average Eco value an arrow and numbers and leaves show the trend compared with the last trip. An upward pointing arrow means a positive trend, i.e. the driver had a higher Eco value than last trip. If the arrow is pointing downwards the driver had a lower Eco value than last trip. The number below the arrow presents the difference, positive or negative, compared with the last trip.

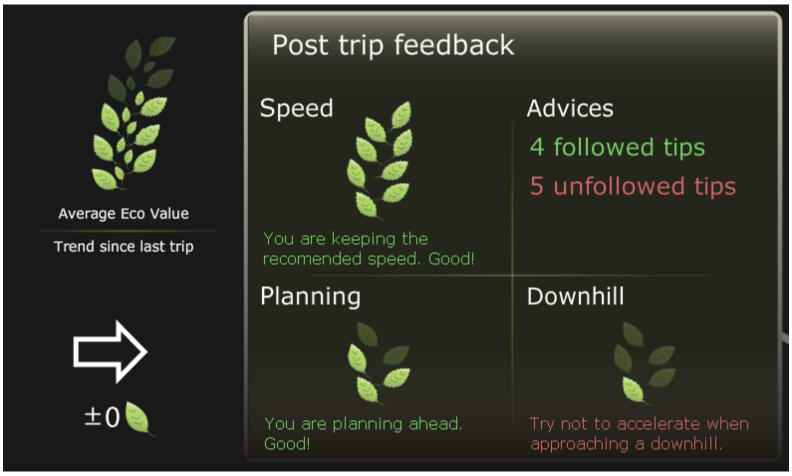


Figure 283: Example of the GUI for the post trip feedback screen.

The post trip feedback box to the right contains four areas:

- The upper left area is for speed performance. It shows a meter containing leaves presenting average speed performance according the speed KPI. Below the meter the most common feedback message connected to speed performance during the trip is shown. If the message is positive it is displayed in green and if it is negative it is displayed in red.
- The upper right area is for advices the driver received during the trip showing the number of followed and disregarded tips.
- The lower left area is for planning, presenting the planning performance in a leave meter with the most common feedback message connected to planning performance.
- The lower right area is for downhill performance, presenting the downhill performance in a leave meter with the most common feedback message connected to downhill performance.

The sum of all leaves in the post trip feedback area is the same as the average eco value.

Demonstration passengers' display

When demonstrating the AGD for an audience during the final event, a large display, visible for all passengers, is mounted in the demonstrator bus purely as a tool for the presenter during demos to enhance the communication with the audience. The presenter can switch between two modes, see Figure 284 and Figure 285:

The first mode presents the hybrid driveline, how the energy transforms on the driveline, and the prediction horizon, what is ahead of the vehicle such as speed limit changes, bus stops, roundabouts and a graph showing recommended and actual speed versus horizon in meters. The second view presents the prediction horizon like in the first mode, and the drivers view GUI.

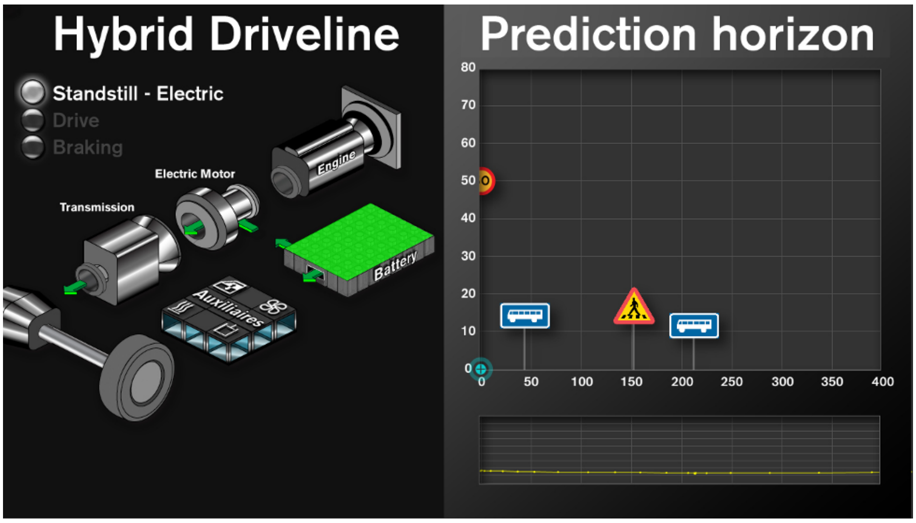


Figure 284: Hybrid driveline and prediction horizon view.

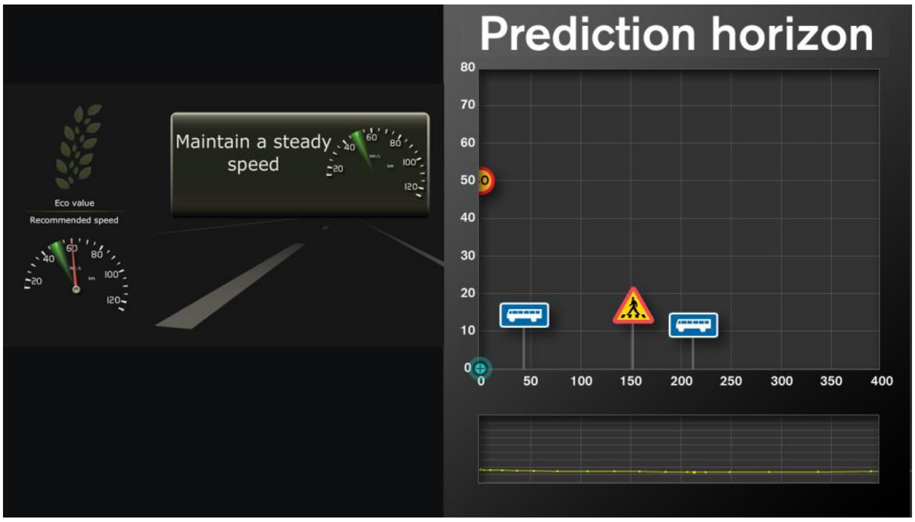


Figure 285: Drivers view GUI and prediction horizon view.

Haptic gas pedal

Tests with potential users in the simulator showed the use of making the characteristic of the haptic feed-back gas pedal stiff when the driver exceeds the re-commended speed.

The behaviour of the haptic pedal becomes normal when the speed is lower than the recommended speed. This feature was very popular by our respon-dents. It was refined and further developed.

Figure 286: Haptic gas pedal installation

7.4.2 System validation

For the development of the driver coaching system (DCS) simulator experiments were intensively used, in particular in the use case “Predicted acceleration”. This use case describes the functionality of the DCS when the driver is speeding. Small scale tests in a driving simulator have been performed as a way to refine the DCS and the outcome was used in the algorithm development phase. User test on the road has not been possible and are difficult due to changing situations. Further, the relation to HMI was in focus here, not the relation between actual fuel savings and driver coaching.

The truck simulator was used during the user clinics and expert evaluations. The simulator consists of a truck cab with buttons, steering wheel and pedals connected to simulator computer that generates a road environment that is projected on a surface in front of the cab. This simulator computer was connected to the HMI computer that generated the GUI on the secondary display in the cab. The respondents drove the same predefined route with traffic objects e.g. cars, traffic lights and road signs. The HMI computer receives signals from the simulator computer about what and when to present messages and eco values etc. During the clinics and tests the same route and traffic behaviour were used to be able to compare results.

Figure 287: Driving simulator

Figure 288: Driving simulator interior setup

The validation of the DCS is based on feedback from test drivers. Their view of the system is to a high degree determined by the graphical HMI. Here is a selection of their comments:

Comment	Recommendation and possible actions
Pleasant feedback wile driving.	
Just enough with information while driving.	
Unclear in what way the driver shall behave regarding the DCS tip: "Adjust to recommended speed". Should the driver brake or remove the foot from the pedal?	Better introduction to the feature and an instruction manual, or more informative message
It is difficult with two speedometers (the ordinary one and the one in the SID)	
It is hard to understand how the DCS tips are connected to the leaves.	Better introduction and instruction manual
The information about a new speed restriction comes too late.	Tuning required
The leaves disappear rather quickly, a bit too quickly.	Tuning required
The weighting of the KPIs is hard to understand. Does a bigger leave indicate that it is more important?	Better introduction to the feature and an instruction manual, or more informative message
Don't like text based popups.	
The instructions can be more explicit and shorter	Trade-off with clarity
Over speeded a lot → not seen in the number of leafs in the post trip feedback	Tuning required
Messages are easy to understand. The same goes for the post trip info	

Table 39: DCS user comments

Some of the issues these comments address could be adjusted by fine tuning parameters with minor adjustments. A better introduction to the system is another remedy. The drivers need to acquaint themselves with the system. The tips and feedback gets easier and faster to understand once the drivers have familiarized themselves with the system.

The DCS presented information on the secondary display is placed outside the line of sight, which makes it awkward for the driver to follow information changes. Instead, the DCS should present information on the primary display (not available to the project).

One suggestion from the test drivers was to present the Eco Value as a percentage of optimal fuel efficient driving. It was suggested that the DCS should compare the driving behaviour to an "optimal driver". Though, what is an optimal driver? This question is out of the scope but in some sense the DCS tries to advice the driver how a driver should act to save fuel.

It was agreed that the driver needs to have the opportunity to disable the DCS. Also, the incentives for the drivers to use a DCS were discussed, what is his motivation for doing so? This is the core problem in driver coaching, and one important piece, which is out of scope of this project, is believed to be the off-line follow up by the employer. However, the post trip feedback presented below when discussing the relevant use cases is a step in this direction.

There are four scenarios which will be identified on-line that have a potential to reduce the fuel consumption in the active green driving (AGD) application.

Use case 1: "Predicted deceleration and acceleration"

There are several traffic situations where sensor information might indicate that the vehicle is most likely to stop, or drive very slowly, within the near future. For instance, train crossings, roundabouts or sharp curves might be known using map-based information, see Figure 289.

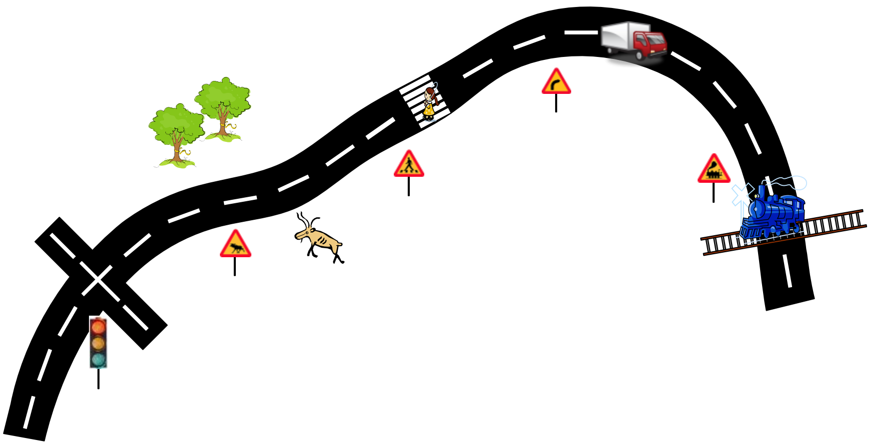


Figure 289: Predicted deceleration followed by acceleration.

In such situations, the driver will most likely release the accelerator and perhaps even push the brake pedal, with the purpose of decelerating the vehicle to stand still or go in very low speed. After a while, the driver releases the brake pedal (if pushed) and starts to push the accelerator to increase the speed of the vehicle. With this information at hand to the powertrain controller, the split of the power flows between the internal combustion engine (ICE) and the energy storage, can potentially be adjusted, perhaps the ICE can be turned off and on and favorable gears can be selected. These are actions with the purpose to reduce the fuel consumption and pollution. The results from use case 1 are divided into an energy management part and a driver coaching part since they could be operated independently of each other.

Energy management

Figure 290 below shows a driving case where a brake sequence is upcoming, followed by a 15s stop, and then an acceleration. The resulting driveline control is as follows:

- A stop is predicted some seconds ahead where energy recovery is possible and hence, the energy management controller allows for energy usage event though state of charge is at a low level.
- The stop approaches, the driver starts decelerating the vehicle which allows for energy regeneration.
- At the acceleration sequence after the stop, the diesel engine is assisted by the electric machine using the stored battery energy reducing the state of charge (SOC).

In Figure 290, the green set speed signal is based on predictions. Using the predicted future operation, the driveline is controlled with respect to the upcoming time of operation, not just the current time instant. Electric energy could either be saved for later usage, or be used to assist the diesel engine (if for instance a regeneration possibility is approaching). With required accuracy on the laser scanners and colour camera deceleration/acceleration phases due to objects such as train crossings, pedestrians, stopped vehicles etc could have been predicted. This would allow for better energy management that would further decrease fuel consumption and pollution levels.

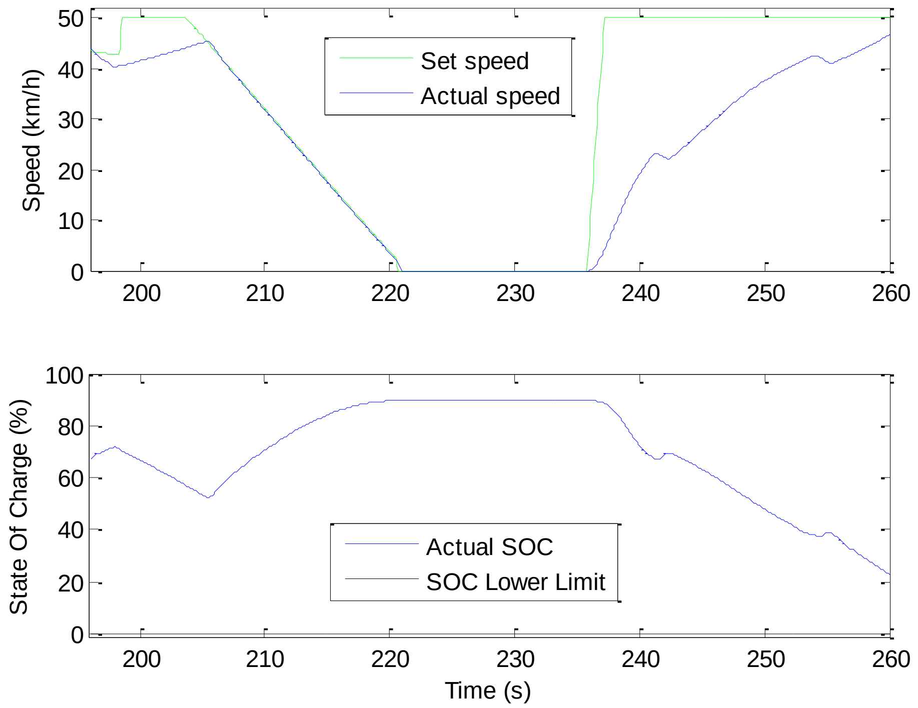


Figure 290: Predicted deceleration followed by acceleration, energy management.

Driver coaching (see also simulator results mentioned above)

The vehicle fuel consumption is heavily dependent on the driving style and hence, fuel economy improvements could be achieved by coaching the driver to a fuel efficient way of driving. However, in this concept study main focus has been put on the human machine interface (HMI).

A city bus is typically driving on roads similar to the one presented in Figure 289 where many accelerations and decelerations occur due to different road limitations and objects.

Figure 291 visualizes a typical HMI view during normal bus driving in city traffic, where road signs up to 150m ahead are shown. The driver is also getting direct feedback on how different driving situations are handled, and an overall Eco Value which is connected to fuel efficient driving.

Figure 291: Driver coaching GUI with upcoming road signs and driver feedback.

Figure 292: Driver coaching GUI with vehicle speed feedback.

In Figure 292, the road speed limit is 50km/h but the recommended speed is 40km/h due to the current traffic situation including prediction on upcoming objects and/or road limitations. In this case the driver is asked to adjust to the recommended speed from the current speed 55km/h. In addition, a force feedback function in the accelerator pedal is used to actively provide a resistance as long as the speed is above the recommended speed. As soon as the vehicle is stopped, for example when the bus route is completed he/she can demand for a post trip feedback window that summarizes the driver's behavior compared to the recommendations along the route, see Figure 293.

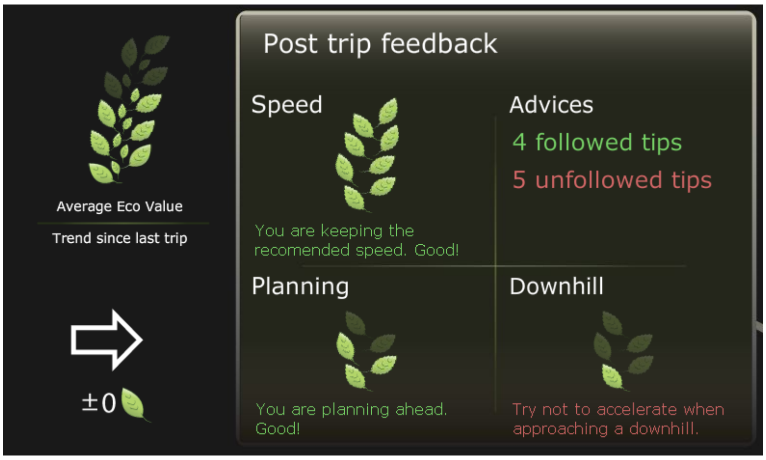


Figure 293: Driver coaching – post trip feedback.

In this case, the driver has obeyed approximately half of the tips. The number of green leaves denotes the "Eco value" connected to fuel efficient driving. The arrow in the bottom left corner is a concept proposal that is intended to show if the driver has improved or become worse since last driving occasion.

Use case 2: "Predicted speed limits"

There are some traffic situations where sensor information might indicate that the vehicle is most likely to reduce (increase) its speed in the near future (but not followed by a consecutive acceleration, as in the previous use case). For instance, using map-based information, it might be known that a speed sign reduces (increases) the speed limit, see Figure 294.

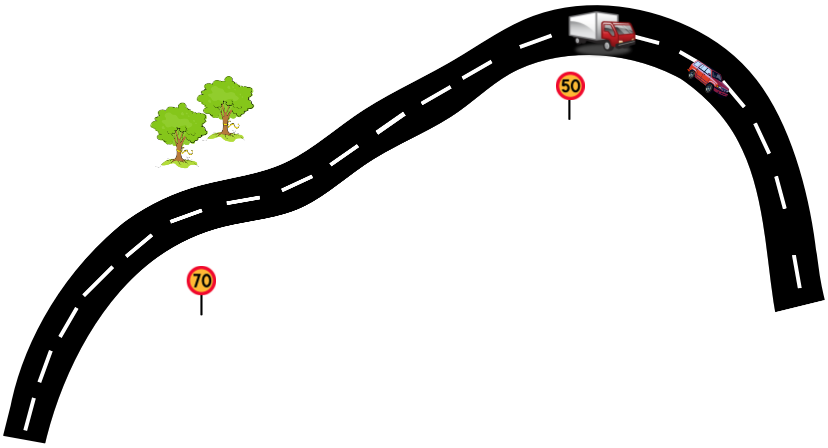


Figure 294: Predicted deceleration

In such situations, the driver will most likely release (push) the accelerator and perhaps even push the brake pedal, with the purpose of decelerating (accelerating) the vehicle. With this information at hand to the powertrain controller, the split of the power flows between the internal combustion engine (ICE) and the energy storage can potentially be adjusted, the ICE can even be turned off or on and favorable gears can be selected. These are actions with the purpose to reduce the fuel consumption and pollution.

In Figure 295 below, a speed limit due to a tight passage with speed bumps is predicted. The pathway is located in a slight downhill and hence, very little power is required for vehicle propulsion to get through the passage. In order to save fuel and decrease pollution, the diesel engine is shut off and the electric machine is used to propel the vehicle alone. After the passage the diesel engine is started up due to the higher torque demand during the acceleration.

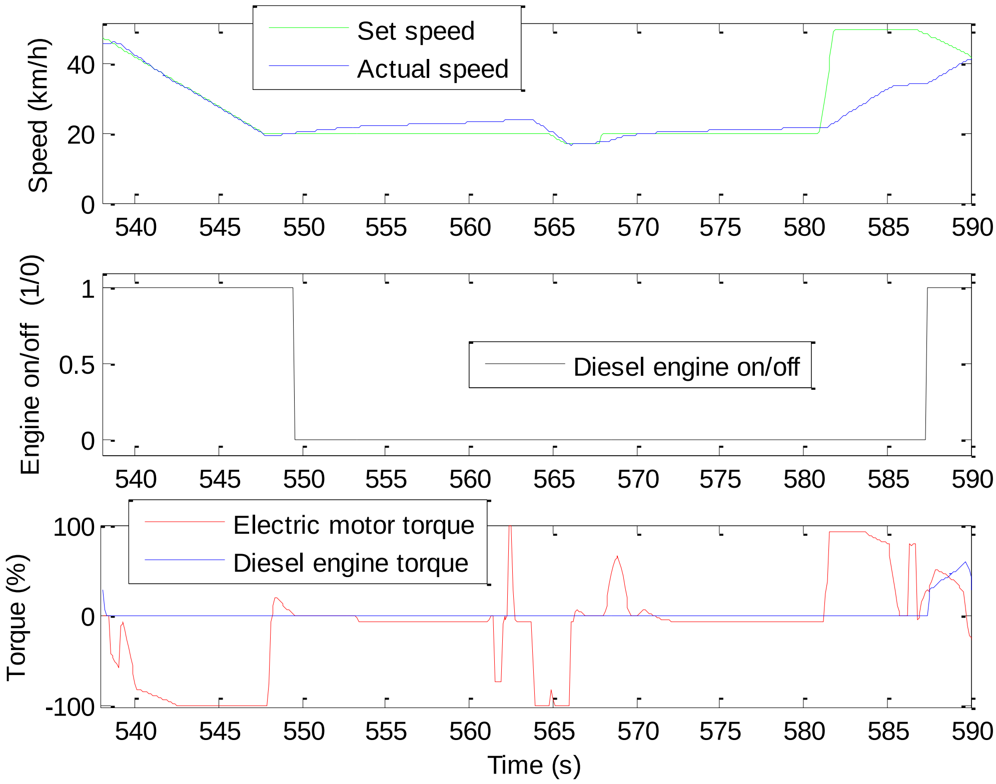


Figure 295: Predicted speed limit.

The small set speed reduction at approximately 565s in Figure 295 is due to the speed bump. Thanks to the downhill slope very little electric energy is needed for increasing the speed again after the speed bump. The major part of the speed reduction sections appears due to stationary objects/conditions such as road speed limits, roundabouts, speed bumps etc. However, in order to cover all speed reductions cases and thereby fully evaluate the potential, laser scanners with satisfactory accuracy would have been required.

Use case 3: "Predicted downhill (uphill) driving"

More or less, the road inclination varies along the vehicle route, see Figure 296. Using map-based information, it is possible to know in advance how the inclination will vary. With this information at hand to the powertrain controller, in combination with the knowledge that a constant speed is desirable, the split of the power flows between the internal combustion engine (ICE) and the energy storage can potentially be adjusted, perhaps the ICE can even be turned off (in the case when predicting downhill driving), and favorable gears can be selected. These are actions with the purpose to reduce the fuel consumption and pollution.

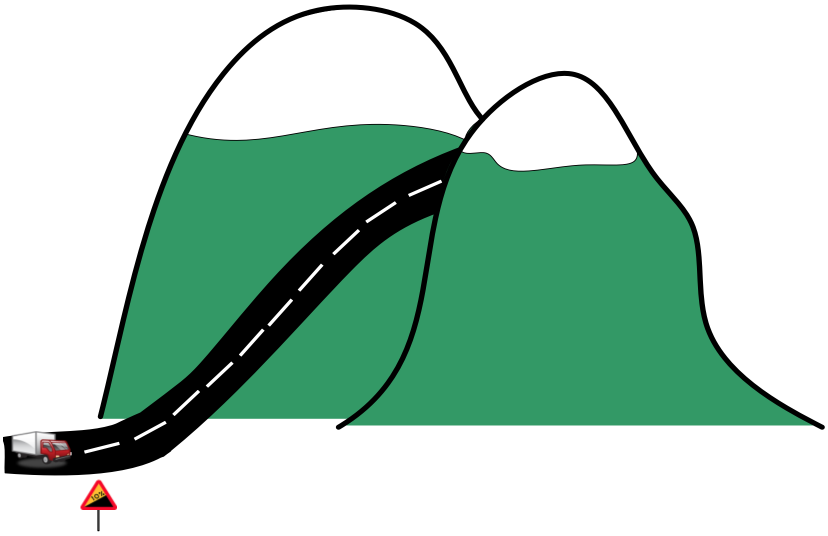


Figure 296: Uphill (downhill) driving.

Figure 297 below describes the electrical energy usage during steep uphill driving using the predicted information that the uphill is followed by a stop on top of the hill, and then followed by a downhill.

The driver is expected to use the brake pedal both at the stop on top of the hill, and then during the steep downhill driving and hence, the potential amount of recoverable brake energy is high. With the predicted information at hand the controller allows for using all available electrical energy for assisting the diesel engine during the steep uphill driving. This decreases fuel consumption and pollution, and increases drivability.

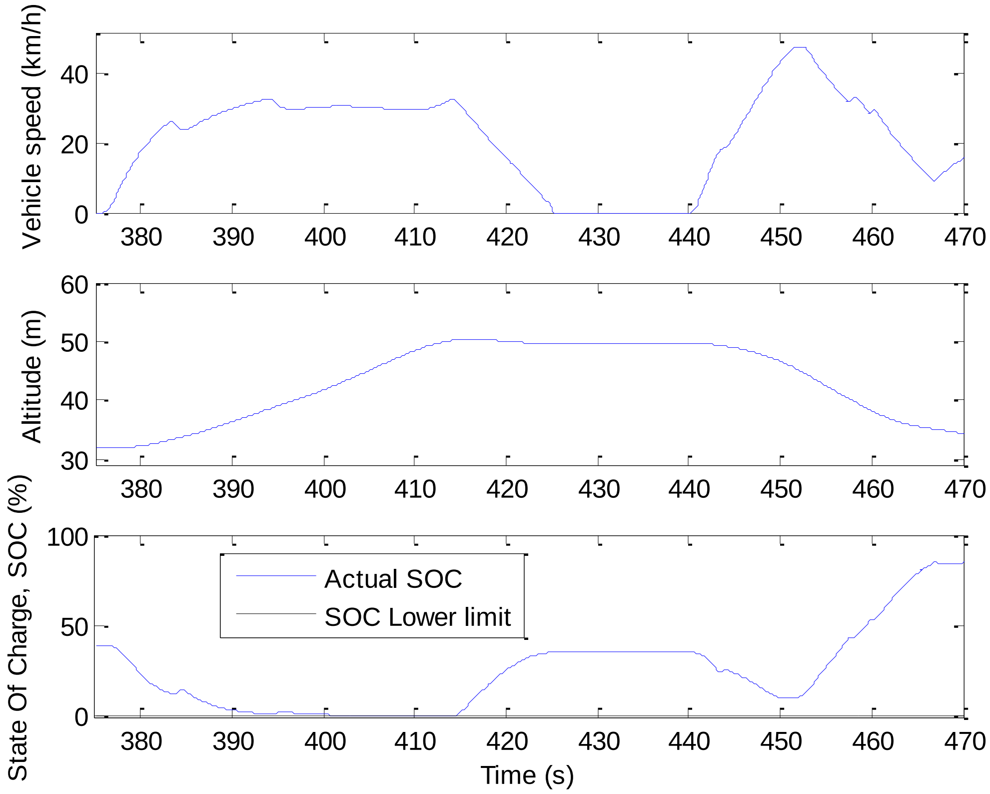


Figure 297: Predicted uphill/downhill driving.

Use case 4: "Congested driving situations"

There are certain traffic situations in which the driving cycle becomes quite repetitive. One situation is congested driving, see Figure 298. In this situation, the type of driving cycle is well defined in the sense that low speeds are most likely, where many starts and stops occur during the driving. In this case, there is a potential for the powertrain controller to reduce the fuel consumption and pollution by adjusting the split of the power flows between the internal combustion engine (ICE) and the energy storage.

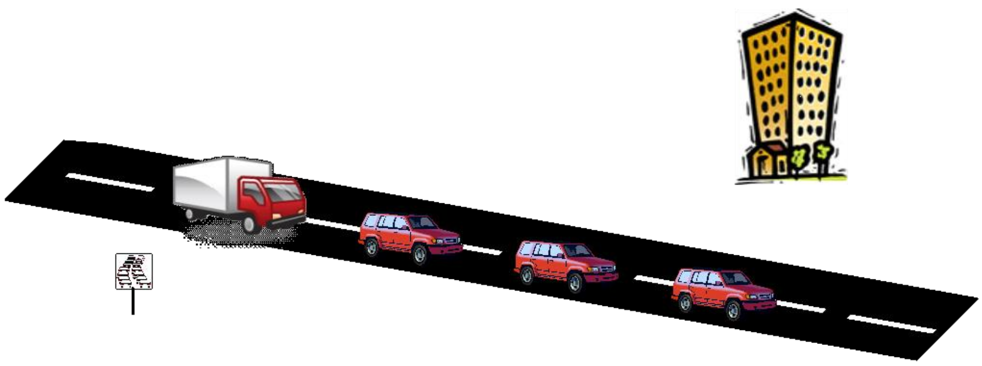


Figure 298: Congested driving situations.

In congested driving described in Figure 298, the prediction functionality is strongly depending on the information sent from the laser scanners. The information is crucial for vehicle detections that have impact on the energy management control strategy.

The prediction functionality is prepared for using laser scanner input, but the function has not yet been fully evaluated. Since slow or standstill objects have major impact on driver behavior, very high demand is put on the robustness of the sensor signals. The intention is to evaluate if the sensor information is robust enough for having a positive impact on fuel efficiency as soon as possible.

7.4.3 Summary and conclusion

The first validation of the AGD functionality (see deliverable D54.3) by use of test scenarios relied on the use of simulation. Due to technical problems with the powertrain of the demonstration vehicle, at that time it was not been possible to carry out all validation scenarios in the vehicle. However, simulators are often a good substitute for early validation of ideas and for verification of algorithms. In the meanwhile real test have taken place, which are reported in D13.1 and summarized above.

The *Speed and Load prediction*, a prerequisite for the AGD functionality of saving fuel, has had a first test in the demonstrator vehicle. The estimated speed profile horizon ahead of the vehicle is very similar to the actual speed profile when comparing the logged data files. It shows that the prediction works as intended.

The *Energy Management* controller of a hybrid vehicle can be made more fuel efficient by using information about future driving conditions. The value of additional information is however difficult to assess, since the possibility of taking advantage of the information depends on the capability of the vehicle, i.e. constraints in the transformation and storage of energy.

The implemented Energy Management control works as expected in simulation. In real world applications, information about future driving conditions needs to be estimated from e.g. map data and measurements from sensors, which will not be fully accurate. It is therefore very important that the Energy Management is robust against errors in the Speed and Load Prediction.

Some of the main challenges with developing a *Driver Coaching System* involve how to present the information to the driver to keep him motivated. The information and advice need to be intuitively understood and easy to ignore by the driver if focus is needed elsewhere. In order to gain acceptance the system also needs to manage the trade-off between fuel-economical driving and performance in such a way that it is possible to follow the recommendations of the system without unreasonably affecting the performance or traffic flow. There are Eco KPIs (Key performance Indices) that the driver can affect (e.g. speeding, hard braking) and there are aspects (possibly Eco KPIs) that the driver cannot affect, e.g. vehicle load and route planning. In order to develop a fair Eco driving functionality, e.g. for back-office use, this second type of aspect must also be taken into consideration.

The preliminary studies in the driving simulator, although not by professional drivers, show the difficulties in getting immediate acceptance, maybe since it challenges the driver’s idea of fuel economic driving. The Eco KPIs must be related to certain threshold values and the driver needs to be able to relate his performance to a threshold value. The reference value should preferably be the optimum value, the best value that a driver can score within a certain KPI. An initial set of these tuning parameters can preferably be found in the driving simulator, where the scenario is easy to repeat. The driver’s HMI, i.e. display of DCS related information and advice and the haptic pedal, has also been tested with good result, but more on-road tests are needed in order to evaluate the fuel saving potential by AGD. Validation tests undertaken indicate a fuel saving potential of 6-8% (depending on drive cycle and driving style).

8 Conclusions

This report describes the key achievements in HAVEit with reference to the objectives set at project start for each individual challenge. The presented measured system performance shows that all project goals have been achieved in all use cases and scenarios relevant for the dedicated highly automated vehicle applications.

Before HAVEit, highly automated driving was just a vague idea, after HAVEit, crisp concepts, several prototypes and first use data on highly automated driving exist. In our research and development we discovered a couple of non trivial issues (e.g. about driver drowsiness) but we proved in HAVEit that they can be controlled. So far we did not discover major road blockers, but many chances that encourage continuing with this promising path of research and development.

The HAVEit project has been very successful, which has also been the main conclusion from the final review:

- The results achieved in terms of highly automated driving based on a network of on-board sensors, were excellent. The project covered a wide range of applications. All applications rely on the general HAVEit architecture. Within the project we were able to show that this generic architecture can be implemented on state-of-the-art automotive control units. In connection with standardized interfaces between the different system blocks, and the use of standard automotive software development processes (namely AUTOSAR methodology), this helps a lot to shorten time to market. Thus, at the end of the research project HAVEit, we are in the position to transfer the results achieved into series development.
- Although the HAVEit research project has been very successful, there are still some questions that are to be further addressed in future research projects, e.g. the level of use of information received "from outside" meaning by V2V and V2I communication. Such information offers the potential to significantly increase the perception horizon of the highly automated vehicle, but raises also a couple of additional questions, e.g. who is responsible for the information, how is ensured that this information is up to date, to which extent can this information be used in the highly automated vehicle (in particular as additional information for the driver to define his global strategy or also for the definition of the optimal vehicle trajectory and control) up to the question of potential misuse. These questions were not in the focus of HAVEit. In the HAVEit architecture we included communication channels and treated the information received from outside as additional sensors. The merging of cooperative driving with highly automated driving offers strong additional potential, i.e. improved environmental modeling and the coverage of many additional use cases and scenarios, which need to be explored within the frame of additional research projects.

As a resume HAVEit opened the road for industrialization of the achievements at all levels: Starting from improved algorithms for sensors, data fusion, control etc. via modules (e.g. the Joint System) up to the highly automated vehicle applications integrating different levels of support by automation.

Acknowledgement

The research project "Highly Automated Vehicles for Intelligent Transport" (HAVEit) was executed under Framework Programme 7 of the European Commission. The Integrated Project was co-funded by the EC, ICT for Intelligent Vehicles and Mobility Services, Grant Agreement no. ICT-2007.6.1 – 212154.

The authors would like to express their thanks to the EC, Directorate Information Society and Media for granting the project. In particular, we would like to thank the whole EC team for providing valuable advice and support on all technical, contractual and financial matters throughout the entire course of the project, namely Juhani Jääskeläinen, Andre Vits, Francisco Ferreira, Dr. Irmgard Heiber, Elina Holmberg, Catherine Deckers and Laura Caniglia.

References

[1] Florian Ahlers, “Laserscanner-based lane detection and relative positioning”, 16th ITS World Congress 2009, Sept. 2009, Stockholm, Sweden

[2] Florian Ahlers, “Object Contour Tracking in Vehicle and Infrastructure Laserscanner Systems“, WIT 2010, 23-24 March 2010, Hamburg Germany

[3] Åkerstedt, T. & Gilberg, M. (1990). Subjective and objective sleepiness in the active individual. International Journal of Neuroscience, 52 (1-2), 29-37.

[4] Anderson, D., Abdalla, A., Pomletto, B., Goldberg, N. & Clement V. (2002). Distracted Driving: Review of current needs, efforts and potential strategies. Senate Document No.14, Commonwealth of Virginia, Richmond.

[5] Angermann, K. (1987). Vigilanzindikatoren bei Dauerbeobachtungsaufgaben untersucht am Beispiel nächtlicher Autobahnfahrten. Zeitschrift für Arbeitswissenschaften, 41(4), 239-242.

[6] Tom P. Alkim & Henk Schuurman, Chris M.J. Tampère, „Effects of External Cruise Control and Co-operative Following on Highways: an Analysis with the MIXIC Traffic Simulation Model“, Proceedings of the 2000 Intelligent Vehicles Conference, The Ritz-Carlton Hotel, Dearborn, MI, USA, October 3-5, 2000, pp.

[7] Bart van Arem, Member, IEEE, Cornelie J. G. van Driel, and Ruben Visser, „The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics“, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 7, NO. 4, DECEMBER 2006, pp. 429-436

[8] M. Armbruster, E. Zimmer, M. Lehmann, R. Reichel, E. Sieglin, G. Spiegelberg, A. Sulzmann,“Affordable X-By-Wire technology based on an innovative, scalable E/E platform-concept”, IEEE Spring Vehicle Technology Conference, Melbourne, 2006.

[9] M. Armbruster, E. Zimmer, M. Lehmann, R. Reichel, J. Klaubert,“ Modularisation of safety & control for X-by-Wire multiapplication-platforms”, SAE World Congress, Detroit, 2008.

[10] M. Armbruster, “Eine fahrzeugübergreifende X-by-Wire Plattform zur Ausführung umfassender Fahr- und Assistenzfunktionen”, Institute of Aircraft Systems, University of Stuttgart, PhD thesis in preparation, December 2008

[11] Automobiltechnische Zeitschrift, 01/2007, year 109, p. 58-62

[12] AUTOSAR Specification of C Implementation Rules V1.0.2 R2.1 Rev 001

[13] AUTOSAR Layered Software Architecture V2.1.0, R2.1 Rev 0014

[14] AUTOSAR List of Basic Software Modules V1.1.0 R2.1 Rev0014.

[15] AUTOSAR Technical Overview V2.1.0 R2.1 Rev 0014

[16] AUTOSAR Methodology V1.1.0 R2.1 Rev 0014

[17] AWAKE (2000). System for Effective Assessment of Driver Vigilance and Warning According to Traffic Risk Estimation - and Vehicle control in Emergency. European project IST-2000-28062 - http://www.awake-eu.org/

[18] Baghdassarians , B, V.: Robust control applied to the lateral guidance of a vehicle; Phd thesis of Evry-Val-d’Essonnes university, 2001.

[19] Bar-Shalom, Y.; Campo, L.: “The effect of the common process noise on the two-sensor fused-track covariance”, IEEE Trans. On Aerospace and Electronic Systems, Vol. AES-22, Nov. 1986, pp. 803-805.

[20] Arne Bartels, “Dynamische Bestimmung von Existenz- und Signifikanzwahrscheinlichkeiten für Umfeldobjekte”, 26th VDI/VW-Gemeinschaftstagung Fahrerassistenz und integrierte Sicherheit, 6-7 Oct. 2010, Wolfsburg, Germany

[21] Arne Bartels, Thanh-Binh To, VW, “Temporary Auto Pilot”, Journal of Automotive Technology, 2.9.2011, ATZ No. 9-2011

[22] Barton T. (2003). Smart Driving Technology vs. Fatigue. Truckers news http://www.etrucker.com/

[23] Batavia, PH. (1999). Driver-adaptive lane departure warning systems. Ph.D. Thesis. CMU-RI-TR-99-25. The Robotics Institute; Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

[24] Baulk, S. D., Reyner, L. A. & Horne, J. A. (2001). Driver sleepiness – evaluation of reaction time measurement as a secondary task. Sleep & Hypnosis, 24(6), 695-698.

[25] R. Benenson, T. Fraichard and M. Parent, "Achievable safety of driverless ground vehicles", in Proceedings of the IEEE International Conference on Control, Automation, Robotics and Vision, 2008. http://hal.inria.fr/inria-00294750

[26] R. Benenson, "Perception for driverless vehicles: design and implementation," Ph.D. dissertation, École des Mines de Paris, 2008.

[27] Benhimane, P. R. S.; Malis, E.; Azinheira, J. R.: "Vision-based control for car platooning using homography decomposition," in IEEE International Conference on Robotics and Automation, pp. 2173–2178, Barcelona, Spain, April 2005.

[28] Bertsekas D. P.: "Linear Network Optimization: Algorithms and Codes, Cambridge MA: The MIT Press, 1992.

[29] Blaauw, G. J. (1982). Driving experience and task demands in simulator and instrumented car: A validation study. Human Factors, 24, 473-486.

[30] Boadle, J. (1976). Vigilance and simulated night driving. Ergonomics, 19(2), 217-225.

[31] Boer, E.R. (2000). Behavioral Entropy as an index of workload. IEA 2000/HFES 2000 Congress, San Diego, 3, July 29 - August 4, 2000 (pp.125-128).

[32] Boussuge, J. (1995). Quinze ans de sécurité sur autoroute. Bilan et perspectives. Revue Gen. Des Routes et des Aerodromes, 726.

[33] Boverie S., D. Daurenjou D., Estève D., Poulard H., Thomas J. (2002). Driver vigilance monitoring - New developments. IFAC'02 World congress in Barcelona Spain, July 2002.

[34] Boverie, S., Giralt A. (2007). Procédé d'évaluation de l'état de vigilance d'un conducteur de véhicule. 2007-0115133SV31FR US Patent 2005P17325 US 16.11.2006 11/600075.

[35] Boverie, S., Giralt A. (2008a). Driver vigilance diagnostic based on eyelid movement observation, IFAC World congress Seoul Corée, 2008.

[36] Boverie, S., Giralt A. & Le Quellec, J.M. (2008b). Diagnostic fusion for in vehicle Driver vigilance assessment. IFAC World congress Seoul Corée, 2008.

[37] Brice, C. & Smith, A. (2001). The effects of caffeine on simulated driving, subjective alertness and sustained attention. Human Psychopharmacology: Clinical and Experimental, 16, 523-531.

[38] Brookhuis K. (1995). Integrated systems. Results of experimental tests, recommendations for introduction. Report Deter, Deliverable 18 to the European commission, University of Groningen.

[39] Brookhuis, K.A., de Waard, D. & Fairclough S.H. (2003). Criteria for driver impairment. Ergonomics, 46 (5), 433-445.

[40] Brown, I.D. (1965). A comparison of two subsidiary tasks used to measure fatigue in car drivers. Ergonomics, 8, 467-473.

[41] Brown, I.D. (1966). Effects of prolonged driving upon driving skill and performance of a subsidiary task. Industrial Medicine and Surgery, 35, 760-765.

[42] Brown, I.D., Simmonds, D.C.V. & Tickner, A.H. (1967). "Measurement of control skills, vigilance and performance of a subsidiary task during 12 hours of car driving. Ergonomics", 10, 665-673.

[43] Maria Bruce, "Optimized hybrid vehicle control using predictions of future driving environment", Reglermöte 2010, 8-9 June 2010, Lund, Sweden

[44] Maria Bruce, “Highly Automated Vehicles for intelligent transport / Active Green Driving”, Press conference at Volvo Premises, 8 July 2010, Gothenburg, Sweden

[45] Buld, S., Krüger, H.-P., Hoffmann, S., Kaussner, A., Tietze, H. & Totzke, I. (2002). Wirkungen von Assistenz und Automation auf Fahrerzustand und Fahrsicherheit, Veröffentlichter Abschlussbericht Projekt EMPHASIS: Effort-Management und Performance-Handling in sicherheitsrelevanten Situationen (Förderkennzeichen: 19 S 9812 7). Würzburg: Interdisziplinäres Zentrum für Verkehrswissenschaften an der Universität Würzburg (IZVW)

[46] Cameron, C. (1973). A theory of fatigue. Ergonomics, 16(5), 633-648.

[47] Campagne, A., Pebayle, T. & Muzet, A. (2004). Correlation between driving errors and vigilance level: influence of the driver’s age. Physiology & Behavior, 80, 515-524.

[48] CityMobil: Deliverable 3.2.2 (2008) Test Results of HMI in Use on Cars and with Simulators. Confidential document of the CityMobil project.

[49] M. Darms, M. Komar und M. Lueke: Gridbasierte Straßenrandschätzung für ein Spurhaltesystem, 6. ADAS Workshop, Löwenstein/Hößlinsülz, Sept. 2009

[50] Davies, D.R. & Tune, G.S. (1969). Human Vigilance Performance. American Elsevier Publishing Company, Inc. New York.

[51] Deram, P. (2004). Vehicle-based detection of inattentive driving for integration in an Adaptive Departure Warning System- Distraction Detection. Master Thesis. Royal Institute of Technology, Department for Signals, Sensors and Systems, Signal Processing, Stockholm.

[52] Desmond, P. A. & Matthews, G. (1997). Implications of task-induced fatigue effects for in-vehicle countermeasures to driver fatigue. Accident Analysis and Prevention, 29(4), 515-523.

[53] Dickmanns, E. D. (2002). Vision for ground vehicles: History and prospects. International Journal of Vehicle Autonomous Systems, 1(1), 1-44.

[54] Donges, E. (1978). A two-level model of driver steering behavior. Human Factors, 20, 691-707.

[55] Van Driel, Cornelie J. G. and Van Arem, Bart(2010) „The Impact of a Congestion Assistant on Traffic Flow Efficiency and Safety in Congested Traffic Caused by a Lane Drop“, Journal of Intelligent Transportation Systems, 14: 4, 197 — 208; online publication: DOI: 10.1080/15472450.2010.516226

[56] ECE Regulation No. 13, Rev. 1, Add. 12, Rev. 5: uniform provisions concerning the approval of vehicles of categories M, N, and O with regard to braking. October 2004.

[57] A. Eidehall and F. Gustafsson “Obtaining Reference Road Geometry Parameters from Recorded Sensor Data”, IEEE Intelligent Vehicles Symposium 2006

[58] Fairclough, S. H. & Graham, R. (1999). Impairment of driving performance caused by sleep deprivation or alcohol: a comparative study. Human Factors, 41(1), 118-128.

[59] Fairclough, S. H. (1997). Monitoring driver fatigue via driving performance. In Y. I. Noy (Ed.), Ergonomics and safety of intelligent driver interfaces (pp. 363-379). Mahwah, NJ: Lawrence Erlbaum Associates.

[60] Flemisch, F. (2000): The horse metaphor as a guideline for vehicle automation. Proposal to the National Research Council, Washington, USA.

[61] Flemisch, F.O., Adams, C.A., Conway, S.R., Goodrich, K.H., Palmer, M.T., and Schutte, P.C., "The H-Metaphor as a Guideline for Vehicle Automation and Interaction", Hampton: NASA, Langley Research Center, 2003.

[62] Flemisch, F.O.; Goodrich, K.H.; Conway, S.R.: At the crossroads of manually controlled and automated transport: The H-Metaphor and its first applications (progress update 2005); ITS’05; Hannover, 2005

[63] Flemisch, F.; Kelsch, J.; Schieben, A.; Schindler, J.: Stücke des Puzzles hochautomatisiertes Fahren: H-Metapher und H-Mode; 4. Workshop Fahrerassistenzsysteme; Löwenstein; 2006

[64] Flemisch, F.O., Kelsch, J., Löper, C., Schieben, A., and Schindler, J., "Automation spectrum, inner / outer compatibility and other potentially useful human factors concepts for assistance and automation", Human Factors and Ergonomics Society Europe Chapter Annual Meeting, 2007.

[65] Flemisch,F,O, Vollrath,M, Rataj,J, „Automation and assistance in trucks and cars: Diversity versus homogenization?“, 8. Symposium AAET, Braunschweig 2007

[66] Flemisch, F.O., Schindler, J., Kelsch, J., Schieben, A., and Damböck, D., "Some Bridging Methods towards a Balanced Design of Human-Machine Systems, Applied to Highly Automated Vehicles", Applied Ergonomics International Conference, 2008.

[67] Flemisch,F.O, Kelsch,J, Löper,C, Schieben,A, Schindler,J, Heesen,M, „Cooperative control and active interfaces for vehicle assistance and automation”, FISITA 2008

[68] Flemisch, F. et al.: Towards Highly Automated Driving: Intermediate report on the HAVEit-Joint System, “Transport Research Arena Europe”, Brussels, 2010

[69] Frank Flemisch, “Integrierte Sicherheit und Komfort durch eine dynamische Balance zwischen Fahrer und Assistenz/Automation: Konzept und erste Ergebnisse des HAVEit Joint Systems”, 26th VDI/VW-Gemeinschaftstagung Fahrerassistenz und integrierte Sicherheit, 6-7 Oct. 2010, Wolfsburg, Germany

[70] Fotis Karamitsos, European Commission, "EU Approach to Road Safety", 2000

[71] Fuller, R. G. (1984). Prolonged driving in convoy: the truck driver’s experience. Accident Analysis and Prevention, 16, 371-382.

[72] T. Fraichard, “A short paper about motion safety,” in Proceedings of the IEEE International Conference on Robotics and Automation, 2007. http://hal.inria.fr/inria-00134467

[73] Garder, P. (1998). Continuous Shoulder Rumble-Strips – A safety evaluation. In: CSRS Seminar, Linkoeping, August 1998.

[74] Gillberg, M., Kecklund, G. & Akerstedt, T. (1996). Sleepiness and performance of professional drivers in a truck simulator – comparisons between day and night driving. Journal of Sleep Research, 5(1), 12-15.

[75] Glaser, S; Vanholme, B; Mammar, S; Gruyer, D; Nouveliere L.: “Maneuver based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction”, IEEE Transaction on Intelligent Transportation Systems, Jun 2010

[76] Green, P., Cullinane, B., Zylstra, B. & Smith, D. (2004). Typical values for driving performance with emphasis on the standard deviation of lane position: a summary of the literature. Report SAFE-IT Task 3a (Safety Vehicles using adaptive interface technology).

[77] Griffiths, P. G. and Gillespie, R. B. (2005). Sharing control between humans and automation using haptic interface: Primary and secondary task performance benefits. Human Factors. Vol. 47, No. 3, pp.574-590.

[78] Grant Grubb, „Automatic Queue Assistance to Aid Under-loaded Drivers”, 16th ITS World Congress 2009, Sept. 2009, Stockholm, Sweden

[79] Haldex Brake Products, “FlexRay communication matrix in the HAVEit-project”, Jacob Svendenius, 2010

[80] Richard Hartley, Andrew Zisserman. Multiple view geometry in computer vision. Cambridge University Press, Cambridge, 2000

[81] Hancock, P.A., Simmons, L., Hashemi, L., Howarth, H. & Ranney, T. (1999). The effects of in-vehicle distraction on driver response during a crucial driving maneuver. Human Factors, 1 (4), 295–309.

[82] Hancock, P.A., Lesch, M. & Simmons, L. (2003). The distraction effects of phone use during a crucial driving maneuver. Accident Analysis and Prevention, 3S, SOI-SI4.

[83] Handbook of multisensory data-fusion, David L. Hall, James Llinas

[84] Hardee, L., Dingus, T.A. & Wierwille, W.W. (1985). A comparison of three subsidiary tasks used as driver drowsiness countermeasures. (IEOR Department Report #8505). Vehicle Simulation Laboratory, Human Factors Group. Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

[85] Harbluk, J.L. Noy, Y.I. & Eizenman, M. (2002). The Impact of Cognitive Distraction on Driver Behaviour and Vehicle Control. Transport Canada, Ottawa, Ontario.

[86] Harbluk, J.L. & Lalande, S. (2005). Performing e-mail tasks while driving: the impact of speech-based tasks on visual detection. In: Proceedings of the third international driving symposium on Human Factors in Driver Assessment, Training and Vehicle Design, 311-317.

[87] Hargutt, V. (2000). Eyelid movements and their predictive value of fatigue stages. 3rd International Conference of Psychophysiology in Ergonomics, San Diego, California, July 30.

[88] Hargutt, V. (2003a). Die Auswirkung von Müdigkeit auf die Fahrsicherheit (unveröffentlichte Literaturrecherche): Interdisziplinäres Zentrum für Verkehrswissenschaft an der Universität Würzburg.

[89] Hargutt, V. (2003b). Das Lidschlussverhalten als Indikator für Aufmerksamkeits- und Müdigkeitsprozesse bei Arbeitshandlungen. In VDI-Gesellschaft Fahrzeug- und Verkehrstechnik (Hrsg.), Reihe 17: Biotechnik/Medizintechnik (VDI-Fortschritt-Bericht, Nr. 233). Düsseldorf: VDI-Verlag.

[90] Harris, W.C., Hancock, P.A. & Arthur, E.J. (1993). The effect of taskload projection on automation use, performance, and workload. In Proceedings of the Seventh International Symposium on Aviation Psychology (pp. 178-184). Columbus, O.H.: The department of Aviation, The Aviation Psychology Laboratory, The Ohio State University.

[91] Harris, W.C., Goernert, P.N., Hancock, P.A. & Arthur, E. (1994). The comparative effectiveness of adaptive automation and operator initiated automation during anticipated and unanticipated taskload increase. In Mouloua & Parasuraman (Eds.), Human Performance in automated systems: Current research and trends (pp.40-44). Hillsdale, NJ: Lawrence Erlbaum Associates.

[92] Hartley, L., Horberry, T., Mabbott, N. & Krueger, P.K. (2000). Review of fatigue detection and prediction technologies. National Road Transport Commission, Melbourne, Australia.

[93] HAVEit Technical Annex (Annex I), v. 1.2, November 2007

[94] HAVEit deliverable D11.1 “Requirements“, September 2008

[95] HAVEit deliverable D11.2 “Specification”, February 2009

[96] HAVEit deliverable D12.1 “Architecture“, February 2009

[97] HAVEit deliverable D13.1 “Optimized and validated demonstration vehicles”, 2011

[98] HAVEit deliverable D21.1 “CSC & XCC HW-development complete”, July 2009

[99] HAVEit deliverable D21.2 “Software- and Configuration-Process-Concept Available”, July 2009

[100] HAVEit deliverable D21.3 “Generic platform core demonstrator available in lab”, 2010

[101] HAVEit deliverable D21.4 “Bare ECU platforms tested and ready for applications integration”, 2010

[102] HAVEit deliverable D21.5 “Platform with integrated partner applications available”, 2010

[103] HAVEit deliverable D22.1 “FlexRay and CAN communications available”, 2009

[104] HAVEit deliverable D22.2 “Communication hardware available (C2X)”, 2010

[105] HAVEit deliverable D23.1 “Brake-by-Wire”, 2010

[106] HAVEit deliverable D23.2 “Steer-by-Wire”, 2010

[107] HAVEit deliverable D31.2 “Co-driver command vector available – 2nd version”, December 2009

[108] HAVEit Deliverable D32.1 “Report on driver assessment methodology”, April 2009

[109] HAVEit Deliverable D32.2 “Model of driver behaviour for the assessment of driver’s state”, December 2009

[110] HAVEit Deliverable D33.1 (2009). Scenario Modelling and Transition Testing. Confidential document of the HAVEit project.

[111] HAVEit Deliverable D33.2 (2009). Preliminary concept on optimum task repartition for HAVEit systems. Confidential document of the HAVEit project.

[112] HAVEit Deliverable D33.3 (2009), Validation by simulation

[113] HAVEit Deliverable D33.4: “Algorithm (C-code, 1st version), available for partners”. January 2010.

[114] HAVEit Deliverable D33.5: “Algorithm (C-code, 2nd version), available for partners”. June 2010.

[115] HAVEit Deliverable D33.6 (2010), Concept on optimum task repartition in the Joint System

[116] HAVEit deliverable D41.1 “Concept and algorithm (1st version) validated in vehicle”, 2009

[117] HAVEit deliverable D41.2 “Components (including steer-by-wire) available, installed and working”, 2010

[118] HAVEit deliverable D41.3 “Joint system validation in vehicle (2nd version)”, July 2010

[119] HAVEit deliverable D42.1 “Drive-by-wire platform installed and working. Components for brake-by-wire available, installed and working”, 2010

[120] HAVEit deliverable D42.2 “EWB truck pre-homologized for open roads”, 2011

[121] HAVEit deliverable D43.1 “Transfer of HAVEit architecture to embedded platforms completed”, 2011

[122] HAVEit deliverable D43.2 “Architecture migration validation”, 2011

[123] HAVEit deliverable D51.2 “Automated Assistance in Roadworks and Congestion: Components installed, working and tested”, January 2010

[124] HAVEit deliverable D51.3 “Automated Assistance in Roadworks and Congestion: 1st System Functionality”, October 2010

[125] HAVEit deliverable D52.2 “Automated queue assistance: all components installed, working and tested”, January 2010

[126] HAVEit deliverable D52.3 “Automated queue assistance: 1st system functionality”, October 2010

[127] HAVEit deliverable D53.2 “Temporary autopilot: all components installed, working and tested”, January 2010

[128] HAVEit deliverable D53.3 “Temporary autopilot: 1st system functionality”, October 2010

[129] HAVEit deliverable D54.2 “Energy optimizing active green driving system: all components and hybrid installed, sub-systems in operation and tested”, May 2010

[130] HAVEit deliverable D54.3 “Active green driving application: 1st system functionality”, 2010

[131] HAVEit deliverable D67.1 “Report on legal issues and homologation path”, 2011

[132] Heesen, M., Schieben, A., & Flemisch, F. (2009). Unterschiedliche Automatisierungs-grade im Kraftfahrzeug: Auswirkungen auf die visuelle Aufmerksamkeit und die Kontroll-

übernahmefähigkeit,. Paper presented at the TEAP: Tagung Experimentell Arbeitender Psychologen, Jena.

[133] S. Heinrich: “Sensor-Fusion von Radar und Kamera mittels Kalmanfilter“, CTI-conference, Stuttgart, Germany, November 2005

[134] Hima, S., Lusetti, B., Vanholme, B., Glaser, S.,Mammar S., ” Trajectory Tracking for Highly Automated Passenger Vehicles”, IFAC World Congress 2011 , Milan, Italy, 30 September 2011

[135] Hoeger, R.; Amditis A. , Kunert M.; Hoess, A.; Flemisch, F.; Krueger, H.-P.; Bartels, A.; Beutner, A. (2008): Highly Automated Vehicles for Intelligent Transport: HAVEit Approach; ITS World Congress, NY, USA

[136] Hoeger, R., Amditis, A., Kunert, M., Hoess, A., Flemisch, F. O., Krueger, H.-P., Bartels, A. Beutner, A., Pagle, K. (2008). Highly Automated Vehicles for Intelligent Transport: HAVEit Approach. Paper presented at the ITS World Congress, New York.

[137] Reiner Hoeger, “Selective automated driving as a pivotal element to solve safety and environmental issues in personal mobility”, 16th ITS World Congress 2009, Sept. 2009, Stockholm, Sweden

[138] Reiner Hoeger, „Driver centric and context-sensitive automation to enhance safety and improve fuel efficiency”, 16th ITS World Congress 2009, Special Interest Session 43, Sept. 2009, Stockholm, Sweden

[139] Reiner Hoeger, „HAVEit concept and migration ideas”, METI workshop on automation, 16th ITS World Congress 2009, 25th Sept. 2009, Stockholm, Sweden

[140] Reiner Hoeger, “Driver centric automation to enhance safety and to improve fuel efficiency”, TRA conference, 7-10 June 2010, Brussels, Belgium

[141] Reiner Hoeger, Panel session: “Automated Driving: Real-World Feasibility and Benefits”, TRB Conference, 26 Jan. 2011, Washington, USA

[142] Reiner Hoeger, “Towards automation: the HAVEit approach”, ADOSE Workshop aside of IEEE Intelligent Vehicles Symposium, 05 June 2011, Baden-Baden, Germany

[143] Alfred Hoess, „Automated driving and the HAVEit approach”, EC-Japan cooperation forum on automated driving, 2nd July 2009

[144] A. Hohm, C. Wojek, B. Schiele und H. Winner: „Multi-Level Sensorfusion and Computer-Vision Algorithms within a Driver Assistance System for Avoiding Overtaking Accidents” FISITA World Automotive Congress, München, September 2008

[145] Hollnagel, E. and Woods, D. D. (2005). Joint cognitive systems: Foundations of cognitive systems engineering. Boca Raton, FL: Taylor & Francis/CRC Press.

[146] Holzmann, F. (2006). Adaptive cooperation between driver and assistant system to improve the road safety. Unpublished PhD, Swiss Federal Institute of Technology, Lausanne - Switzerland.

[147] Holzmann, F; Flemisch, F.; Siegwart, R.; Bubb, H. (2006): From Aviation down to Vehicles – Integration of a Motions-Envelope as Safety Technology: SAE 2006 Automotive Dynamics Stability and Controls Conference; Novi, Michigan, USA

[148] Holzmann, F. (2007) Adaptive Cooperation between Driver and Assistant System. Springer, Berlin

[149] Horne, J. & Reyner, L, (1999). Vehicle accidents related to sleep - a review. Occupational and Environmental Medicine, 56, 289-294.

[150] Huntley, M. S. & Centybear, T. M. (1974). Alcohol, sleep deprivation, and driving speed effects upon control use during driving. Human Factors, 16(1), 19-28.

[151] Inagaki, T. (2003). Adaptive automation: Sharing and trading of control. In E. Hollnagel (Ed.) Handbook of cognitive task design, pp. 46–89, Mahwah: Lawrence Erlbaum Publishers.

[152] Itoi, A., Cilveti, R., Voth, M.,Dantz, B., Hyde, P., Gupta, A., Dement, W.C. (1993). Can drivers avoid falling asleep at the wheel? Technical report, AAA Foundation for Traffic Safety. February, 8th.

[153] Erika Jakobsson, "Volvo to add virtual co-driver system to reduce accidents whilst in traffic", Truck & Bus Builder, Volume 31, Issue no. 8, August 2009

[154] Erika Jakobsson, "Feature: Virtual co-driver means safer trucks", Press information Volvo Trucks, Gothenburg, Sweden, 23rd July 2009

[155] Johns, M.W. (2003). The amplitude-velocity ratio of blinks: A new method for monitoring drowsiness. Sleep, 2003 (26), A51-52.

[156] Johns W., Tucker A., Chapman, R. (2005). A New Method for Monitoring the Drowsiness of the Drivers. International Conference on Fatigue Management in Transportation Operations, Seattle, USA, Sep 11-15, 2005.

[157] Armin Kaussner, "Die Bedeutung der Fahrerbeobachtung beim Fahren mit hochautomatisierten Systemen", Safety Week 2011 - die Zukunft der Fahrzeugsicherheit, Kooperationsforum „Fahrerassistenzsysteme", organized by Bayern Innovativ, 26 May 2011, Aschaffenburg, Germany

[158] Khardi, S. & Vallet, M. (1994). Drowsiness of the driver: EEG and vehicle parameter interaction (paper nr. 94 S3 O 06). Paper presented at the 14th International Conference on Enhanced Safety Vehicles (ESV), Munich; May 1994.

[159] Kelsch, J.; Flemisch, F.; Löper, C.; Schieben, A.; Schindler, J., 2006, Links oder rechts, schneller oder langsamer? Grundlegende Fragestellungen beim Cognitive Systems Engineering von hochautomatisierter Fahrzeugführung. DGLR Fachauschusssitzung Anthropotechnik, Karlsruhe, Germany

[160] Kelsch, J., Heesen, M., Löper, C., & Flemisch, F. (2009). Balancierte Gestaltung kooperativer multimodaler Bedienkonzepte für Fahrerassistenz und Automation: H-mode beim Annähern, Notbremsen, Ausweichen. Paper presented at the Der Mensch im Mittelpunkt technischer Systeme 8. Berliner Werkstatt Mensch-Maschine-Systeme, Berlin.

[161] King, D. J., Mumford, D. K. & Siegmund, G. P. (1998). An algorithm for detecting heavy-truck driver fatigue from steering wheel motion (paper nr. 98-S4-O-10). Paper presented at the 16th International Technical Conference on the Enhanced Safety of Vehicles, Washington DC.

[162] Kircher, A., Vogel, K., Törnros, J., Bolling, A., Nilsson, L., Patten, C., Malmstrom, T. & Ceci, R. (2003). Mobile Telephone Simulator Study. Swedish National Road and Transport Research Institute (VTI), Linkoping, Sweden.

[163] Kircher, K. (2007). Driver distraction: A review of the literature (VTI Report No. 594A). Linköping, Sweden: VTI (Swedish National Road and Transport Research Institute).

[164] Knappe, G., Keinath, A. & Meinecke, Ch. (2006). Empfehlungen für die Bestimmung der Spurhaltegüte im Kontext der Fahrsimulation. MMI-Interaktiv, 11. Available at www.useworld.net/ausgaben/12-2006/02-Knappe_et_al.pdf [06.06.2007].

[165] Knipling, R.R. & Wierwille, W.W. (1994). Vehicle-based drowsy driver detection: Current status and future prospects. Paper presented at the IVHS America Fourth Annual Meeting (April 17-20). Atlanta: Georgia.

[166] Kraiss, K.-F. (1994). 99% Langeweile und 1% panische Angst" – über die Schwierigkeiten beim Umgang mit hochautomatisierten Systemen. In M. Kerner (Ed.). Technik und Angst. Zur Zukunft der industriellen Zivilisation. Aachen: Verlag der Augustinus Bhg.

[167] Kuhn, F. (2005). Methode zur Bewertung der Fahrerablenkung durch Fahrerinformationssysteme. Presentation on the World Usability Day 2005, 3.11.2005 in Stuttgart. Available online: www.gui-design.de/download/wud_LCT_2005-11_Stuttgart.pdf.

[168] Lemke, M. (1980). Monotoniebelastung von Kraftfahrern. Automobil-Industrie, 3, 77-85.

[169] Lenné, M. G., Triggs, T. J. & Redman, J. R. (1997). Time of day variations in driving performance. Accident Analysis and Prevention, 29(4), 431-437.

[170] Lenné, M. G., Triggs, T. J. & Redman, J. R. (1998). Interactive effects of sleep deprivation, time of day, and driving experience on a driving task. Sleep, 21(1), 38-44.

[171] Löper, C., Kelsch, J., and Flemisch, F.O., "Kooperative, manöverbasierte Automation und Arbitrierung als Bausteine für hochautomatisiertes Fahren", Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel, S. 215–237, 2008.

[172] Löper, Christian; Flemisch, Frank O.: "Ein Baustein für hochautomatisiertes Fahren: Kooperative, manöverbasierte Automation in den Projekten H-Mode und HAVEit. In: Stiller, Christoph; Maurer, Markus (Hg.): 6. Workshop Fahrerassistenzsysteme. Karlsruhe: Freundeskreis Mess- und Regelungstechnik Karlsruhe e.V., S. 136–146., 2009.

[173] S. Lüke, M. Komar und M. Strauss: Reduced Stopping Distance by Radar-Vision Fusion, contribution to congress Advanced Microsystems for Automotive Applications, Springer Berlin Heidelberg, Berlin, Mai 2007

[174] S. Lüke, M. Wintermantel, T. Raste und P. Rieth: Ein neuartiges Fernbereichsradar und dessen Anwendung zur Lenkunterstützung in Ausweichsituationen, Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel AAET, Braunschweig, 2009

[175] S.Lüke, M.Darms, M.Komar, D.Waldbauer: Ein Assistenzsystem zur Unterstützung des Fahrers in Baustellen. 11. Braunschweiger Symposium AAET 2010.

[176] Mackie, R.R. & Miller, J.C. (1978). Effects of hours of service, regularity of schedules and cargo loading on truck and bus driver fatigue (Technical Report 1765-F). Goleta, CA: Human Factors Research Inc.

[177] Mammar, S.: Assisted and automated lateral control of vehicles: Robust control approach; Habilitation thesis of Evry-Val-d'Essonnes University, 2001.

[178] Markkula, G., Kutila, M., Engström, J., Victor, T.W. & Larsson, P. (2005). Online detection of driver distraction- preliminary results from the AIDE Project. In: Proceedings of the 2005 International Truck and Bus safety and Security Symposium (pp. 86-96). Washington, Alexandria, Virginia, USA, 4-16. 11. 2005.

[179] Mast, T.M., Jones, H.V. & Heimstra, N.W. (1966). Effects on fatigue on performance in a driving device. Highway Research Record, 122, 93 (abridgement), Cited in Haworth, N.L., Vulcan, P., Triggs, T.J. & Fildes, B.N. (1989). Driver fatigue research: Development of methodology. Accident Research Center, Monash University, Australia.

[180] Matthews, G. & Desmond, P. A. (2002). Task-induced fatigue states and simulated driving performance. The Quarterly Journal of Experimental Psychology, 55A(2), 659-686.

[181] Merat, N., & Jamson, H. (2008). Is drivers' situation awareness influenced by a fully automated driving scenario? Paper presented at the Human Factors and Ergonomics Society Europe Chapter conference, Soesterberg, The Netherlands.

[182] Micronas, "Product information HAL 2810, Linear Hall-Effect with LIN Bus", 2008

[183] Miller, C. A., & Parasuraman, R. (2007). Designing for Flexible Interaction Between Humans and Automation: Delegation Interfaces for Supervisory Control. Human Factors, 49(1), 57-75.

[184] Mills and Hobbs; Mills, P.J.; Hobbs, C.A.: "The probability of injury to car occupants in frontal and side impacts". In: Stapp Car Crash Conf, Chicago, IL, pp. 223–232, 1984

[185] MISRA-C:2004 Guidelines for the use of the C language in critical systems, October 2004

[186] M. Mitschke, "Dynamik der Kraftfahrzeuge", Springer-Verlag 2004; ISBN 3-540-42011-8

[187] Bruce Morey: Continental integrates driver assistance, Automotive Engineering International (AEI) Online, Magazine, September 2007

[188] Henning Mosebach, “Ein Steer-by-Wire-System für hochautomatisierte PKW”, VDI – Mechatronik 2011, 31March 2011, Aachen, Germany

[189] Multi-Sensor Data Fusion: An Introduction, H.B. Mitchell

[190] Muzet A., Penzel T., Ktonas P., Virkkala T., Anderer P., Dorffner G. (2006). List of needs and suggestions for novel sensors and improved methods for physiological data recording. Deliverable 1.2.2; Sensation EU project n°507231.

[191] National Transportation and Safety Board of US (1999). Department of Transportation efforts in the 1990s to address operator fatigue, Safety Report NTSB/SR-99/01.

[192] Neukum, A., Krüger, H.-P. & Schuller, J. (2001). Der Fahrer als Messinstrument für fahrdynamische Eigenschaften? VDI-Bericht 1613: Der Fahrer im 21. Jahrhundert. Düsseldorf: VDI-Verlag.

[193] OMG Object Management Group: Software Process Engineering Metamodel Specification, http://www.omg.org/technology/documents/modeling_spec_catalog.htm

[194] Oron-Gilad, T., Ronen, A. & Shinar, D. (2008). Alertness maintaining tasks (ATMs) while driving. Accident Analysis and Prevention, 40, 851-860.

[195] Papadimitriou, I.; Tomizuka, M.: "Fast lane changing computations using polynomials", American Control Conference, 4-6 June 2003.

[196] Peters, B., Kloeppel, Alicandri, E., Fox, J. E., Thomas, M. L., Thorne, D. R., Sing, H.C & Balwinski, S.M. (1999). Effects of partial and total sleep deprivation on driving performance. Public Roads, 62 (4).

[197] Petit, C., Chaput, D., Tarriere, C., Le Coz, J.Y. & Planque, S. (1990). Research to prevent the driver from falling asleep behind the wheel. Proceedings of the 34th Annual Conference of the Association for the Advancement of Automotive Medicine. Arizona, USA.

[198] Petti, S.: “Safe navigation within dynamic environments: a partial motion planning approach,” Ph.D. dissertation, Ecole des Mines de Paris, 2007.

[199] Stefan Pettersson, “Hybrid Vehicles for a sustainable mobility”, 16th ITS World Congress 2009, Stockholm, Sweden

[200] Pohl, J., Birk, W. & Westervall, L. (2007). A driver-distraction-based lane-keeping assistance system. Proceedings of IMechE, Vol. 221, Part I: J. Systems and Control Engineering, 541-552.

[201] A. Polychronopoulos, M. Tsogas, A. Amditis, A. Etemad, “Extended path prediction using camera and map data for lane keeping support’ ’IEEE Conference on Intelligent Transportation Systems, 13-16 September, 2005, Vienna, Austria

[202] PReVENT / ProFusion2 Deliverable, D15.12: Final report Su-Birm Park, Delphi, 31.12.2007

[203] PReVENT / INSAFES Deliverable, D60.10 Final Report, N. Floudas, 31.01.2007

[204] PReVENT / LATERAL-SAFE Deliverable, D32.11: Final report, ICCS, N. Floudas, 31.01.2007

[205] Rattaphol Pueboobpaphan and Bart van Arem, „Driver and Vehicle Characteristics and Platoon and Traffic Flow Stability - Understanding the Relationship for Design and Assessment of Cooperative Adaptive Cruise Control“, Transportation Research Record: Journal of the Transportation Research Board, No. 2189, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 89–97. Online publication: DOI: 10.3141/2189-10

[206] Raharijaona, T.: Robust control for assistance to lateral control of road vehicle; Phd thesis of Paris XI Orsay University, 2004.

[207] J. Rataj, "Beyond driver assistance - the DLR project Assistance for Traffic Participants, FISITA 2008, F2008-2-035, Sept. 2008

[208] Rauch, N., Totzke, I. & Krüger, H.-P. (2004). Kompetenzerwerb für Fahrerinformationssysteme: Bedeutung von Bedienkontext und Menüstruktur. In VDI Gesellschaft Fahrzeug- und Verkehrstechnik (Hrsg.), Integrierte Sicherheit und Fahrerassistenzsysteme, VDI-Berichte 1864 (S. 303-322). Düsseldorf: VDI-Verlag.

[209] Rauch, N., Schoch, S. & Krüger H.-P. (2007). Ermittlung von Faheraufmerksamkeit aus Fahrverhalten. Unpublished report within the BMWi Project AKTIV-AS, subproject FSA. Würzburg: WIVW GmbH

[210] Rauch, N., Kaussner, A., Krüger, H-P., Boverie, S. and Flemisch, F. (2009) The importance of driver state assessment within highly automated vehicles. ITS World Congress, September 2009, Stockholm.

[211] Rauch, N., Kaussner A., Krüger, H.-P., Boverie, S. & Flemisch, F. (2010). Measures and Countermeasures for impaired driver's state within highly automated driving. Paper presented at the Transport Research Arena Europe 2010 (TRA2010), Brussels, Belgium, 07.-10. June 2010.

[212] Renner, G & Mehring, S. (1997). Lane departure and drowsiness- two major accident causes-one safety system. Transport Research Laboratory, TRL Report 220, 1997.

[213] Paulo Resende, "Real-time Dynamic Trajectory Planning for Highly Automated Driving on Highways", 13th International IEEE Conference on Intelligent Transportation Systems, 19-22 Sept. 2010, Madeira Island, Portugal

[214] RESPONSE3 Deliverable D11.2 (2006). Code of practice for the design and evaluation of ADAS. Public document of the PReVENT project.

[215] Riemersma, J.B.J., Sanders, A.F., Wildervanck, C. & Gaillard, A.W. (1977). Performance decrement during prolonged night driving. In R.R. Mackie (Ed.), Vigilance; Theory, overational performance and physiological correlates (pp. 41-58). New York: Plenum.

[216] Rogé, C., Pébayle, T., Kiehn, L. & Muzet, A. (2002). Alteration of the useful visual field as a function of state of vigilance in simulated car driving. Transportation Research Part F, 5, 189-200.

[217] Rouse, W. B. (1976). Adaptive allocation of decision making responsibility between supervisor and computer. In T. B. Sheridan and G. Johannsen (Eds.). Monitoring behavior and supervisory control. New York: Plenum Press.

[218] Rouse, W. B. (1977). Human-computer interaction in multi-task situations. IEEE Transaction on Systems, Man, and Cybernetics, SMC-7, 384-392.

[219] Rouse, W. B. (1991) Design for success: A human-centered approach to designing successful products and systems. New York: Wiley.

[220] Safford, R. & Rockwell, T.H. (1967). Performance decrements in twenty-four hour driving. Highway research record, 163, 68-79.

[221] Sarter, N. B. and Woods, D. D. (1995). "How in the world did we ever get into that mode?" Mode Error and Awareness in Supervisory Control. Human Factors 37(1), pp. 5-19.

[222] SAVE (1996). System for effective Assessment of the driver state and Vehicle control in Emergency. European project TR1047. Available at: http://www.iao.fhg.de/Projects/SAVE

[223] Govert Schermers, Kerry M Malone, Bart van Arem, „DUTCH EVALUATION OF CHAUFFEUR ASSISTANT - Traffic flow effects of implementation in the heavy goods vehicle sector", ITS WORLD CONGRESS NAGOYA, AICHI 2004,

[224] Schieben, A., Damböck, D., Kelsch, J., Rausch, H., and Flemisch, F.O., "Haptisches Feedback im Spektrum von Fahrerassistenz und Automation", Aktive Sicherheit durch Fahrerassistenz, 2008.

[225] Schieben, A., & Flemisch, F. O. (2008). Who is in control? Exploration of transitions of control between driver and an eLane vehicle automation. In V. Wissensforum (Ed.),

Integrierte Sicherheit und Fahrerassistenzsysteme (pp. 455-469). Wolfsburg: VDI-Verlag.

[226] Schieben, A., Heesen, M., Schindler, J., Kelsch, J., & Flemisch, F. (2009). The theater-system technique: Agile designing and testing of system behavior and interaction, applied to highly automated vehicles. Paper presented at the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Essen, Germany.

[227] Anna Schieben, "Was passiert, wenn ich Gas gebe? Untersuchung und Gestaltungs-hinweise zu Fahrer initiierten Transitionen in einem hochautomatisierten Fahrzeug für das EU-Projekt HAVEit", VDI Tagung, Fahrer im 21. Jahrhundert, 8-9 Nov. 2011, DLR, Braunschweig, Germany

[228] Schomerus, J., Flemisch, F. O., Kelsch, J., Schieben, A., and Schmuntzsch, U. (2006). Erwartungsbasierte Gestaltung mit der Theatersystem- / Wizard-of-Oz-Technik am Beispiel eines haptischen Assistenzsystems. Proceedings of the AAET 2006, Braunschweig

[229] Nadja Schoemig, "Measures and countermeasures for impared driver's state within highly automated driving", TRA conference, 7-10 June 2010, Brussels, Belgium

[230] Andreas Weiser, "A probabilistic lane change prediction module for highly automated driving", WIT 2010, 23-24 March 2010, Hamburg Germany

[231] Sciarretta, A. Back, M. Guzzella, L. 'Control Strategy of Parallel Hybrid Electric Vehicle', IEEE Transactions on control systems technology, Vol. 12 No. 3, may 2004, p 352-363

[232] SENSATION (2004). Advanced Sensor Development for Attention Stress Vigilance and Sleep/Wakefulness Monitoring. European project IST-507231 - www.sensation-eu.org.

[233] Y. B. Shalom, X.-R. Li, T. Kirubarajan, "Estimation with Applications to Tracking and Navigation", John Wiley & Sons,2001

[234] Shamir, T.: "How should an autonomous vehicle overtake a slower moving vehicle: design and analysis of an optimal trajectory", IEEE Transactions on Automatic Control, April 2004.

[235] Sheridan, T.B. and Parasuraman, R. (2006). Human-automation interaction. In R. S. Nickerson (Ed.), Reviews of Human Factors and Ergonomics, vol. 1. Human Factors and Ergonomics Society, Santa Monica, CA. pp. 89-129.

[236] Slavnich D. (2009). Safety Drive showcasing the advances in Driver assistance Systems, Traffic Technology International journal, 22.

[237] Spiegelberg; G. (2002) Ein Beitrag zur Erhöhung der Verkehrssicherheit und Funktionalität von Fahrzeugen unter Einbindung des Antriebstrangmoduls MOTionX-ActR., University Karlsruhe, Dissertation.

[238] Staubach, M. (2009). Identifikation menschlicher Einflüsse auf Verkehrsunfälle als Grundlage zur Beurteilung von Fahrerassistenzsystem Potentialen. Unpublished Dissertation, Technische Universität Dresden.

[239] S. Steinmeyer, "Handling contradictory sensor data in environment maps for advanced driver assistance systems", 4^{th} German Workshop Sensor Data Fusion, Lübeck, 2009

[240] Stephan, K., Hosking, S., Regan, M., Verdoorn, A., Young, K. & Haworth, N. (2006). The relationship between driving performance and the Johns Dorwsiness Scale as measured by the Optalert System (No. 252). Clayton, Victoria: Monash University Accident Research Center.

[241] Matthias Strauss, Frank Flemisch, "Designing human computer interfaces for Highly Automated Vehicles: Issues under consideration in the EU-project HAVEit", HCI Congress 2011, 9-14 July 2011, Orlando, Florida, USA

[242] Stutts, J. et al. (2003). Distractions in everyday driving. AAA Foundation for Traffic Safety, Washington, DC.

[243] Stutts, J. C., Reinfurt, D. W., Staplin, L. & Rodgman, E. A. (2001). The role of driver distraction in traffic crashes. Available at: http://www.aaafoundation.org/pdf/distraction.pdf (09.07.2007).

[244] The 100-Car Naturalistic Driving Study, Phase II - Results of the 100 Car Field Experiment, Report No. DOT HS 810 593, April 2006

[245] Thiffault, P. & Bergeron, J. (2003). Monotony of road environment and driver fatigue: a simulator study. Accident Analysis and Prevention, 35(3), 381-391.

[246] Tränkle, U. (1980). Ermüdung und Monotonie bei Autobahnfahrten. Zeitschrift für Verkehrssicherheit, 26(4), 166-172.

[247] Thrun, S., Montemerlo, M. et al. (2006). Stanley: The Robot that Won the DARPA Grand Challenge. Journal of Field Robotics 23(9): 661-692.

[248] S. Thrun, W. Burgard, und D. Fox: "Probabilistic robotics", Intelligent robotics and autonomous agents, MIT Press, Cambridge, Mass. [u.a.], 2005

[249] Toffetti, A., Wilschut, E., Martens, M., Schieben, A., Rambaldini, A., Merat, N., & Flemisch, F. (2009). CityMobil: Human Factors issues regarding highly-automated vehicles on an eLane. Paper presented at the Transportation Research Board, New York.

[250] M. Tsogas, A. Polychronopoulos,and A. Amditis.. "Unscented Kalman Filter Design for Automotive Safety Applications" In Proc. 8th International Conference on Information Fusion,pages 1295–1302.IEEE, 2005

[251] Technical Report, Approval Report No: EB177.0E according to ECE - Regulation No. 13/11, TÜV NORD Mobilität GmbH & Co. KG, IFM – Institute for Vehicle Technology and Mobility, Adlerstraße 7, 45307 Essen, Germany

[252] Van der Hulst, M., Meijman, T. & Rothengatter, T. (2001). Maintaining task set under fatigue: A study of time-on-task effects in simulated driving. Transportation Research Part F, 4, 103-118.

[253] Vanholme, B.; Glaser, S; Mammar, S, Gruyer, D.: "Manoeuvre based trajectory planning for highly autonomous vehicles on real road with traffic", In Proceedings of ECC'09, Budapest, Hungary, 23-26 August 2009.

[254] Vanholme, B.; Glaser, S; Mammar, S, Member, IEEE, Gruyer, D.: "Manoeuvre based trajectory planning for highly autonomous vehicles on real road with traffic", In Proceedings of ECC'09, Budapest, Hungary, 23-26 August 2009.

[255] Vanholme, B; Gruyer, D; Glaser, S; Mammar, S:"Fast prototyping of a Highly Autonomous Cooperative Driving System for public roads", In Proceedings of IV'10, San Diego, USA, 21-24 June 2010

[256] Vanholme, B; Gruyer, D; Glaser, S; Mammar, S:" A Legal Safety Concept for Highly Automated Driving on Highways", In Proceedings of IV'11, Baden Baden, Germany, 5-9 June 2011

[257] Verwey, W. B. & Zaidel, D. M. (1999). Preventing drowsiness accidents by an alertness maintenance device. Accident Analysis and Prevention, 31, 199-211.

[258] Verwey, W. B. & Zaidel, D. M. (2000). Predicting drowsiness accidents from personal attributes, eye blinks and ongoing driving behaviour. Personality and Individual Differences, 28, 123-142.

[259] Victoria Transport Policy Institute, "Transportation Cost and Benefit Analysis", chapter 5.3 "Safety and Health Impacts", May 17, 2007

[260] Volvo Trucks "Active wheel speed sensor specification" report no. 26331-256, Erik Carresjö, 2004

[261] Volvo 3P "Requirements on SAE J1939 messages for Haldex EBS" report no. 26331-252, Jan-Inge Svensson, 2004

[262] De Waard, D. & Brookhuis, K. A. (1991). Assessing driver status: A demonstration experiment on the road. Accident Analysis and Prevention, 23(4), 297-307.

[263] De Waard, D. (1996). The measurement of drivers' mental workload. PhD thesis, University of Groningen. Haren, The Netherlands: University of Groningen, Taffic Research Centre.

[264] De Waard, D., Steyvers, F.J.J.M., & Brookhuis, K.A. (2004). How much visual road information is needed to drive safely and comfortably? Safety Science, 42, 639-655.

[265] Wang, J. S., Knipling, R. R. & Goodman, R. J. (1996). The role of driver inattention in crashes: new statistics from the 1995 crashworthiness data system, 40th Annual Proceedings of the Association for the Advancement of Automotive Medicine (pp. 377-392). Vancouver, Canada.

[266] A. Weiser, "A probabilistic lane change prediction module for highly automated driving", 7th International Workshop on Intelligent Transportation, Hamburg, 2010

[267] White, Jr., F.E., Data Fusion Lexicon, Joint Directors of Laboratories, Technical Panel for C3, Data, Fusion Sub-Panel, Naval Ocean Systems Center, San Diego, 1987

[268] Wierwille, W.W., Ellsworth, L.A., Wreggit, S.S., Fairbanks, R.J. & Kim, C.L. (1994). Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness (DOT HS 808 247). Blacksburg, Virginia: Office of Crash Avoidance Research, National Highway Traffic Safety Administration.

[269] Wierwille W. W. (1999). Historical perspective on slow eyelid closure: When PERCLOS? Technical Conference on Ocular Measures of Driver Alertness, Herndon.

[270] Winner, H. and Hakuli, S. (2006). Conduct-by-Wire: Following a new paradigm for driving into the future. Proceedings of FISITA 2006 World Automotive Congress. Octobre 2006, Japan.

[271] I.R. Wilmink, G.A. Klunder, and B. van Arem, „Traffic flow effects of Integrated full-Range Speed Assistance (IRSA)", Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, June 13-15, 2007, pp. 1204-1210

[272] Winner, H., Hakuli, S., Bruder, R., Konigorski, U., & Schiele, B. (2006). Conduct-by-Wire - ein neues Paradigma für die Weiterentwicklung der Fahrerassistenz. Paper presented at the Workshop Fahrerassistenzsysteme 2006, Löwenstein.

[273] Saul Wordsworth, „Drive by Wire", Traffic Technology International, UK, January 2010, pp. 33-35

[274] Wreggit, S.S., Kirn, C.L. & Wierwille, W.W. (1993). Research on vehicle-based driver status/performance monitoring (ISE Department Report 93-06). Vehicle Analysis and Simulation Laboratory, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

[275] Van Winsum, W., Brookhuis, K.A. & de Waard, D. (2000). A comparison of different ways to approximate Time-to-Line-Crossing (TLC) during car driving. Accident Analysis and Prevention, 32 (1), 47-56.

[276] Yabuta, K., Iizuka, H., Yanagishima, T., Kataoka, Y. & Seno, T. (1986). The development of drowsiness warning devices. Paper presented at the 10th International Technical Conference on Experimental Safety Vehicles, Oxford.

[277] Zeng Holger, "Tomorrow's vehicle has a co-driver: Continental is coordinating HAVEit", Conti Intern Automotive online, 1/2010

[278] Holger Zeng, "HAVEit: A driver centric vehicle automation system with a scalable and flexible architecture", Aachener Kolloquium 2010, 4-6 Oct. 2010, Aachen, Germany

[279] Holger Zeng, "HAVEit – Highly Automated Vehicles for Intelligent Transport", VEJ Forum 2010, 8-9 Dec. 2010, Nyborg, Denmark

[280] Zhou, K.; Doyle J. C.: Essentials of robust control; Prentice-Hall; 1998.

Keywords

ABS	Anti Blocking System
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistant System
ADC	Analog Digital Converter
AGD	Active Green Driving
AM	Attention Monitor
API	Application Programming Interface
AQuA	Automated Queue Asssitance
ARC	Automated assistance in Roadworks and Congestion
ASIC	Application Specific Integrated Circuit
ASIL	Automotive Safety Integrity Level
AUTOSAR	AUTomotive Open System Architecture
CALM	Continuous Air-interface for Long and Medium range
CAN	Controller Area Network
CMOS	Complementary Metal Oxide Semiconductor
CPC	Central Platform Core
CSC	Chassis and Safety Controller
DA	Driver Assisted
DBC	DataBase CAN
DCS	Driver Coaching System
DHCP	Dynamic Host Configuation Protocol
DM	Driver Monitoring
DMA	Direct Memory Access
DMS	Driver Monitoring System
DDM	Driver Drowsiness Monitoring
DIM	Driver Inattention Monitoring
DO	Driver only
DSA	Driver State Assessment
EBA	Electronic brake actuator
EBS	Electronic Braking System
ECMS	Equivalent Fuel Consumption Minimization Strategy
ECU	Electronic Control Unit
EDP	Enhanced Data Port
ECG	Electrocardiography
EEG	Electroencephalography
EM	Electrical Machine (high voltage)
EMB	Electro Mechanical Brake
ESP	Electronic Stability Control
FBD	Functional Block Diagram
FPS	Functional Performance Specification
HA	Highly Automated
HA-1	Highly Automated: manoeuvre-based (used in VW study)
HA-2	Highly Automated: higher automation (used in VW study)

HA-L	Highly Automated Light (used in VW study)
HA-S	Highly Automated Strong (used in VW study)
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
ICE	Internal Combustion Engine
IP	Internet Protocol
IPV6	Internet Protocol version 6
IR	Infrared
ISO	International Standardization Organization
ITS	Intelligent Transport Systems
KPI	Key Performance Indices
KSS	Karolinska Sleepiness Scale
LKAS	Lane Keeping Assistance System
MAP	Mega-application (in XCCs)
MRM	Minimum Risk Manouvre
MSU	Mode Selection and Arbitration Unit
NEMO	Network Mobility
NIR	Near Infrared
OLSR	Optimised Link State Routing
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PDM	Power Distribution Module
PERCLOS	Percent Eye Closure
RTE	Runtime Environment
SADT	Structured Analysis and Design Techniques
SCI	Serial Control Interface
SCM	System Control Module
SD	Standard deviation
SF, SDF	Sensor Fusion, Sensor Data Fusion
SDLP	Standard deviation of lateral Position
SoC	State of Charge (of the high voltage battery)
SPI	Serial Peripheral Interface
SW-C	Software Component
SWR	Steering Wheel Reversal
SWRR	Steering Wheel Reversal Rate
S&T	Scientific and Technological
TAP	Temporary Auto-Pilot
TCS	Traction Control System
TDMA	Time Division Multiple Access
TLC	Time-to-Line-Crossing
TOR	take-over-request
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus

V2V	Vehicle to Vehicle Communication (also known as C2C)
V2I	Vehicle to Infrastructure Communication
VWSCF	Vehicle Web Service Communication Framework
WiFi	Wireless Fidelity
XCC	X-by-wire Control Computer
XML	Extensible Markup Language