The Vehicle as Mobile Sensor
in a Collaborative Network

Dissertation

accepted in fulfilment of the requirements

for the degree of Doctor of natural sciences (Dr. rer. nat.)

Department of Mathematics and Computer Sciences

FernUniversität in Hagen

Clemens Dannheim

Place of birth: Lünen, Germany

Hagen 2015

Date submitted: 10.04.2015

Date of the examination: 02.06.2015

First examiner: Prof. Dr. Christian Icking, FernUniversität Hagen, Germany

Second examiner: Prof. Dr. Philip Sallis, Auckland University of Technology, New Zealand

Abstract

The global increase in vehicle numbers and resulting traffic flow, especially in urban areas, presents a growing challenge to governments and local authorities in terms of road infrastructure and journey logistics. A growing awareness of the harmful effects of vehicle emissions and some social issues in terms of time spent travelling and in vehicles, especially during heavy periods of traffic, has in recent times also led to substantial resources being allocated to resolving problems and improving the negative aspects of these situations. The will to reduce air pollution and road accidents while also improving fuel economy is at the heart of the issues motivating studies looking into technology and changes in driver behaviour.

The thesis first discusses some previous research that has sought robust solutions to these matters. The concepts described in this thesis are concerned with on-board vehicular sensors that receive climate and weather data in real-time, such that the control system of the vehicle is informed by and can process this data rapidly, with consequent operating behaviour modification. Coming from external sources such as on-board sensors and regional weather reports, the data is blended with other vehicle sensor data to provide a nascent ‘intelligent’ vehicle. A new contribution to the solution-finding quest in this research domain relates to climate and weather anticipation.

In describing the technical aspects of the sensors, their operation and their relationship with a functioning vehicle, the thesis sets out to examine and test the claim that weather detection by means of light and radar data gathering instruments, when used in concert with camera and other terrestrial sensing equipment, can generate a reliable set of data to inform vehicle and cloud based control systems. The integral components for meeting the challenges outlined above, contribute to cooperative driver assistant functionality in particular for driving safety or fuel efficiency, and the creation of a universally accepted, future autonomous vehicle design.

Zusammenfassung

Die weltweit steigende Zahl von Fahrzeugzulassungen und die insbesondere in städtischen Gebieten daraus resultierende steigenden Verkehrsdichte stellt für Regierungen und Verkehrsplaner, aus Sicht der Straßenverkehrswegeinfrastruktur und des Verkehrsflussmanagements, eine wachsende Herausforderung dar. Das steigende Bewusstsein für die schädlichen Auswirkungen der Kraftfahrzeugemissionen und der sozialen Aspekte von Reise- und Fahrzeiten, besonders während hoher Verkehrsaufkommen, werden in jüngster Zeit auch verstärkt Anstrengungen unternommen, um diese Probleme zu überwinden und ihre negativen Aspekte zu mindern. Das für den Straßenverkehr erklärte Ziel zur Senkung von Luftverschmutzung und Unfallzahlen, bei gleichzeitiger Verminderung des Kraftstoffverbrauches, ist ein zentraler Aspekt für Studien in den Bereichen ‚Technologie' und des ‚Fahr- und Fahrerverhaltens'.

Diese Arbeit beschäftigt sich zunächst mit den Ergebnissen vorausgegangener Forschungen die nennenswerte und tragfähige Lösungen zu diesen Themen erschlossen haben. Die Kernthemen dieser Arbeit beschäftigen sich mit Fahrzeugsensoren, die Klima- und Wetterdaten in Echtzeit empfangen und sie schnellstmöglich an die fahrzeugüberwachenden Systeme zur Herleitung von operativen Verhaltensmodifikationen weiterreichen. Die Daten externer Quellen, bordeigener Sensoren und regionaler Wetterberichte werden mit denen aus anderen Fahrzeugsensoren kombiniert, um das aufkeimende ‚intelligente' Fahrzeug bereitstellen zu können. Ein neuer Beitrag zur Lösungsfindung in diesem Forschungssektor betrifft die Klima- und Wettererwartung.

Ein Schwerpunkt dieser Forschungsarbeit ist die Wettererkennung mit Hilfe von Licht und Radar erfassender Systeme, wobei die technischen Aspekte der Sensorik und ihr Zusammenwirken im Fahrzeug detailliert erforscht werden. Die Ausstattung dieser Systeme mit Kameras und weiteren terrestrischen Sensoren ermöglicht die Bereitstellung einer Menge von zuverlässigen Daten, sowohl für die Fahrzeugsysteme, als auch für Cloud basierte Controller. Die wesentlichen Bestandteile zur Erreichung der obigen Herausforderungen tragen zur Umsetzung von neuen Funktionen für kooperative Fahrerassistenzsysteme bei. Insbesondere profitieren hiervon die Fahrsicherheit, die Optimierung des Treibstoffverbrauches sowie der CO_2-Emissionen und die Schaffung eines allgemein akzeptieren Designs zukünftiger autonomer Fahrzeuge.

Content

List of Figures

List of Tables

1 Introduction

This thesis is the result of an investigation into the future for autonomous vehicles. That is, vehicles that can journey with the aid and intervention of on-board technology. For some time, vehicle control instruments such as automatically activating window wipers and automatic stability braking systems have been standard driver assistance features. Parking sensors, automatic illumination and cruise control systems are other examples. More recently, blind-spot and vehicle-to-vehicle distance detection with resulting intervention for speed control have become options in some vehicles. The work described here, collates the concepts underlying the instruments and their implementation for what is now, together with some new research motivated by both safety and eco-environmental considerations for what can be in the future. It describes recent research in vehicle sensor technology and its interface with in-journey vehicle control systems. It also addresses new approaches to data sharing from vehicle-to-vehicle in a swarm (crowd/network) of vehicles on the road, supplemented by information from Cloud Technology based repositories of weather and road condition data. The technology infrastructure to enable this dynamical system to operate is described and discussed. It is proposed here, that the collation and experimental work of this research, towards the design and future implementation of a fully integrated 'vehicle as a mobile sensor', provides a unique contribution to the state-of-knowledge in the *Autonomous Vehicle Concept* domain. In essence, this research describes a novel paradigm, which integrates existing research with some new ideas, proposing a viable way forward for future work in this field of endeavour.

An assumption, based on empirical evidence, is that with continuing growth in traffic volume, there is an increase in the probability of vehicle accidents. Figure 1-1 depicts this trend. It is proposed therefore that the need for improvement in road safety will become increasingly important. Figure 1-2 illustrates this need. One way to approach this need is to design and implement configurations of on-board sensors in a collaborative architecture such that they communicate with each other and collectively with the vehicle control systems. Sensors that monitor the environment through which the vehicle is travelling is one vital component of the overall need to improve road and passenger safety. Communication between vehicles to share environment information, especially weather condition data, is a natural extension of the individual information system proposed for on-board sensor arrays. Due to improvements in mobile communication technology with higher bandwidth and comprehensive mobile network coverage, the infrastructure exists for this potential vehicle-to-vehicle collaboration. The vehicle as a mobile real-time sensor can acquire data such as:

- Environmental weather
- Temperature, pressure, humidity
- Rain sensor, illumination
- Speed, acceleration
- Position, relative position

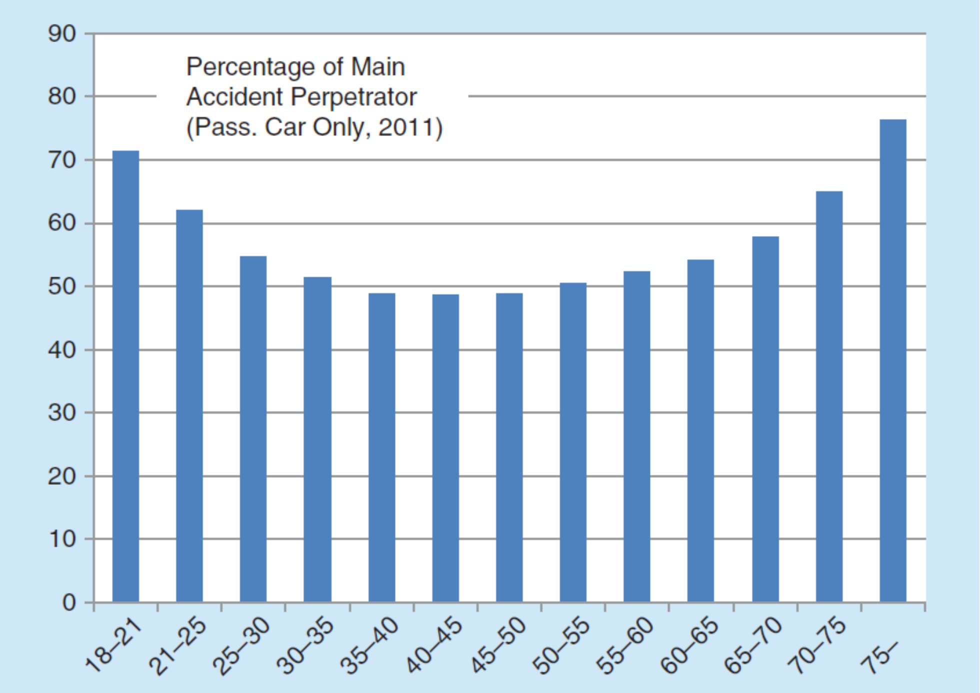


Figure 1-1. Origin of accidents with injuries grouped by age [1, 2]

Using a wide variety of sensors, modern vehicles already collect extensive data such as outside temperature, vehicle speed, rainfall intensity, brake usage, acceleration, road condition according to ASP and ESP data, traffic density, adaptive cruise control, collision detection and prevention, number of passengers as well as current GPS position. It is proposed here that these sensors could be supplemented with additional climate and weather monitoring devices to enable a whole new generation of assistant systems.

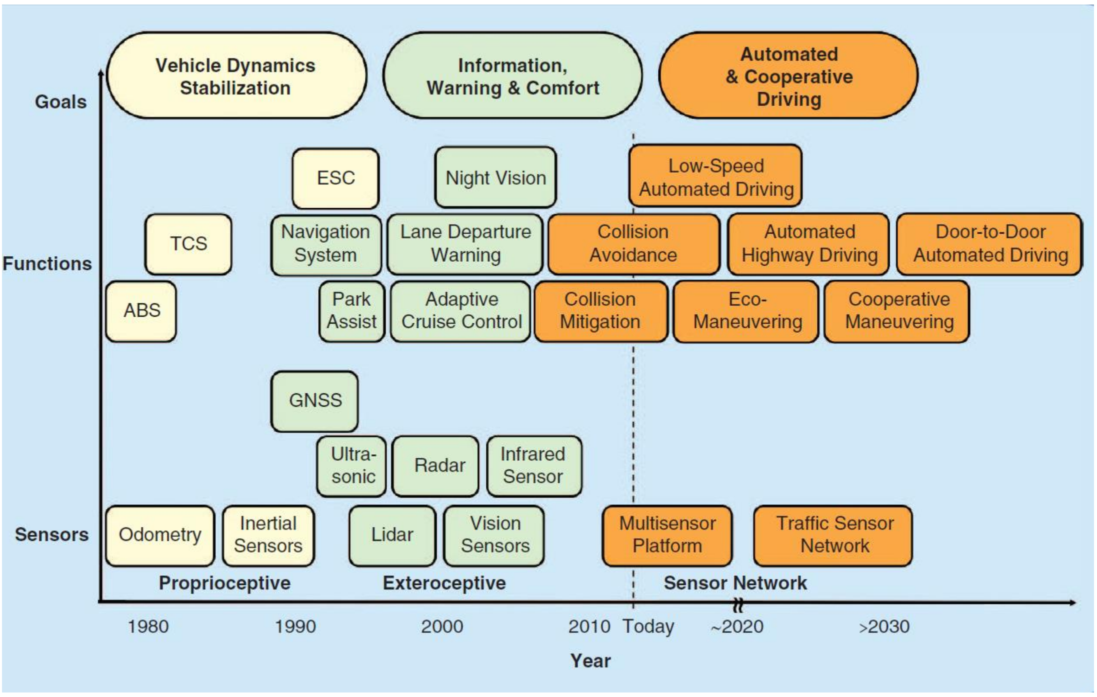


Figure 1-2. Evolution of future driving and safety [1]

Cloud-based services are able to process the incoming real time data stream originating from a large number of vehicles at various different locations. After clustering the incoming data, associations with the world wide geographic street map system are made with respect to the particular data each vehicle delivers at certain locations and time during its journey. Subsequently a real time semantic analysis of the data will assure the provision of:

- Weather warnings (hazards)
- Traffic hazards
- Traffic navigational proposals
- Weather updates for the existing static weather systems

This thesis presents a concept for intelligent and networked driving assistant systems, which in a timely manner alerts the driver to external events and conditions. It is suggested that this in turn would lead to an increase in both safety and comfort.

1.1 Outline of the thesis

State of the art

This section of the thesis outlines the capabilities of contemporary telemetry technology (sensor instruments and associated software) to recognize and in a controlled manner, to guide an operating road vehicle in the context of the ambient weather influences and traffic flow. The following chapter (Chapter 2) describes research and development related to the on-board technology required to carry out the experiments.

The research car

The research car will be described as a set of sensor clusters: focusing first, on the technical equipment that enables the vehicle to be used as a mobile multi-purpose sensor; second, its wireless (and thereby, flexible, communication system; then the cars actuators to implement the optimization strategies delivered from analyzing the sensor-supplied data; and finally, the vehicle's decentralized computational intelligence functionality, in order to apply vehicle local control algorithms…in other words, to inform the vehicle's in-drive control system. For any driving session then, this concept layer regards the vehicle as a mobile sensor.

The vehicle as a mobile sensor

The concept of the vehicle as mobile sensor (probe) within wireless and ad-hoc networks is introduced here. As an autonomous device, the vehicle can be 'promoted' to being a collaborative member of the traffic cloud because its data is submitted with that from other vehicles to a central repository where information may be shared to and from individual vehicles. In this way, the collaborative concept expands to include the analysis, implementation and validation of a distributed, dynamic vehicular sensor array. This is in effect, a dynamical system; non-orchestrated and providing a real-time closed loop control for optimizing traffic management. The next section deals with more complex sensors instrumentation, designed to gain ambient weather information usable by the vehicle for road condition management.

Weather detection by means of camera and LIDAR

A rain sensor output fused with wiper status (level of action) indicates rain and its intensity. Supplemental information might be obtained from more complex sensors deriving weather data like fog, snow, rain or mist. Experiments being conducted on vison and LIDAR (Light Detection and Ranging) based sensors will be researched and discussed in detail. The fusion of different types of sensor data complements the accuracy of expected outputs. Forming traffic messages out of those data elements, the next chapter deals in particular with addressing and routing to the appropriate receivers such as vehicle groups or traffic management centers.

Traffic messages and their addressing and routing

Traffic messages are spatial-temporal in nature. They relate to location and time. . Their potential informational impact on the flowing traffic is more or less spatial-temporal too. For this concept to become reliable, a comprehensive traffic message addressing and routing system is necessary. To illustrate this by example, a location based addressing and routing approach will be described, followed by a cloud based weather service subscriber concept. As traffic volumes increase, so will the volume of data and messages, so an infrastructure appropriate to this load is required. In the next section, a technology solution to deal with this large-scale real-time data storage and analysis is proposed.

Massive Online Analysis (MOA) of the vehicular sensor data

It follow that with future vehicles acting as mobile sensors in vehicular networks, a huge data volume will occur in real time. For a subsequent real time analysis, methods and algorithms have to be provided to manage the incoming data stream. For location and time based interpretation of those inputs with regard to interesting questions like traffic flows, congestions or weather hazards, methods of massive online analysis and deployments of ‘spatial-temporal’ data base patterns will be deployed. One of the exemplary results is affecting the advised speed setting for succeeding traffic and this will be discussed in more detailed within the next chapter.

Vehicles speed profile as a key for efficiency and safety

In the context of the proposition made here that a vehicle can be conceived as a mobile sensor array, an obvious influence on its in-journey behavior would be speed control. The effect of this is the contribution to safety provided by the sensor informed vehicle control system. There is also a contribution to vehicle efficiency due to the minimization of energy consumption coefficients for every journey travelled. This aspect of the underlying concept in this research is discussed later in the thesis, see Chapter 8.5.

Applications for eco-friendly and safe driving

‘*Active driving safety*’ is a concept that puts the focus on improved driving safety by the introduction of highly sensitive ‘all round’ vehicle sensors. Additionally, wireless communication technologies enable a collaborative exchange of traffic messages among vehicles and traffic management centers. In this way, the vehicle obtains essential data along its routing to increase driving safety in order to respond in real time when in situations such as traffic congestion, accident occurrences or degrading weather – mist or fog, rain with wet and slippery roads, high winds, sleet, snow and so on.

‘*Green, eco-friendly and economical driving*’ utilizes sensor data and collaborative traffic messages to improve green and eco-friendly driving. The main target is to obtain fuel efficiency and CO_2 reductions.

‘*Collaborative vehicles and smart traffic lights*’ will even boost the harmonization of the traffic flow and consequently deliver a valuable contribution to fuel efficiency and reduction of emissions. It is an example of a consistent change of the control paradigms of future traffic systems and opens a wide range of new research aspects in this context.

‘*Drive-Guard ADAS weather assistant application*’ treats in sequence the vehicle as a mobile sensor, the traffic management system as a controller and at least the vehicle as an actuator within a closed control loop. Particular focus will also initially be put on more complex sensors describing their function and benefit for selected ADAS (Advances Driver Assistance Systems) functions. It is followed by a geographical traffic message addressing and routing, combined with big data analysis within the traffic cloud backend system. An analysis of the real life data captured during the conducted experiments proves and shows the validity of the researched and developed concepts.

Conclusion

A comprehensive summary of this thesis looking also to potential future work in terms of technology and application research.

2 State of the art

Today’s traffic management (TM) systems consider the conventional vehicle flow. For measurement of flow indicators such as density, congestions or other traffic hazards, many technical systems such as induction loops, cameras, magnetometers, radar or ultrasonic devices (spot sensors) are in daily operation. The communication between these sensors and the traffic management backend takes place through almost wired networks. Vehicular group communication among the traffic members itself does not happen at the moment. Traffic management systems (backend systems) are organized centrally, on a decentralized basis and hierarchically. Traffic management systems make use of traffic lights, road semaphores, traffic flow restrictions (such as speed limits, hazard warnings, no-passing’s …). These actuators are installed at fixed locations and due to computerized visualization techniques, the content can be easily changed on demand. Wired networks are in charge of the communication between backend-based traffic management systems and roadside actuators. In summary, today’s vehicle does not play any role as a sensor or as an actuator automatically influenced by TM.

Today’s traffic sensors, actuators, TM systems and their communication networks have been discussed. These devices and systems are important integral components of a traffic control process as illustrated in Figure 2-1. Using current technology, the real world’s road traffic sensors deliver (not always in real time) measurements (see above) for backend (TM) data processing. Control strategies influenced by technology and methodology (Figure 2-1) are in place to trigger and control the respective actuators for subsequent traffic flow interference [3]. The weaknesses in terms of growing traffic at least on today’s urban roads are the limited real live sensors, their distribution, the type of sensors and their limited real-time communication to network nodes, backend systems or among ambient traffic.

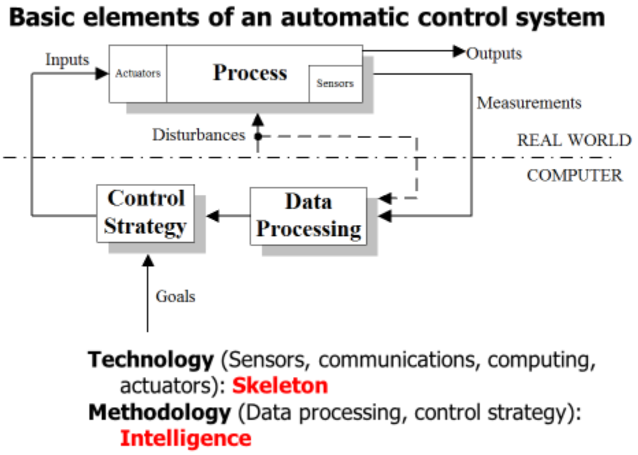

Basic elements of an automatic control system
Inputs
Actuators
Process
Sensors
Outputs
Measurements
Disturbances
REAL WORLD
COMPUTER
Control Strategy
Data Processing
Goals
Technology (Sensors, communications, computing, actuators): Skeleton
Methodology (Data processing, control strategy): Intelligence

Figure 2-1: Basic traffic control system [3]

The deficiencies regarding today’s possible actuators in terms of intervention in traffic flow control are latency, restricted action types, weak type of intervention (speed advice is not mandatorily observed). In summary it can be pointed out, that the control loop from sensing – traffic situation analysis – applying goals for control strategy –actuators intervention is too cumbersome too meet the needs of current traffic systems, at least at cities. The traffic management control loop requires a much higher dynamic level to fulfil today's and in particular tomorrow's TM requirements [4]. Figure 2-2 illustrates how an increased degree of automated and assisted driving leads to a higher degree of (wireless) cooperation.

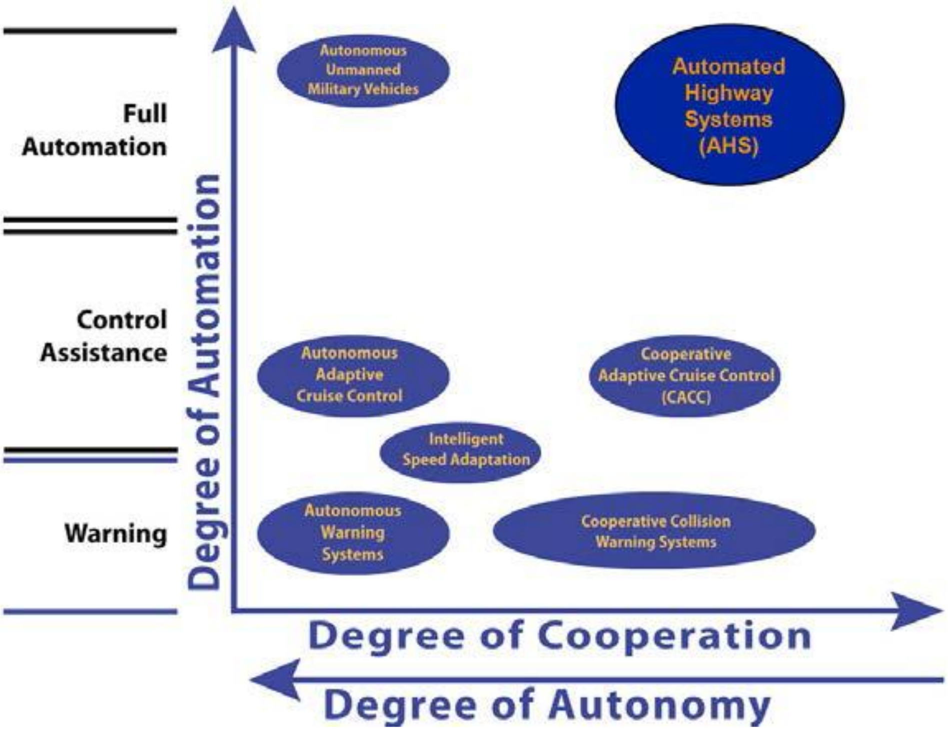

Full Automation
Control Assistance
Warning
Degree of Automation
Autonomous Unmanned Military Vehicles
Automated Highway Systems (AHS)
Autonomous Adaptive Cruise Control
Cooperative Adaptive Cruise Control (CACC)
Intelligent Speed Adaptation
Autonomous Warning Systems
Cooperative Collision Warning Systems
Degree of Cooperation
Degree of Autonomy

Figure 2-2. Automation and cooperation alternatives according to Shladover [5]

The more automated driver’s assistance applications become mature the more corresponding TM actuator applications will have to be made available for traffic infrastructure deployment [6]. The illustration shown in Figure 2-3 reflects the increasing need for automated TM infrastructure systems to cope with the corresponding applications. The marked areas are the particular focus of this thesis.

Since an increasing extent of new future ADAS functionality (Advanced Driver Assistance Systems) requires wireless network connectivity, an important step towards the next generation of traffic systems is taking place. As discussed above, sensing and responding directly at the vehicle is a key entry factor for the next generation of future traffic systems. Performant wireless communication (latency and data bandwidth) will enable real time sharing of the vehicle's sensors. The “vehicle will be treated a vehicular sensor in a collaborative network” – the title and focus of this thesis. Communication will take place among traffic partners and between traffic and infrastructure (traffic cloud services).

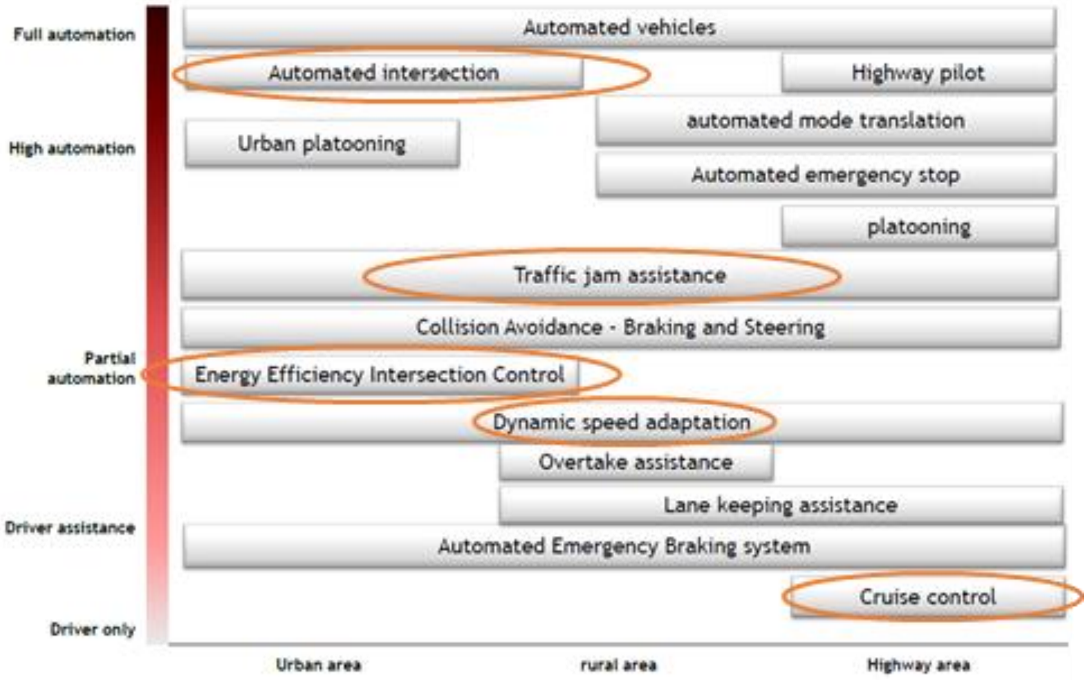


Figure 2-3. Functional mapping of applications [6]

Cloud-based centralized computing enables a wide range of data processing and data mining with respect to situational traffic analysis. TM systems can easily broadcast location-based messages to the flowing traffic to alert but also enforce changes in the vehicle's setting to achieve particular goals (harmonized traffic, reduce emission, reduce noise, …). Vehicle to vehicle communication (V2V, Car to Car - C2C) in combination with massive decentralized computing opens up a wide range of ADAS applications such as WLAN traffic lights, weather assist, hazardous event warning and rerouting, lane merging, etc. The vehicle itself becomes an actuator of the cloud-based TM and its own on-board TM controller.

TM authorities will be enabled to define particular goals for comprehensive traffic management as shown in Figure 2-1. Geographical location-based broadcasting is required to set such goals or rules jointly in place. On basis of such a traffic flow influencing technology, the target to reduce emissions and to improve traffic safety can be researched comprehensively through the different threads of this thesis.

3 The research car

The BMW 530 illustrated in Figure 3-1 is a standard serial production car especially instrumented for the research work of this thesis. It is equipped with a wide range of supplemental devices to enable scientific research studies and evaluations. The picture shows some of the primary sensors, communications and the head up display (HUD).

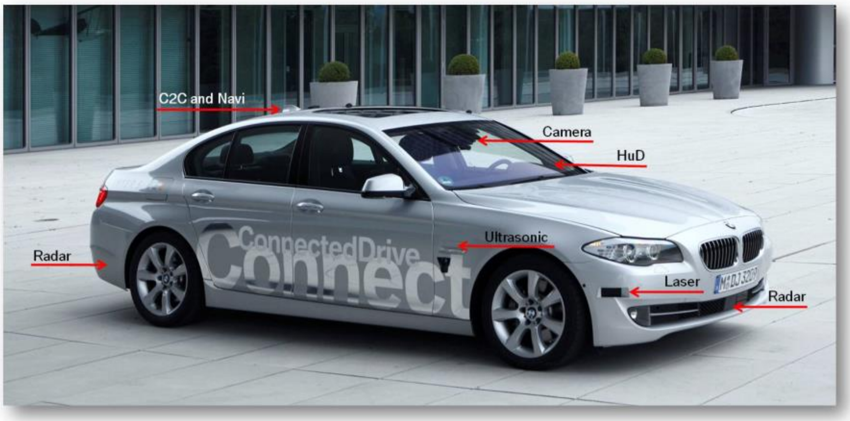


Figure 3-1. BMW research car

The researcher’s workplace architecture is grouped into four categories, see Figure 3-2. Within the first group the CAN-based (Communication Area Network) vehicle networks connects the non-complex sensors (such as temperature, fuel consumption or pressure), the car' engines state (maintaining the current operational state of the involved technical components) and access to the actors. The ethernet-based second group comprises the more complex sensors such as LIDAR or the camera.

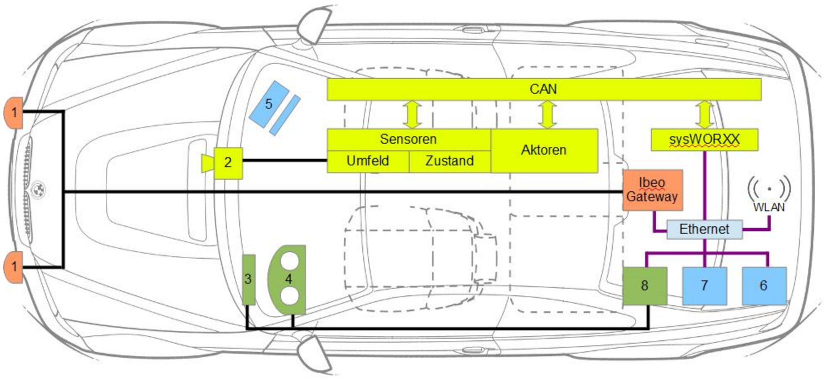


Figure 3-2. Sensor architecture for the research scope

1	LIDAR	5	Notebook for ADAS Control Purposes
2	Video Camera	6	dSPACE Autobox for Odometry & Control
3	Head up Display	7	ADAS Computer (Black-Hole)
4	Instrument Display	8	ADAS-RP Computer, for MAP, Weather, Control Instrument Panel, HUD (FLORIS)

The driver's user interface (comprehensive visualization) and the information aggregation and formation are part of the third group. Last but not least, the forth group comprises the researcher’s workplace, in particular the computer to access the car's network to obtain evaluation data and to modify particular settings for the experiments.

3.1 Sensors

A wide range of sensors are installed in the BMW 530 research car. They are categorized into functional groups. For this research the following groups are considered:

- Vehicles environment and position (objects, traffic, distances, road temperature)
- Odometry sensors
- Acceleration, deceleration
- Fuel consumption
- Pollution, emissions
- Direct meteorological parameters (temperature, barometric pressure)

These types of sensors deliver more or less **direct** meteorological measurements or observations of their environment (outside temperature, barometric air pressure). This type of weather data tokens should be made available to public weather and traffic services as input for weather or environmental models. Secondly, these precise data tokens should be made available for spatial and temporal resolution purposes for meteorological weather awareness functionality.

Vehicle sensor data can be achieved **indirectly** from other input sources, too. One example is the status of the windshield wiper and its current switching state. In the case of a high wiper frequency, heavy rainfall can be implied. The following derived sensor inputs are regarded as **indirect** inputs from virtual sensors:

- Wiper – rain intensity
- Traction control ABS, stabilization control ESP – road friction
- Headlight status – vision, weather hazards (during daytime), fog, snow, rain
- Heating – indicator for reduced temperature (in summer: weather hazard)
- Built-in rain sensor – camera used to measure visibility and fog
- MAP derived inputs – distances to map known objects
- Rate of change of steering – road conditions
- Time, date – influencing weather factors
- Position, location, altitude – influencing weather factor

Sensor (weather) data tokens are made available within the vehicle systems. As soon as the vehicle is a member of a vehicular network, the sensor data marked for data exchange within the vehicular cloud will be published for any other collaborative applications.

More complex sensors such as:

- Ultrasonic: standard equipment for PDC (Park Distance Control).
- Radar: Series equipment for ACC (Adaptive Cruise Control)
- LIDAR: research device for e.g. weather assistant function
- Video camera: research device for traffic signs, weather assistant

Those complex sensors do not deliver raw data to the vehicular network. Specific algorithms are used to derive particular information, such as distance to preceding cars or obstacles. Thereupon theses data aggregations including their semantics are conveyed for submission and propagation to the vehicular cloud. Figure 3-3 illustrates some complex sensors at the front bumper of the research car.

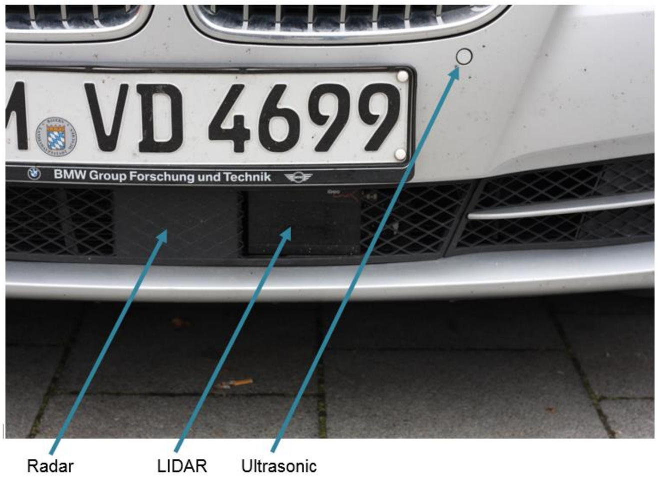


Figure 3-3. The front sensors

3.2 Sensing, measuring, communication and controller equipment

In order to conduct research under real-life conditions, particular test and research vehicles are equipped with latest sensors as well as high-performance communication and computer technology. For this thesis, BMW provided a ConnectedDrive vehicle. ConnectedDrive combines BMW´s extraordinary high-technology equipment and provides the driver with an intelligent networking of vehicle and environment [7].

Inside the vehicle's luggage compartment there are three computers installed to house the advanced driving assistance systems (ADAS). Special devices serve as gateways to interface these computers with the different vehicle network systems. Sensors and actors can be addressed therefore through the controlling computers. Figure 3-4 shows the so called “Black-Hole”: a slot for changeable hard disks. New or updated software versions can easily be deployed at the vehicle, the entire driver assistance system (DAS) can be easily replaced. The permanently installed second hard disk contains a Windows 7 partition but can be bypassed by an external drive to boot from another operating system like Linux. This computer runs and operates the main logic of the driver assistant system. The complementary dSPACE AutoBox is a real-time system for running software for odometry and the controllers. A computer called FLORIS is responsible for the dashboard and the heads-up and control display.

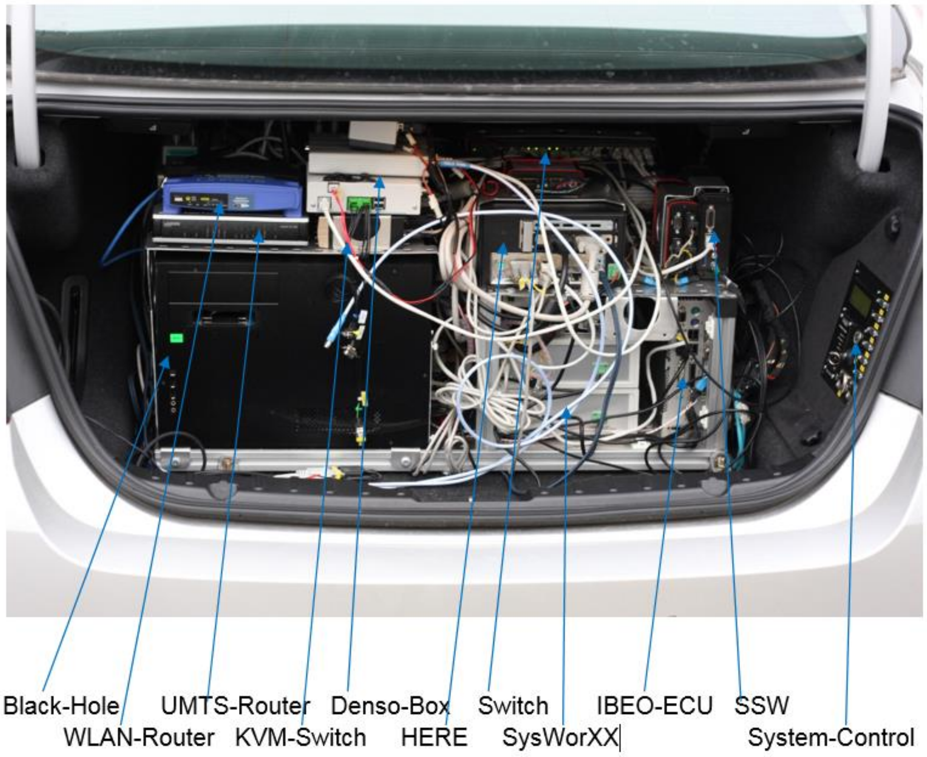


Figure 3-4. Rear compartment

The NAVTEQ software is installed in the third computer: its main task is to deal with traffic information and electronic map data. The dashboard can be used as a FLORIS monitor too, while the other two computers will exploit the control display. The control screen is located in the middle of the instrument panel.

A wireless keyboard with integrated touchpad acts as the input device for the above-described computers. Fig. 2-2 shows the control panel of the ConnectedDrive research platform. The screen of the NAVTEQ computer is shown on the control display. Particular keyboard control sequences can swap the control display between Blackhole and NAVTEQ computer so that the researcher or developer can control program sequences during test runs on the road.

The on-board computers can be controlled via remote desktop connection. An IP-based internet connection is assured through LTE connectivity. The communication system failover functionality covers emergency calls in case of a severe accident. Two stereo cameras with LIDAR (Light Detection and Ranging) and radar (short and long range) systems are used for environmental recognition. The front camera is installed at the windshield mirror, the rear camera is mounted at the rear window. The LIDAR sensor can be found below the front radiator, the radar devices are integrated into the front bumper. Figure 3-2 illustrates the hardware architecture of the research vehicle.

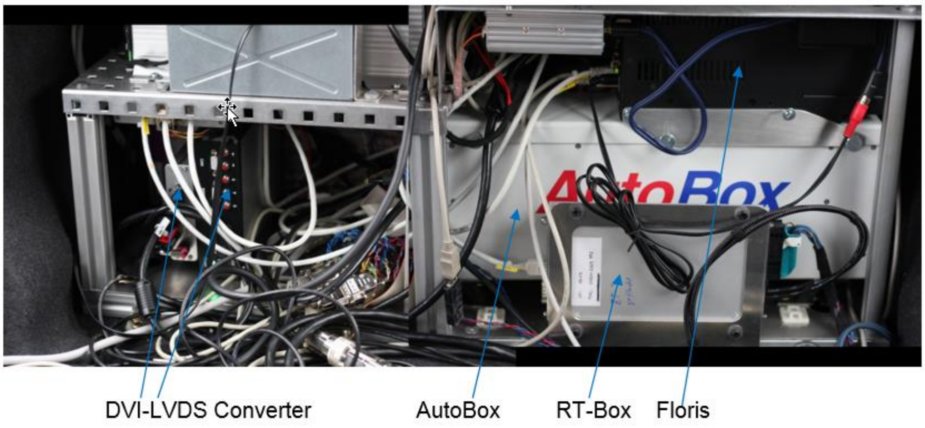


Figure 3-5. Rear seat compartment

Inside the research car’s rear luggage compartment illustrated in Figure 3-4 and Figure 3-5, the following technical infrastructure for computing, sensing, communication and controllers has been installed. The devices shown have the following functions:

- **Blackhole**: Regular computer hosting the measuring framework. Can work on BMW MicroFW (Micro Framework is a BMW specific development, simulation and integration framework for vehicular research purposes) or ROS (Robot Operating System).

- **UMTS/LTE-router**: The interface to the mobile internet. Can be used as bridge to other vehicles (C2X communication).

- **Denso Box**: Interface to several WLAN antennas.

- **Network Switch**: Ethernet and IP connector for the research HW/SW system.

- **IBEO-ECU**: IBEO-ECU for object detection, fusion and tracking of the LIDAR sensors.

- **SSW**: Checks for special software release versions of the research vehicles ECU’s and alerts the driver if no regular production release is being used.

- **WLAN-Router**: vehicular wireless LAN access on 802.11p basis.

- **KVM-Switch**: Display and keyboard multiplexer for the on board computers.

- **Nav-Tec**: Particular computer for controlling and management of the ACC research experiments. It controls the HuD (Head Up Display) visualization. This computer is directly connected via Vector CAN-case to the vehicles CAN bus and controls the experimental ACC.

- **SysWorXX**: Connects the required vehicles CAN buses via USB to the Blackhole computer.

- **System Control**: Vehicles CAN access panel and flash interface connector. Wake up and entering sleep mode of the vehicle can be controlled at this panel.

- **DVI-LDVS Converter**: Connector of the vehicles screens with the computers using DVI interfaces.

- **RT-Box**: Real time Tracing and Tracking Box for tracing vehicle cruising data such as speed, light, fuel flow, emissions and other data from interest of a certain experiment.

- **AutoBox**: The dSPACE AutoBox is a real-time system for in-vehicle control experiments such as test drives for powertrain, ABS or chassis control development. In this particular research study the AutoBox is used for experiments using a modified non-standard ACC.

- **Floris**: Computer responsible for creation of the instrument panel representation. Receives data from the RT-Box.

3.3 Display unit and instrument panel

For drivers' information and interaction, at least a visualization device and an interaction panel has to be made available within the vehicle. Three complex, high-performance display units such as HuD (Head-Up Display, see Figure 3-6), combined instrument panel (see Figure 3-7) and dashboard (see Figure 3-8) are used for the primary purposes described below:

Head Up Display (HuD)

The full-colour head-up display is used to project essential vehicle driving parameters, navigational and driving security parameters into the driver's windshield view without requiring the driver to divert his eyes. Figure 3-6 illustrates the current speed, the target ACC speed, legal speed limit and navigational directions. Fuel efficiency, CO_2 efficiency and current relevant road weather disturbances (recognized through methods introduced within this thesis) will be displayed in an UI-ergonomic manner.

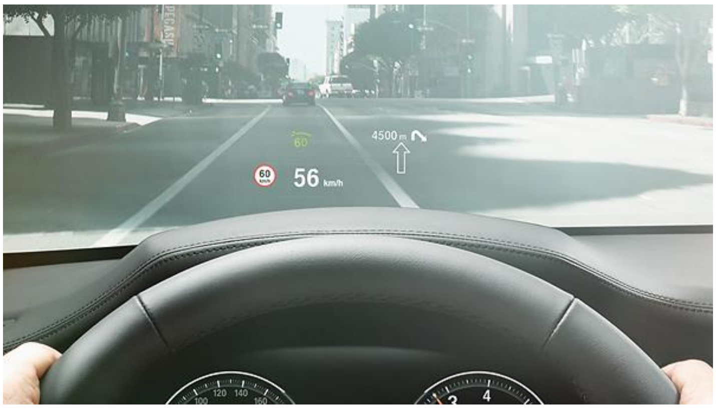


Figure 3-6. Head Up Display (HUD)

As an input methodology the iDrive[1] control (knob) device in combination with an optional associated keyboard (see above) is used to control the research vehicle and the surrounding experiments.

Combined instrument panel

A LCD screen acts as the basis for the combined instrument panel. All of the single instrument indicators are software based images and can easily be modified for research, evaluation and demonstration purposes.

[1] iDrive -The BMW iDrive system enables the driver to completely control many of the vehicle’s functions. A rotating knob with some buttons plays a central role as input device.

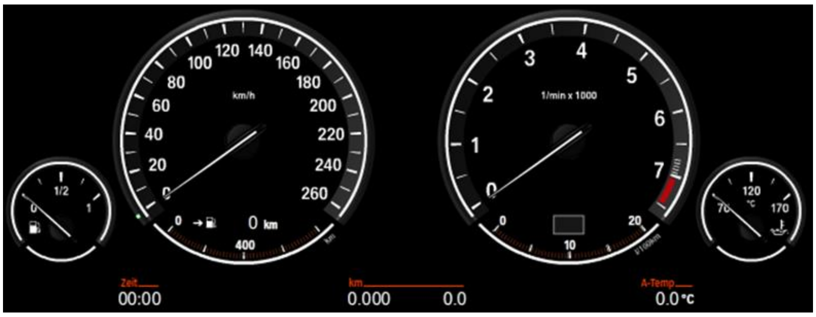


Figure 3-7. Floris based combined instrument panel

The example in Figure 3-7 illustrates a fully featured combined instrument panel of a standard series BMW vehicle. Later in this thesis some modifications are introduced to reflect different drive parameter settings and feed-back loops for fuel-efficient and secure driving.

Dashboard

Auxiliary information and interaction between the researcher and the experimental functions under validation in the research car can be dealt with through the dashboard installed at the centre of the car's main dashboard panel, see Figure 3-8. A particular component using the dashboard during experiments and validation tracks is the ADASRP[2] (Advanced Driver Assistant System Research Platform) application.

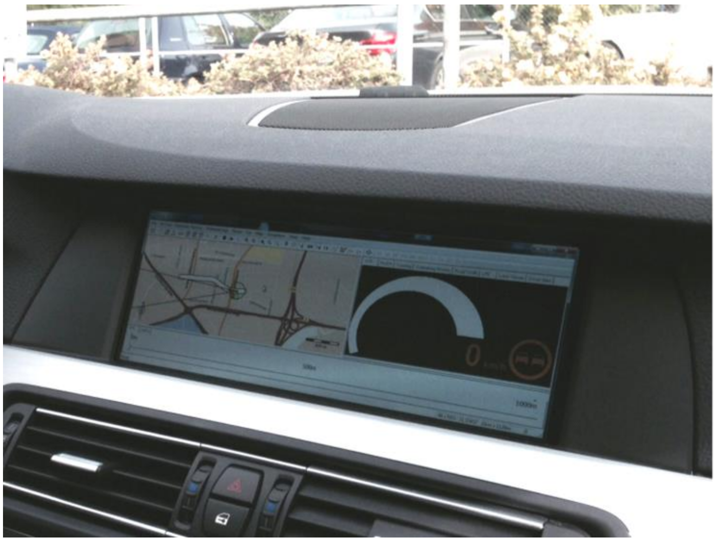


Figure 3-8. Dashboard with ADASRP view

[2] ASASRP form NAVTEQ now HERE (www.here.com)

Other applications are also to use the dashboard for researcher and driver information and interactions. In particular during testing, the dashboard can be segmented for various kinds of trace, tracking and status information and for interaction with the developers. Figure 3-8 shows the ADASRP navigation window and the HuD content (current speed and traffic sign).

3.4 CAN-logger

For the experiments performed in the context of this thesis, it is essential to have access to trace and logging data of the vehicles controller network systems (mainly CAN buses). It eases the debugging of applications under research or development. The CAN logger control device provides a utility for tracing, analysis and visualization of any type of CAN messages. The following detailed information can be obtained:

- Visualization of CAN signals
- Logging of signals and messages
- Visualization of CAN raw data on the buses
- Tracing and logging of GPS positioning data
- Management of the CAN trace and logging database (CANdb)

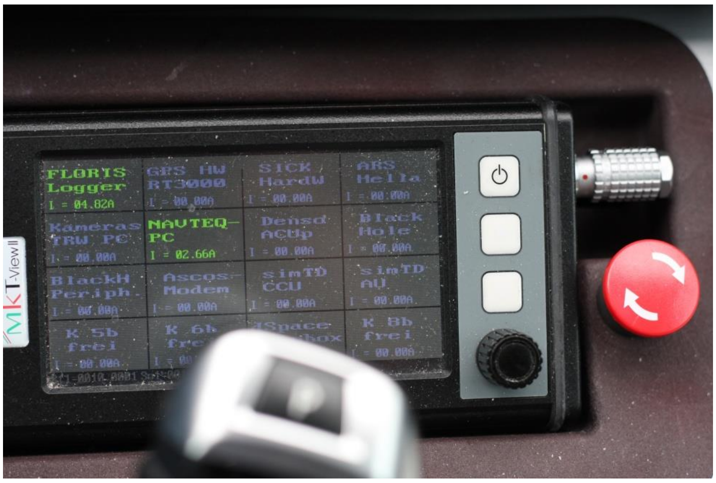


Figure 3-9. CAN-logger switch board

The logging of the “Floris Logger” and the “NAVTEQ-PC” is enabled as illustrated in Figure 3-9. The traced data of the selected device is kept in a particular CANdb database and can be used for further detailed problem analysis. Furthermore the captured data can be used to approve the validity of particular experiments. In order to approve the reduction of CO_2 emissions the associated CAN message is captured over a certain time interval. The subsequent analysis of the measurement series is examined.

3.5 The ACC controller

Advances in sensor development and significant performance improvements in signal processing are opening up new possibilities for the driver in the field of active safety to provide significant support. The technique is to use the driver's skills in terms of cognitive performance and flexibility and to compensate for human weaknesses in terms of precision and speed of reaction. This results in complex technical systems such as the anti-lock braking system and electronic stability program working predominantly in the background. Interventions of the examples occur only in emergency situations and are thus rare and carried out automatically without the direct involvement of the driver, who is informed of their use only by visual, acoustic or tactile means. While such stabilizing systems have already been able to demonstrate their effectiveness with the Adaptive Cruise Control (ACC), there is now a new system available that can take over the driver's managerial responsibilities with the control of speed and distance to the traffic ahead [8] and converging vehicles. In this case, the driver initiates the use of the system aware by delegating tasks to ACC, and even in this respect only exerts control. The goal is to relieve the driver of operation of ACC given the increasing complexity of manoeuvres. In addition to the relieving the driver of the burden of traditional tasks, however, new functions have been added relating to the operation of ACC and the monitoring of its function. In case of a positive result, the question arises to what extent the increase in security is compensated for by adjustments in speed and distance behaviour. The technical performance limits of the vehicle will eventually be fully usable without at the same time increasing the driver's risk tolerance to permanent exhaustion of reserves. The topic of this study is the observation of the vehicle as a mobile sensor and the achievable increase in driving safety by combined use of ACC. The usage of mobile C2C (Car to Car Communication) allows communication among neighbouring vehicles and enables CACC (**C**ollaborative **A**daptive **C**ruise **C**ontrol) concepts and their functionality. After presentation of the sensor technology and the possibilities of influencing the adaptive (collaborative) cruise control system [9, 10], methods are described that can be used to increase driving safety.

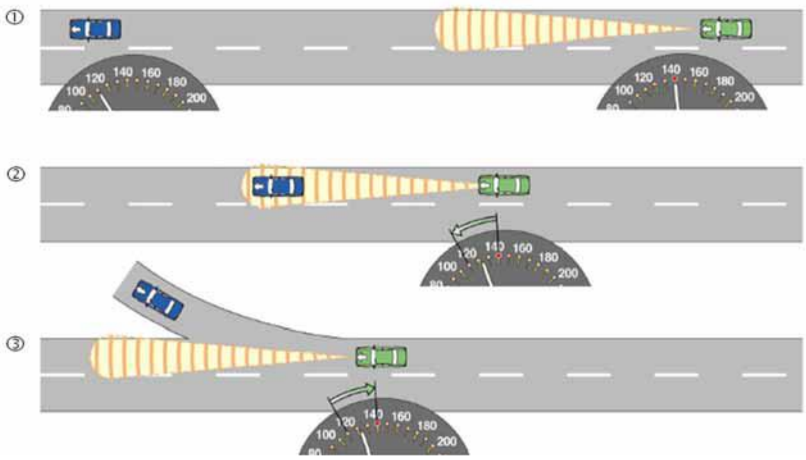


Figure 3-10: Functioning of the ACC [11]

The illustration in Figure 3-10 shows in (1) the ACC vehicle (right) approaching the preceding vehicle at its pre-set speed, with the distance continuously decreasing. Road segment (2) shows the ACC vehicle linked to the preceding vehicle (new speed is applied from the preceding vehicle), automatically following a certain distance and a new target speed. When the preceding vehicle leaves the road, the ACC vehicle accelerates to regain its previous target speed.

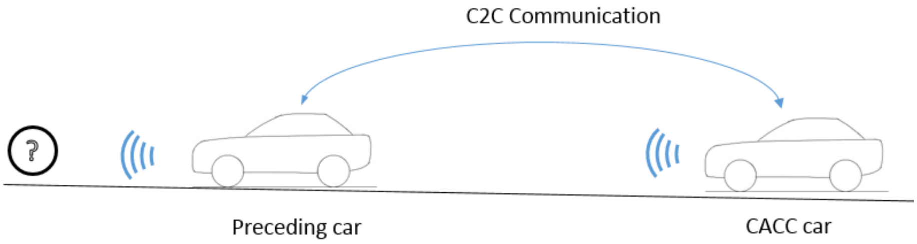


Figure 3-11. CACC through mobile communication

Usually an ACC system uses sensors such as radar or cameras to obtain the distance to the preceding traffic to maintain longitudinal control of a vehicle [12]. Taking the vehicle as a mobile sensor into account, it can even deliver information about current position, speed, intentions, strategies, speed limits, events, weather, etc. On that basis the functional complex “Collaborative ACC” can be designed for much more advanced driver assistance system (ADAS) functionality, such as weather assistance function or hazard warnings.

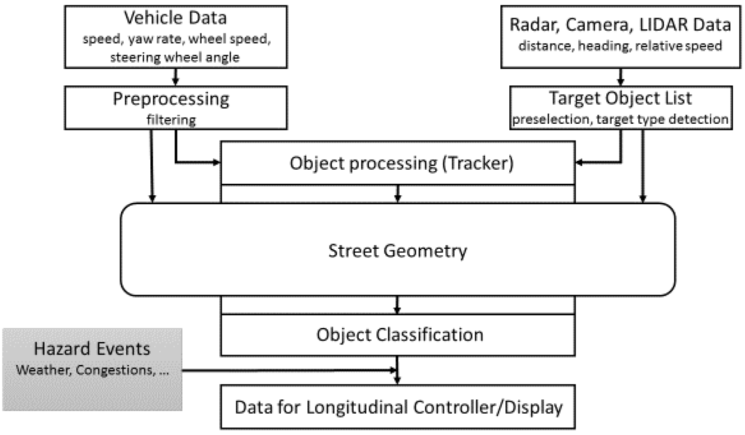

Vehicle Data
speed, yaw rate, wheel speed, steering wheel angle
Radar, Camera, LIDAR Data
distance, heading, relative speed
Preprocessing
filtering
Target Object List
preselection, target type detection
Object processing (Tracker)
Street Geometry
Object Classification
Hazard Events
Weather, Congestions, ...
Data for Longitudinal Controller/Display

Figure 3-12: ACC controller becoming CACC-enabled [13]

The ACC controller shown in Figure 3-12 is the basis for a CACC System (Collaborative Adaptive Cruise Control) used in several experiments conducted in context of this thesis. The above figure shows the functional blocks for object detection (preceding objects, traffic signs, road curvature and relative speed) on the right-hand side. On the left, the functional blocks for dealing with vehicle data (speed, yaw rate and other odometry data) are shown. Together with map data (street geometry in the middle of the illustration) and wirelessly transmitted messages (left-hand side – hazards, events, weather and congestions) the ACC controller becomes a CACC controller taking preceding traffic messages into account. The resulting longitudinal data serves as controller input for the ACC algorithm.

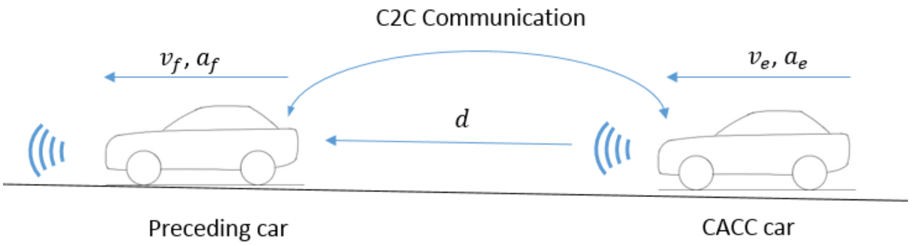


Figure 3-13. CACC control algorithm

The real technical challenge is the determination and development of a well-suited controller algorithm taking the car's dynamic model into account (see Figure 3-14, Vehicle Dynamics box). Drivers comfort and security must not be affected, so particular attention has to be paid to obtaining seamless and smooth acceleration and deceleration; overshoots and controller oscillations may not appear [8, 14]. The sensors illustrated in Figure 3-14 have to provide a high-resolution image of the longitudinal distance and relative speed to preceding objects, as shown in Figure 3-13. Within equation (3.1) the differential or relative speed is computed from the ego speed v_e , the distance d and the speed v_f of the vehicle ahead.

$$\Delta_v = v_f - v_e = \frac{d}{dt} d \tag{3.1}$$

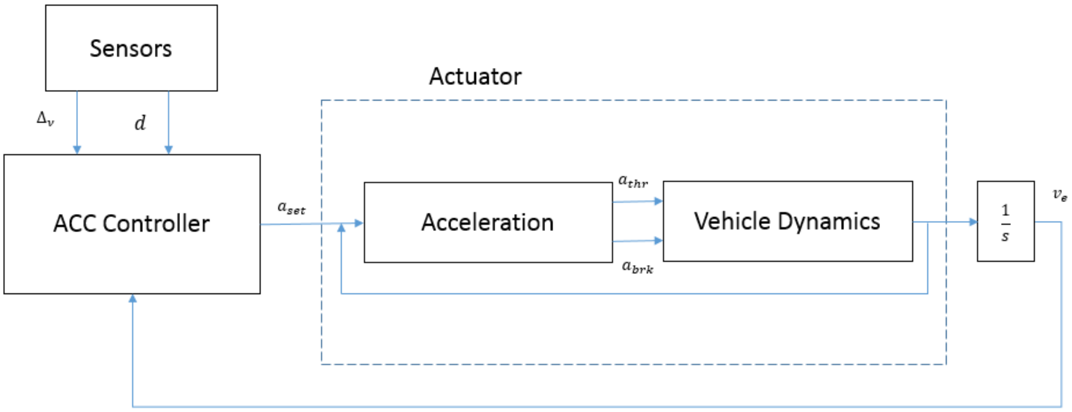


Figure 3-14. ACC controller and actuator

It should be pointed out again that the use of C2C introduces collaborative perception. More precision and an extended anticipatory view of the vehicle will be obtained. These factors are key elements for harmonized and more secure driving. The (collaborative) ACC controller is an important unit for controlling and actuating the vehicle's cruising settings.

Figure 3-14 describes the entire control process, beginning with the sensors (radar, camera, C2C communication or even LIDAR) delivering relative approaching speed and distance (refer to Figure 3-13). The ACC controller itself computes the resulting +/- accelerationa_{set}. Within the actuator element the acceleration controller computes acceleration or deceleration outputs which will be influenced by the vehicle dynamics model. Finally, the best manoeuvre speed will be applied. The whole process remains a control loop until deactivated. HMI outputs and driver' interactions are discussed at a later stage in this thesis.

3.6 TEAM integration

The research vehicle is used in the context of the EU research project TEAM[3] as well (EU Research Project: **T**omorrows **E**lastic **A**daptive **M**obility). Some comprehensive use cases are foreseen in TEAM:

- C-ACC (Collaborative Adaptive Cruise Control)
- Collaborative driving and merging

[3] http://www.collaborative-team.eu/overview/objectives

- Collaborative eco-friendly navigation
- Collaborative parking

These use cases are enriched due to the research of this thesis by treating “the vehicle as a mobile sensor in a vehicular network”.

The integration of the TEAM (EU Research Project: **T**omorrows **E**lastic **A**daptive **M**obility) Application Unit to the Research Car is illustrated in Figure 3-15. Through the illustrated architecture (bi-directional) the vehicle CAN bus data is routed via the “Vehicle Data Provider”, “Application Unit” and Communication Unit (CCU) via LTE communication to the “Roadside Unit” and/or the backend system (traffic cloud).

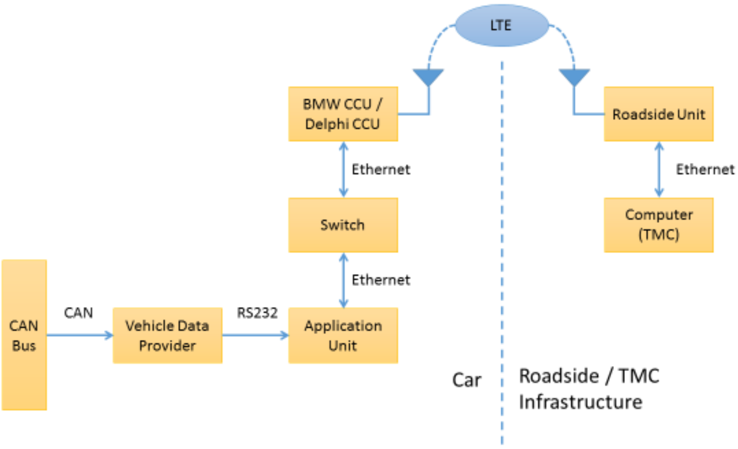


Figure 3-15. Integration of the TEAM application unit into the research car

Some of the above mentioned TEAM use cases will be evaluated from the point of view of addressing a particular set of vehicles in a dedicated local area (e.g. issuing weather hazard warnings for all vehicles in a particular local area) for traffic safety. A message queuing technique [15] is applied to guarantee failsafe operation due to weak network coverage spots [16]. The research vehicle will be “TEAM research”-enabled through the above architecture and its related hardware and software components. Previous functionality remains unaffected, both alternatives can be tested independently.

4 The vehicle as a mobile sensor

Sensors in cars are capable of providing instantaneous information relating to both current and time-averaged speed as well as position. Position information can either be acquired directly by means of an on-board global navigation satellite system (GNSS) if available, or else by means of a collaborative configuration based on positions received from vehicles in the vicinity that are fitted with GNSS and corrected by the position relative to them (HIGHTS Project). Distance sensors such as radar, LIDAR and optical sensors not only provide information on the space to the front and rear of the vehicle, they can also be used to determine local traffic density and improve position accuracy.

All these sensors can be combined to form a sensor network that is superimposed on the Vehicular Adhoc Network (VANET) [17] communication structure. VANET are classified as a mobile ad hoc network (MANET) application with the potential to improve road safety and provide convenience for travellers [18]. A VANET is built by means of wireless communication between moving vehicles using dedicated short-range communication (DSRC). DSRC is essentially IEEE 802.11a that has been amended for low overhead operation to IEEE 802.11p. The IEEE standardizes the whole communication stack according to the 1609 family of standards applicable to wireless access in vehicular environments (WAVE). Vehicles can communicate with other vehicles directly by means of vehicle-to-vehicle communication (V2V) or else using a piece of equipment installed at the side of the road known as a roadside unit (RSU), enabling vehicle-to-infrastructure communication (V2I) [19] or a combination of both.

Contemporary applications designed to benefit from this type of vehicular sensor network can be classified either as safety applications or as traffic monitoring and management systems [20].

Safety applications

Sensor data is exchanged between the cars themselves (V2V). One example here is emergency braking. In the event of an accident or if the brakes are fully deployed, a warning message goes out to vehicles in the proximity. This information, together with the exact location, can even be conveyed to vehicles which are not in the direct line of sight of the source using multi-hop wireless technology.

Other possible safety applications include overtaking assistance as shown in Figure 4-1, safe distance warning and coordination of cars filtering into the main traffic flow [20].

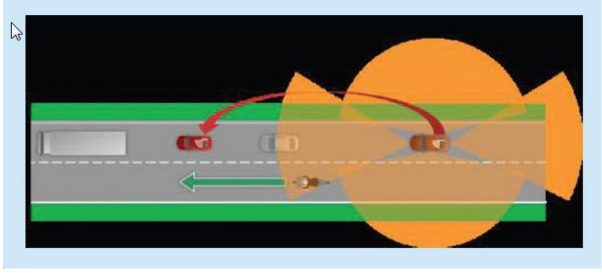


Figure 4-1. C2X Communication preventing accidents [1]

Traffic monitoring and management systems

Nowadays, traffic-monitoring systems are based on a centralized structure in which fixed sensors such as loop detectors and cameras along the roadside monitor traffic density. All sensor data is processed at the traffic management centre. Centralized systems such as this are expensive to develop, maintain and expand. Their widespread deployment is restricted by the extremely large investment that is required in a communication and sensor infrastructure.

An alternative approach is based on VANETs: this is currently being developed in several projects and is under investigation in field trials worldwide. Here vehicles form part of a ubiquitous sensor system: each vehicle monitors the surrounding traffic flow using its own on-board sensors and reports pre-processed parameters such as traffic density and average speed through the network via a wireless data link. In a pilot project in Atlanta (Georgia, USA) fleets of 500 road users - called traffic spies - are equipped with on-board sensors, computers and GPS units so as to provide live monitoring of traffic flow inside the city [23]. Other projects include Advanced Cruise-Assistant Highway Systems (AHS) [21] and Vehicle Information and Communication systems (VICS) [22], FleetNet [23], Auto Net [24] and PATH[4]. The aim of the Japanese AHS system is to reduce traffic accidents, thereby enhancing safety and improving transportation efficiency by reducing manual involvement of drivers. VICS supports the driver by supplying road and traffic information in real time. FleetNet is attempting to develop wireless multi-hop ad hoc networks for inter-vehicle communication so as to improve safety and convenience for both drivers and passengers by providing up-to date information on traffic, weather and road conditions [20].

4.1 Wireless sensor networks

A sensor web [25] is a macro-instrument concept that allows spatiotemporal understanding of an environment based on the coordinated efforts of several different kinds of sensing platform. It is therefore a web of interconnected heterogeneous sensors that are interoperable, intelligent, dynamic, flexible and scalable [26]. While the latter implies that a sensor web is a hardware network of sensors, it can alternatively be defined as a universe of network-accessible sensors, sensory data and information. This definition is more network-oriented and includes sensory data and information, although it does appear to exclude virtual sensors, which are defined in the next paragraph [27].

In order to arrive at a definition of a sensor web, a service-oriented architecture (SOA) is introduced for web services, see **6.1**. In this case, a sensor web can be defined as a group of interoperable web services, all of which comply with a specific set of sensor behaviours and interface specifications. A sensor web can be applied to a specific application domain and is uniquely identified by combining the application domain and the specifications it complies with [27].

If we define a sensor web as a group of interoperable web services, all features of web services are also applicable to the web-ready sensors, which include but are not limited to dynamics, flexibility, plug-and-play, self-description and scalability. This suggests that the data and information generated by individual web-ready sensors is interoperable and that the sensor services are chainable [27].

Such a Wireless Sensor Network (WSN) of small sensor notes is used in [28] to detect all vehicles entering and leaving the area of a logistics centre in the freight village of Turin. The main requirement here was easy sensor allocation on the roads so as to be able to adapt the monitor system to evolving activities and objectives. The sensor network consists of communication hubs using GPRS for data transmission, repeaters and magnetic sensors for detection. Data processing and storage are performed by a software which also allows diagnosis and maintenance of WSN operation. WSN technology was chosen as it allows the highest flexibility at low purchase and operational costs and because of the fact that, unlike most other systems, it has negligible visual recognition ability: in this way, the data collected is not influenced by changes in behaviour that are caused by being watched.

4.2 Vehicular ad hoc networks within the cloud

The automobile sector is perhaps one of the most exciting application areas of mobile wireless ad hoc networks. In future, ad hoc networking technology will be used in vehicle on-board communication units to exchange real-time data relating to traffic and road conditions. Areas of application include services such as safety warning systems, traffic control and real-time traffic re-routing by means of intelligent traffic management systems. This thesis will discuss the unique features and challenges that distinguish such systems from other types of ad hoc sensor networks. We will also consider possible applications of wireless grids in addressing the data aggregation and processing challenges that ubiquitous traffic monitoring and management systems will face [20].

As already mentioned, Vehicular Ad hoc Networks (VANET) [17] are created by vehicles equipped with short and medium range wireless communication technology (DSRC/DMRC) and by communicating without relying on a fixed infrastructure. Communication is conducted between the participating vehicles within each other's radio range (V2V). Fixed gateways along the road called roadside units (RSUs) might also be involved, but these are not a prerequisite. Taking the resulting sensor network as a basis, it is possible to extend this to build safety applications, driver assistance and driver information systems as well as intelligent traffic management systems. The sensor data collected from vehicles and roadside sensors is used for traffic analysis and ultimately to control and manage traffic in real time with an improved granularity as compared to what is common today.

The creation of high-speed, highly scalable and secure vehicular sensor networks poses an extraordinary challenge due to a combination of highly dynamic mobility patterns resulting in highly dynamic network topologies, combined with the high speeds that are sometimes involved. On the other hand, certain limitations commonly assumed in other ad hoc sensor networks are not present in these systems. For example, vehicular sensor networks have access to ample computational and power resources within the network itself and employ high-performance wireless communication technology combined with advanced antenna technology [19, 20].

4.3 Collaborative members of the vehicular cloud

Requirements of the Wireless Sensor Network (WSN)

The environment in which the vehicular sensor network operates is highly dynamic, so accuracy, responsiveness, reliability and timeliness are crucial factors for the success of a system designed to improve safety and convenience at speeds of up to 250 km/h. The density of vehicles - which we might regard as nodes within a wireless sensor network (WSN) - can vary from 1-2 vehicles per kilometre during low density rural night traffic and more than 300 vehicles in bumper-to-bumper traffic during the morning rush hour in a city centre. Additionally, node densities can show large spatial-temporal variations due to dynamic traffic phenomena such as platooning or stop-and-go traffic jam waves [29–32].

The IEEE 802.11p protocol may be used for wireless communication in V2V. In [33] the performance of IEEE 802.11p was analysed on a highway and in a rural and urban environment. Packet loss, Doppler effect, multipath fading and the Hidden Terminal Problem make communication difficult here, to name just a few of the problems. As all nodes share the same medium of communication, an increasing number of nodes will lead to a greater delay. Together with the increasing signal-to-noise ratio, this results in reduced network capacity. What is more, [33] shows that 802.11p is sensitive to vehicle speed, network traffic load, the amount of participating vehicles and vehicle density.

Maintaining end-to-end (E2E) network connectivity, packet routing, timely and reliable information dissemination and high-speed wireless communication in such highly dynamic networks is extremely challenging [20].

V2V and V2I communication, message routing and broadcasting

V2V requires a certain vehicle density to operate effectively, requiring vehicles equipped with a high penetration rate of IEEE 802.11p (also called ITS-G5) [34]. Figure X depicts the process of penetration rate for different introduction strategies of the ITS-G5 technology over the years.

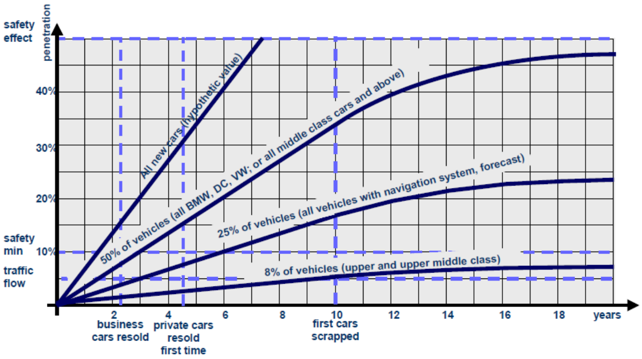


Figure 4-2. Penetration rate of ITS-G5 vehicles in Germany over years for different introduction strategies as estimated by Volkswagen and others [34]

Regarding V2X network connectivity, the main issues are dealing with large inter-vehicle gaps in sparsely populated vehicular networks and the formation of isolated clusters of vehicles as a result of dynamic traffic conditions. One possible solution is the introduction of infrastructure-based servers or gateways along the road with a backbone interconnection to a public or private network. Such gateways might act as bridges between isolated clusters of vehicular networks. Another approach is to use message relay boxes which store messages received from vehicles and relay these at a future time as necessary. In this way, temporary gaps that would otherwise separate an established network can be avoided, the flow of information is maintained and packet loss is prevented.

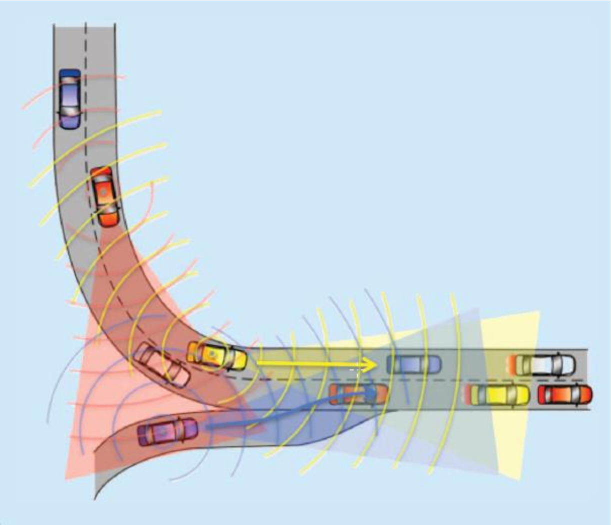


Figure 4-3. Cooperative traffic of communicating vehicles [1]

The next challenge is routing in a mobile ad hoc network as the network's topology is constantly subject to very dynamic change. In vehicular ad hoc networks this problem is inherent and conspicuous due to the high node velocities involved and the alternating velocity pattern.

Routing protocols for mobile ad hoc networks (MANETs) can be broadly divided into topology-based and position-based types. Topology-based methods include pro-active methods based on distance vector or link state and reactive methods such as dynamic source routing. In position-based routing, each node is addressed by a unique identifier which contains information on the position of the node, e.g. its GPS coordinates. In order to forward a packet that contains a destination as a position, an intermediate node only has to be aware of its own position plus the position of a number of its one-hop neighbours. The intermediate node simply forwards the packet to a node closer to the destination than itself. The key feature of position-based routing (see Figure 4-3) is that unlike topology-based routing, no route set-up or route maintenance is required. On the other hand, position-based routing requires a system for dissemination and management of position information so that nodes can acquire the current positions of their anticipated communication partners. The results of recent simulation studies of routing in vehicular ad hoc networks suggests that position-based routing has advantages over purely topology-based methods in terms of adapting to changes in network topology and scalability.

In addition to routing, multi-hop forwarding (i.e. advancing a message along a roadway by transmitting the message from vehicle to vehicle) plays an important role in vehicular networks as it provides a simple but robust method of information dissemination. For example, in traffic monitoring applications, data packets containing information on velocity and local density could hop from car to car until they reach a vehicle close to an internet gateway from where the data can be transferred to the monitoring system. Due to the broadcast nature of communication in vehicular networks, nodes can receive the same packet multiple times and have to contend with their neighbours for retransmission. Hence simple multi-hop forwarding of messages is highly inefficient and a number of algorithms have been proposed to eliminate this inefficiency. Epidemic (or *gossip*) routing protocols are currently being investigated to improve message forwarding in vehicular ad hoc networks.

Data analysis within the automotive cloud

Intelligent traffic management systems will be an exciting leap beyond current traffic management systems. In such systems, data collected from vehicular sensor networks is fed into traffic analysis and simulation software for real-time detection and forecasting of undesirable traffic conditions such as stop-and-go waves and congestion. The results are then used by the control system to dynamically adjust traffic conditions via lane and traffic signal control, ramp metering, cooperative driving and traffic re-routing. Such systems are currently still at an early stage and their development has been hampered by the inability of existing monitoring systems to deliver traffic flow data with sufficient spatial granularity and timeliness. With large-scale deployment of vehicular sensor networks, this can be expected to become available in the near future. When deployed on a global scale, however, the volume of data that vehicular sensor networks will generate will be vast. Real-time aggregation and processing of this data and its storage is non-trivial. Issues such as these present a serious challenge to system designers. These types of problem may be addressed using computational grids [20]. Grids are distributed infrastructures that join together and coordinate vast amounts of computing resources in order to provide users with on-demand access to computing power and disk storage, for instance.

One suggested solution that makes use of grid concepts is where vehicles play the role of both mobile sensors and mobile computers, linked together via wireless connections in order to act as grid computers on the road [26]. A high density of nodes in a segment of the road results in a higher density of potential nodes that can be temporarily organized spontaneously to perform distributed computing and solve a single problem. This network/computing capability can be used to empower vehicle-driver safety applications. It has been suggested that such a vehicular grid might also be used to activate traffic control mechanisms such as ramp metering. Although the above architecture is very attractive because of its autonomous character and the fact that it harnesses the idle CPU cycles of in-vehicle computers, it has some major drawbacks. In particular, optimizing traffic flow locally in a stretch of road using the above system can result in the emergence of undesirable traffic conditions elsewhere in the road network. Perhaps an alternative approach might be intelligent traffic management using a global traffic grid system. This system is fed continuously with sensor data from vehicular networks, creating a pool of computing resources for aggregation, analysis and storage of data as well as real-time simulations, forecasting and control of traffic flows. The above-mentioned wireless vehicular grids could form one of the components in such a global traffic grid, to provide some of the required computing power, for example, or for initial local processing and aggregation of sensor data [20].

The next step will be so-called vehicle automation and communication systems (VACS). With VACS - vehicles used both as sensor and actor - enhanced possibilities will lead to improved traffic management. [3] presents a model of a predictive control framework (MPC) to control a motorway system, taking into account the increased control freedom as well as a more precise assessment of the state of motorway traffic. The core of the methodology is the convex optimization problem based on a piece-wise linear macroscopic traffic flow model. As the necessary quantity of vehicles equipped with the required VACS technology is not available at the moment, the proposed control strategy was evaluated in simulation.

5 Weather detection by means of camera and LIDAR

5.1 Motivation

The method described here is for the automatic recognition of weather conditions from a moving car [35]. The system consists of sensors to acquire data from cameras as well as from Light Detection and Recognition (LIDAR) instruments. Within the scope of this thesis the chapter discusses how data can be collected, analysed and merged to assist the control systems of moving vehicles in making autonomous decisions. Raw, aggregated and pre-analysed vehicular sensor data is provided within the vehicular network for collaborative use [28]. Laboratory based experimental results are presented for weather conditions such as rain and fog, showing that the recognition scenario works with better than adequate results. It is demonstrated here how LIDAR technology, already on-board for the purpose of autonomous driving independently of auxiliary light sources, can be used to improve weather condition recognition as compared to a camera-only system. The conclusion from this study is that the combination of a front camera and a LIDAR laser scanner is well-suited as a sensor instrument set for weather recognition that can contribute accurate data to driving assistance systems.

5.2 Introduction

Driving assistance functions are core technologies in intelligent vehicles. Well-known applications are lane keeping, traffic sign detection and interpretation, lighting assistance and adaptive cruise control. All of these functions are capable of scanning visual images and determining the lanes, signals and lights of preceding or opposite traffic. The recognition of the contextual driving situation by means of the surrounding weather situation is a real challenge and can provide important auxiliary information for many other driving applications. The spectrum of weather condition information obtained by this combination of instruments is categorized using both quantitative and qualitative methods.

LIDAR technology enables advanced object recognition, used for more qualified weather detection in quality and quantity (rain, snow, fog, dust, drizzle, mist). LIDAR is a promising technology for weather detection; due to its very high sensitivity and spatial resolution, this active detection technique enables efficient location of weather effects during both day and night and over a considerable range.

The fusion of camera and LIDAR weather data provides valuable input for gaining reliable and robust information on certain weather effects.

Finally, a neural-network technique for automatic recognition of severe weather signatures on incoming camera and LIDAR data is considered and several neural-network architectures and learning algorithms are evaluated and presented.

5.3 Camera based weather recognition

Vehicle on-board mounted cameras capture the visual signal within which the driver needs to drive safely. Processing this in real time allows the detection of pavement markings, road signs or hazards such as obstacles, in order to assess time to lane crossing, time to collision or other risk indicators. It is also possible to detect reduced visibility conditions such as fog, rain, or glare, and even to quantify their impact on sight distance or target visibility, in order to provide speed warning, distance warning to vehicles ahead or to control light beams. In some cases, visibility can even be restored, up to a point, by de-hazing and in-painting techniques for Head-Up-Displays [36].

5.3.1 Detection and characterization of adverse weather conditions

Contemporary modern cameras are widely used and are a component of almost every advance driver assistance system because of the price and flexibility. Nevertheless the major problem is that cameras can be blind under adverse weather conditions when they are needed the most. To identify and quantify the cause of the degradation of the signal and estimate a confidence index is a major challenge to ADAS development.

5.3.2 Detection of rain, snow and fog

Previous camera-based fog detection systems analyse distinct objects in the image or image regions such as the road region or the horizon. These approaches are not reliable for everyday use. A new method is to use the only reliable visible attribute of foggy weather conditions: the decrease in contrast and blurring in the whole image [37], the power spectrum being the squared magnitude of the Fourier transform for the image that holds information about the frequencies in the image. The matter of discarding spatial information and its decision-informing potential will be analysed. From the power spectrum one builds the image features that can be then fed to the classifier that has been trained for fog and fog-free images. It turns out that in the case of an observed fog scene, the frequency components are concentrated at the zero frequency, whereas in a scene without fog one finds a broadly spread spectrum caused by the contrast attenuation and blurring in the image provoked by the fog. Sharp edges are modelled by many different low and high frequencies, whereas smooth edges are only created by low frequencies.

The key to a single-step fog detection process is how to derive values for the differences in the power spectrum [37, 38].

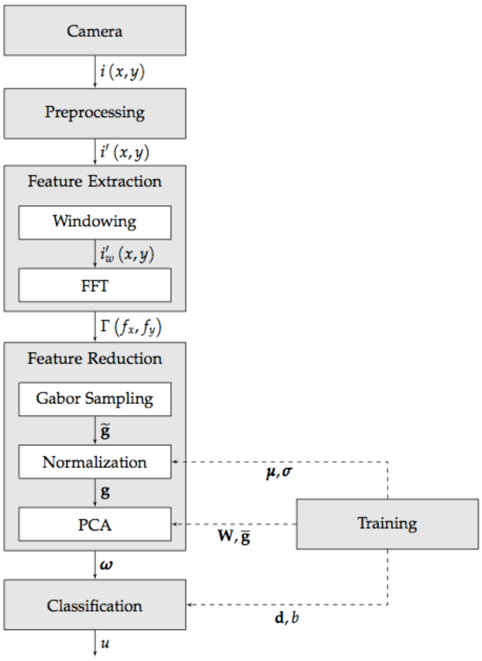


Figure 5-1. Fog Detection Flow Diagram [38]

In the pre-processing step (see Figure 5-1) the image is filtered and normalized to reduce illumination effects and to avoid some image regions dominating the spectrum. For the normalization a square image section is used and the intensity values are equalized.

$$i'(x,y) = \frac{i(x,y) * h\,(x,y)}{\varepsilon + \sqrt{[i(x,y) * h(x,y)]^2 * g(x,y)}} \tag{5.1}$$

The next step is feature extraction. Before the Fourier transformation a windowing with a 2 dimensional Hanning window is applied [37].

$$\Gamma(fx,fy) = \left|\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} i'(x,y)e^{-j2\pi(f_x x + f_y y)}\right| \tag{5.2}$$

For feature reduction as illustrated in Figure 5-1, Gabor sampling by means of a filter bank of scaled and oriented filters is deployed. For the subsequent PCA, the Gabor features are used.

$$G_i\left(f_x, f_y\right) = e^{-2\pi^2\left(\sigma_x^2\left(f'_x - f_r\right)^2 + \sigma_y^2 f_y^2\right)}, i = 1, \ldots K \qquad (5.3)$$

In the last step of the classification process, a SVM (Support Vector Machine) with a Radial Basis Function (RBF) kernel is used [39]. The regularization parameters were determined by training data.

With this approach summarized above, the average accuracy accomplished was 96% on a basis of 44,000 images [37]. The next step would be to analyse how this method would perform on night-time fog images. At night time, light from the headlamps is reflected back by the airborne droplets of the fog in front of the vehicle to the driver, producing a back–scattered veil [37]. This would create a blurring effect as encountered during daytime. Our research investigates the potential for this by extending the method to further distinguish rain and snow from fog.

5.4 LIDAR-based weather recognition

5.4.1 Detection and characterization of adverse weather conditions

Despite their price and versatility, cameras have their disadvantages. One major problem is that cameras can be totally or partially blind under adverse weather conditions, when the systems they feed are most needed. Detecting and quantifying degradations of the signal and identifying their causes constitutes an as yet unresolved challenge for automated driver assistance systems (ADAS). It involves estimating a confidence index, and it involves self-diagnosis [40]. Another sensor technology is LIDAR (Light Detection and Ranging). It is based on laser technology and eliminates most of the common disadvantages of cameras.

5.4.2 The LIDAR principles and pulse reflection recognition

The principle behind LIDAR is quite simple. Shine a small light at a surface and measure the time it takes to return to its source. When you shine a torch on a surface what you are actually seeing, the light reflects and returns to your retina. Light travels very fast - about 300,000 kilometres per second or 0.3 meters per nanosecond. The equipment required needs to operate extremely fast. Only with the advancements in modern computing technology has this become feasible [41].

The actual calculation to measure how far a returning light photon has travelled to and from an object is quite simple [41]:

$$Distance = \frac{(Speed\ of\ light * Time\ of\ flight\ )}{2} \tag{5.4}$$

The LIDAR sends rapid pulses of laser light at a surface (Figure 5-2, 4), some at up to 150,000 pulses per second. A sensor on the device measures the amount of time (Figure 5-2, 3) it takes for each pulse to bounce back (Figure 5-2, 1). Light moves at a constant and known speed so the LIDAR can calculate the distance between itself and the target with high accuracy. By repeating this in quick succession the device builds up a complex 'map' of the surface scanned.

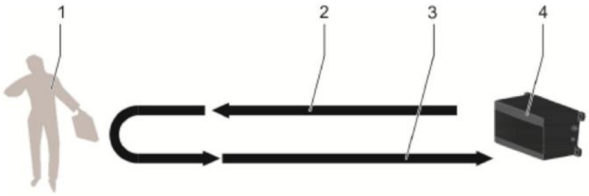


Figure 5-2. 1 Object, 2 Laser pulse, 3 Reflected laser pulse, 4 Laser [42]

Lasers are categorised by their wavelength. 600-1000nm lasers are more commonly used for non-scientific purposes but, as they can be focused and easily absorbed by the eye, the maximum power has to be limited to make them 'eye-safe'. Lasers with a wavelength of 1550nm are a common alternative as they are not focused by the eye and are 'eye-safe', able to be used at higher power levels [41].

For a LIDAR sensor mounted on vehicles, it is necessary to determine the absolute position and orientation of the sensor to retain useable data. Global Positioning Systems provide accurate geographical information regarding the position of the sensor and an Inertia Measurement Unit (IMU) records the precise orientation of the sensor at that location. These two devices provide the method for translating sensor data into static points for use in a variety of systems [41].

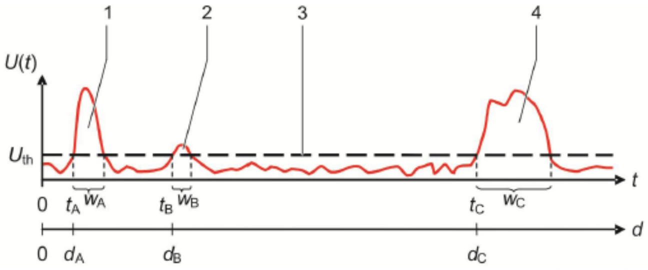


Figure 5-3. Laser echo [42]

5.	Reflection windshield	t	Time
6.	Reflection rain drop	d	Distance
7.	Threshold Voltage Uth	W	Pulse wide
8.	Object reflection	A	Windshield
C	Object	B	Raindrop

LIDAR is becoming more important as an environmental recognition sensor for highly automated vehicles. The speed and accuracy of a scanner allows more or less real-time applications. This enables the device to detect obstacles and recognize its route at very short time intervals. In particular at research level the near range weather situation around a vehicle can be captured.

Within Figure 5-3 the pulse 4 indicates the reflection of an object on the trajectory of a vehicle at a certain sensible distance. The voltage U_{th} indicates the noise threshold, a signal levels above the threshold indicating an object reflection.

5.4.3 Object detection out of the pulse reflection cloud

For the research vehicle the LIDAR system IBEO-LUX was used. The system produces as its output a stream of range and angle tuples; this data is projected into the local coordinate system using the vehicle's position in the local coordinate system (continuously updated as the vehicle moves) and the sensor's position in the vehicle's coordinate system (determined off-line) [43]. The result is a stream of 3D points in the local coordinate frame (Figure 5-4), where all subsequent sensor fusion takes place.

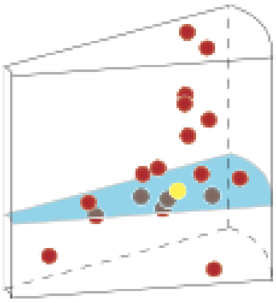


Figure 5-4. Scanned weather objects within a polar grid [42]

The LIDAR returns detected objects marked as ground, dirt, rain, snow, spray and fog. The first phase of the data processing is to classify each return as “ground”, “dirt”, or “rain/spray/fog”. The first step is to filter out the stream of points that correspond only to weather objects (removing ground and outliers). By accomplishing this step, the high resolution of LIDAR data enables us to implement a sophisticated weather classifier. This is essential for our work, which set out to develop a strategy for identifying objects that are likely to be *remarkable* weather objects; those of immediate importance for driving condition recognition. Other points in the cloud that are far enough above the ground model (and satisfy other criteria designed to reject outliers) are outputted as weather detections.

Particle filters such as Kalman or Gabor filters have been the subject of research and verification processes in our project. They are used to classify and quantity the examined weather type objects. Given the non-conformity of the airborne moisture laden entities being detected it is suggested that allometric scaling algorithms may be useful for analysis of cloud/fog/rain/snow/hail/haze and mist entities, especially from visual on-board instruments located in moving vehicles. These are investigated for inclusion in the set of techniques potentially applicable to the domain of research described in this paper. Not only topographic characterization for entity shape is applicable in this domain, with LIDAR bathymetric methods also being considered for the useful provision of detected moisture content.

Both of these geometric recognition and characterization methods are to be the subject of ongoing investigation in the work.

5.4.4 Detection of rain, snow and fog

During weather effects such as fog, rain, and snow, the backscatter leads to the perception of an atmospheric veil. We therefore look at the LIDAR reflected signal power and the backscatter coefficient $\langle\beta(R)\rangle$.

The power P_r received by the LIDAR is defined by the LIDAR equation (Measures, 1984) [44]

$$P_r(R) = E_l \frac{c\langle\beta(R)\rangle}{2} \frac{A_r}{R^2} \tau_T \tau_R exp\left(-2\int_0^R \alpha(R')dR'\right) \tag{5.5}$$

where El is the output laser pulse energy, c the speed of light, $\beta(R)$ the mean backscattering coefficient of the medium, A_r the effective receiver area, τ_T and τ_R the transmitter and receiver efficiencies (the latter is mostly defined by a special filter confining the bandwidth), and α is the extinction coefficient [45].

The backscattering coefficient $\langle\beta(R)\rangle$ is an average over the area illuminated by the laser beam as well as along the line-of-sight within the range $ct_p /2$, where t_p is the laser pulse duration. The extinction coefficient α is controlled by the aerosol scattering factor [44].

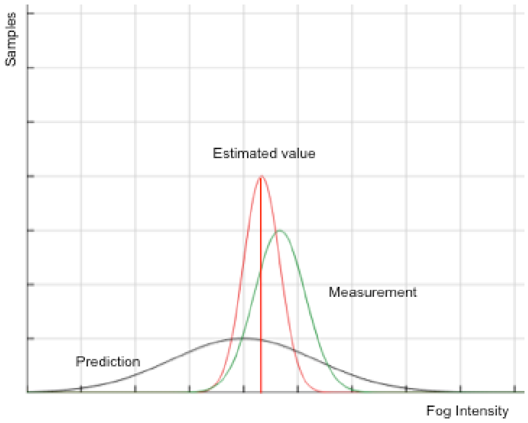


Figure 5-5. Parameter intensity [46]

For object and obstacle detection for the purpose of adaptive cruise control (ACC) or highly automated driving the weather influence is filtered out (resulting as backscatter and power of the received LIDAR reflection). In order to obtain qualified weather object information the LIDAR reflections have to be particle filtered via Kalman filters. A particular focus is placed on the backscatter coefficient *ß* for variation of the filter.

Estimates for fog intensity are statistically calculated using a Kalman filter algorithm in an ensemble of error correcting procedures to produce a most probable estimate value for fog intensity. This method has become known as the *Ensemble Kalman Filter (EnKF)* method. An estimate value is provided to the algorithm as an initial *prior* that is sensitized by intervention values from the sensor data to provide a *posterior* value, which recursively becomes the *new prior* (estimate) for the next iteration of the filter. The Kalman filter correlates the historic data of previous sample data with the value of the new prior, producing an error value (the corrected most probable value), which becomes the estimate, or estimated intensity value as illustrated in Figure 5-5.

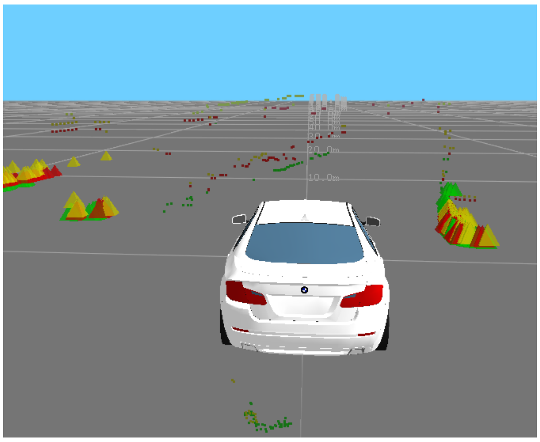


Figure 5-6. Detection scenario

A set of filters was examined under lab conditions to determine weather condition information from the vehicle’s trajectory when in a 200m range of known objects. The set of weather condition data comprises:

- Rain
- Snow
- Fog, dust, mist, haze

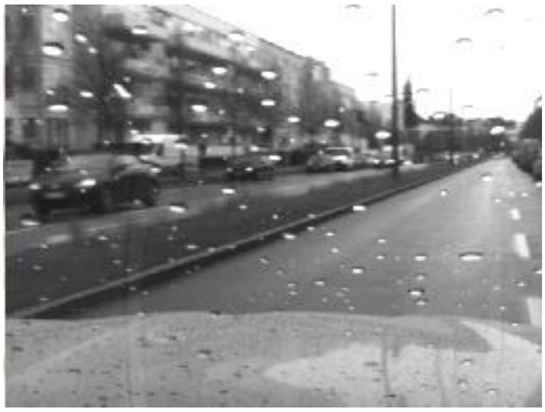
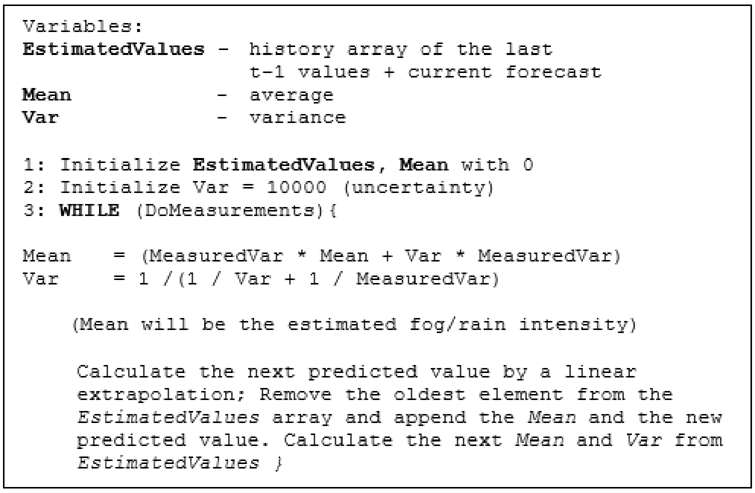


Figure 5-7. Front camera

Figure 5-6 shows the LIDAR-based rain detection scenario on the research vehicle. The associated camera image is illustrated in Figure 5-7. The object and obstacle detection is outlined. As result of the weather depiction, the LIDAR point cloud around the rain object recognition can be seen. A Kalman filter was utilized throughout implementation. The pseudo-code in Figure 5-8 visualizes the parameterization for weather detection purposes.



```
Variables:
EstimatedValues -  history array of the last
                   t-1 values + current forecast
Mean            -  average
Var             -  variance

1: Initialize EstimatedValues, Mean with 0
2: Initialize Var = 10000 (uncertainty)
3: WHILE (DoMeasurements){

Mean    = (MeasuredVar * Mean + Var * MeasuredVar)
Var     = 1 /(1 / Var + 1 / MeasuredVar)

    (Mean will be the estimated fog/rain intensity)

    Calculate the next predicted value by a linear
    extrapolation; Remove the oldest element from the
    EstimatedValues array and append the Mean and the new
    predicted value. Calculate the next Mean and Var from
    EstimatedValues }
```

Figure 5-8. Kalman filter settings

5.5 Sensor fusion, data fusion

“Sensor fusion is a formal framework that comprises methods and tools for the association of data coming from different sensory sources. It tries to win information of high quality, where the specific definition of high quality varies also from one application to another” (Wald) [47].

“Data fusion is the process of combining data or information to estimate or predict entity states“ (Steinberg) [48, 49].

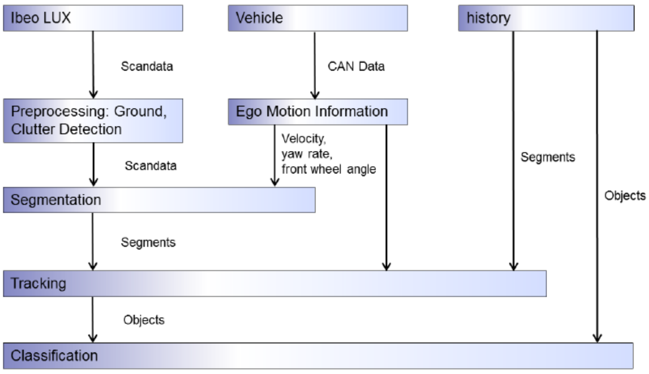


Figure 5-9. Sensor fusion for gaining weather information [42]

As described earlier within the document, different sensors deliver data for ACC or automated driving. Discussion of a lane-keeping function demonstrates that there are essential sensors needed for this such as the following in order to obtain sufficient information about the specific position and movement needed to control the vehicle on its planned trajectory:

- Camera
- LIDAR
- Position (GPS)
- Velocity
- Inertial Measurement Unit (IMA)

To combine this spectrum of sensor data, a fusion model is introduced (see Figure 5-9). Vertical segmentation covers the different sensor types and sources, horizontal segmentation is applied for functional composition and logical grouping.

In case of weather recognition using LIDAR, camera and ego, motion is a primary value used in the weather perception equation. Historical data of these sources is needed to consider the time variable element for the equation. The perception model developed to deliver precise weather data needs historical data for tendency handling.

Figure 5-10 illustrates the sensor data fusion process for LIDAR sensor data generation (data collection) through a process that determines the quality and quantity of that data. These so-called *reflections* are visualized. The final applicable LIDAR point clouds appear as *Type A* assignments. They have been classified by their main characteristic hypothesis (MH) and their object, obstacle or weather object hypothesis (OH) [11].

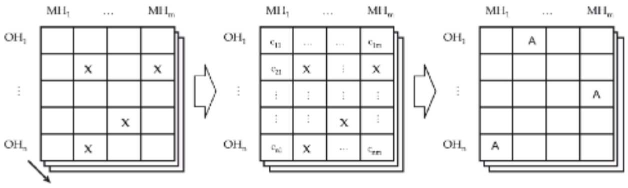


Figure 5-10. Sensor Data Fusion [50]

5.6 Evaluation

The evaluation was carried out using a BMW research car equipped with full instrumentation for autonomous driving. The car is illustrated in Figure 3-1.

Arrows mark the main sensors; in particular the LIDAR scanners are highlighted as lasers. For the different types of weather conditions, particular test runs were executed. Input and output were captured for further studies to optimize the recognition algorithms. The outcome of the near-field weather recognition is valuable driving security information for the driver. Additionally, such weather data can be correlated with the current geo position (map location) and submitted to telematics services or other collaborative traffic systems. On board the vehicle itself, many autonomous driving functions can subscribe to this weather information. Thus the adaptive cruise control (ACC) can automatically reduce the desired speed if weather conditions deteriorate. The same applies to highly automated driving. The speed setting or the time to handover of the driving task to the driver can also be adjusted automatically.

5.7 Summary

Increasingly, the automotive industry is developing and producing innovative driving assistance functions to support vehicle drivers. Well understood as being a very important goal is that of improving driving security. In this context it is an essential functionality to capture local weather in the vicinity of the car. Many dependent functions within a vehicle can benefit from subscribing to an automated, reliable weather source. Focusing on the collaborative aspects of traffic systems as they exist today and as planned for the future will accelerate the process of producing safe and reliable autonomous driving systems. It is clear that the detection of weather (and thus road conditions) is a principal element and data source in this quest. Sharing information from vehicles with the sensors described in this paper is regarded as an intrinsic benefit to driver safety and so is being investigated for viability as part of our research work. Cloud-based technologies are thought to be key in resolving the data communication aspects of this scenario, with traffic management and weather information systems interacting at the data level in a cloud technology space. In our scenario, the data and software in this source would provide both driver information and vehicle control system data relating to vehicle position on the road, speed and road conditions, supplemented by weather condition information.

6 Traffic messages and their addressing and routing

6.1 Location based addressing, routing and broadcasting

6.1.1 Motivation

"The vehicle as a mobile sensor in a vehicular network" requires large-scale mobile networks with particular requirements in terms of real-time availability, performance and addressability of traffic members and their infrastructure. For this reason, the concepts for message routing and addressing within LTE networks described in the upcoming dissertation “Collaborative Positioning” [51], will be used and discussed here.

Pure Vehicular Ad Hoc Networks such as ITS-G5 become problematic when the network gets scattered. Utilizing LTE, a mobile network with a very low latency, could improve the situation where the number of vehicles acting as an ad hoc network communication node is insufficient. This research presents an addressing mechanism based on IPv6-Multicast and LTE which allows position-dependent and dynamic addressing. The initial geographical position of a message is directly incorporated into the IPv6 multicast address. This allows not only effective prefiltering on the receiving end but also routing in the backbone [51, 52].

6.1.2 Introduction

6.1.2.1 LTE as an addition to ad hoc networks

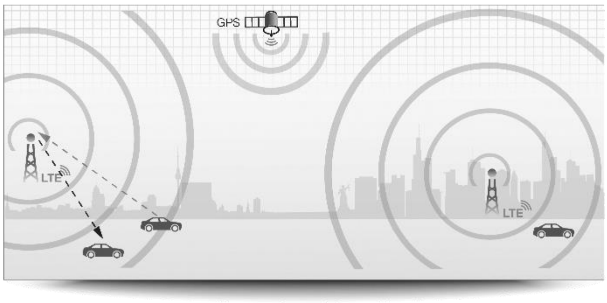


Figure 6-1. LTE

Pure vehicular ad hoc networks (VANET: Vehicular Ad Hoc Network) [17], e.g. based on the 802.11p standard, raise a number of issues [53]: due to their limited communication range, VANETs become problematic in case of an insufficient density of vehicles acting as a communication node. In [54] this issue was addressed by adding a satellite link. The well-established and tested LTE technology constitutes an interesting addition that improves the information propagation in scattered ad hoc networks.

LTE is an evolution of the mobile technology standards of the 3rd Generation (3G) and represents a quantum leap over previous generations GSM and UMTS. Some of the most remarkable advantages and their positive effects in taking the vehicle into account as a mobile sensor” are [55] as follows:

- Low latency → the vehicle sensors are connected in real time to the cloud or sharing data with the ambient traffic.
- Data transfer rates in the UL/DL (50/100 Mbit/s) → map data updates can be applied to the cloud based map database. Object data for high precise positioning can be made thus available to the autonomous driving community.
- Potentially lower power consumption → always essential for on board applications to save electrical energy consumption.
- Pure data-packet-based infrastructure for the IP protocol→ simplifies the applications protocol stack interfaces.
- High mobility, i.e. stable transmission even at high velocity → essential for precise ADAS functionality within a collaborative network.
- Widespread introduction in Europe is planned → even worldwide, the built upon ASAS systems will be country interoperable.
- Further scalable and enhance able in the future (LTE Advanced, QoS) → very important for classification of the real time demands of different applications.

From 2015 onwards, due to the eCall [56] standard of the European Union, installation of GPS and cellular network modules are mandatory in every newly manufactured car. Due to the characteristics of LTE, the technical base for Car-to-Car Communication (C2C) would be greatly simplified. In particular by treating the vehicle as a member of "internet of things", the vehicle's full integration supports an even stronger the focus on driving security and emission reduction aspects by instant information's in the traffic cloud systems. The corresponding data analysis and mining processes will thus work on real time measures for optimization of the individual goals.

Car-to-Car Communication (C2C) means the direct exchange of information messages between vehicles. Based on C2C, new applications such as cooperative traffic safety assistant can be realized. These include just-in-time information on road and traffic conditions such as black ice, aquaplaning, accidents, or broken-down vehicles [57]. In this safety-related application, the affected vehicle reports the information along with the exact road hazard location to other road users nearby [58]. It should be noted in this case that the vehicle steps into the role of acting "as a mobile sensor in his vehicular network". The limited range of the physical transport medium provides the position-dependent delivery of relevant messages in ad hoc networks. In the case of LTE this can be achieved by geo-based IPv6 routing.

This document discusses a feasible procedure for the generation of IPv6 multicast addresses with the help of location data.

6.1.2.2 Geo-based IPv6 multicast (group-addressing)

Geo-based multicast refers to the location-dependent addressing of multiple network nodes. In particular IPv6 offers a sufficiently large address space to encode geographical coordinates directly into the IPv6 address. The geographical position is used for the dynamic generation of multicast addresses [59] of a specific multicast group. The sender generates the multicast address using its own current geographical position. Dependent on their own geographical position, the recipients that belong to this specific multicast group can then quickly and effectively decide on the relevance and further processing of the received message.

6.1.3 Geo-based IPv6 multicast addresses

As the basis for position determination we use the widespread geodetic reference system World Geodetic System 1984 (WGS 84). WGS 84 is the reference coordinate system for GPS and was accepted by ICAO resolution for use in aviation.

6.1.3.1 IPv6 multicast addresses

IPv6 addresses consist of 128 bits [59] and are mainly divided in 2 parts: for multicast this is the prefix ff00::/8 followed by an up to 112-bit multicast group ID.

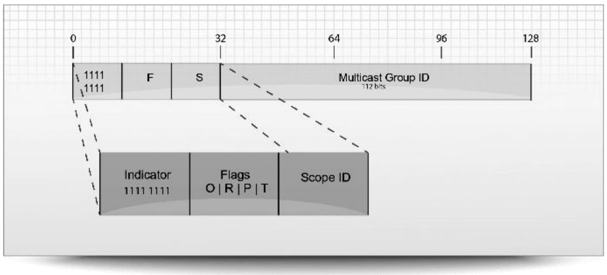


Figure 6-2. Multicast group header

The flags determine the type of address. The range is determined by the scope ID. Permanent multicast addresses and address ranges can be requested from the IANA (Internet Assigned Numbers Authority).

IANA allocates permanent multicast group IDs consisting of 32 bit. The T-Bit ("transient") and the P-bit ("unicast based") in multicast addresses assigned by IANA is always 0.

The last bit of the multicast group ID is also the last official and fixed part of the IPv6 address. The following is our proposal for encoding the geographical coordinates.

6.1.3.2 Encoding of the geographical position

Longitude and latitude are encoded with a 24-bit resolution. The resolution achieved is therefore more accurate than that of GPS receivers available today. Defining the WGS-84 equatorial radius a: 6378137.0 m, a 24-bit resolution will generate a grid on equator level with the maximum edge length of 2.39 m.

For encoding, the coordinates of northern latitude and eastern longitude are used with positive signs. The coordinates of southern latitude and western longitude however are used with negative signs. The latitude thus ranges between -90 ° and +90 °, the longitude between -180 ° and +180 °.

The coordinates are then converted to integers of the selected 24-bit resolution by multiplying by:

$$\frac{2^{24}}{180} \text{ for latitude} \tag{6.1}$$

$$\frac{2^{24}}{360} \text{ for longitude} \tag{6.2}$$

Through the addition of a so-called bias of 2^{23} these re stored in "offset binary" format in the respective fields [9]. This simplifies the comparison of the values because no explicit bit containing a sign needs to be considered.

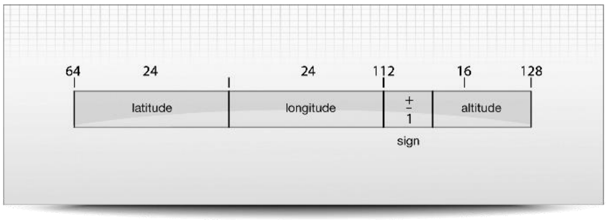


Figure 6-3. Geographical position

6.1.3.2.1 Filtering according to distance (grouping)

In C2C environments warnings accumulate inherently. Just think of the flood of automatically generated warnings by vehicles approaching a tailback. An effective and simple process for estimating the distance is therefore necessary in order recognize and filter out non-relevant (i.e. far distant) messages. The process needs to be simple and resource-efficient to implement. We would like to discuss the following proposal:

To roughly estimate the distance between two positions [B_1, L_1] and [B_2, L_2] that have to be encoded, we perform the following simplified calculation:

$$|B_1 - B_2| + |L_1 - L_2| = a \tag{6.3}$$

For the actual distance d between the positions, the inequality applies:

$$\frac{1}{2}a \leq d\frac{2^{24}}{180} \leq a \tag{6.4}$$

This resembles the estimation between Manhattan distance and Euclidean distance in the plane, only in this case on the surface of a sphere and at a certain distance to the poles. Distance is smaller than the comparative distance c if:

$$a \leq c\frac{2^{24}}{180} \tag{6.5}$$

6.1.3.2.2 Messages type ID (designed for traffic events)

To clearly and positively identify the message type, 16 bits are used. This means that each message can be assigned to a specific type. In addition to warnings it comprises general information about weather, traffic flow, road conditions, event notifications and parking occupancy. Thus easy filtering according to categories is possible for both receiver and router.

6.1.3.2.3 Complete geo-based IPv6 multicast address

In summary, the proposed multicast IPv6 header now appears as shown below:

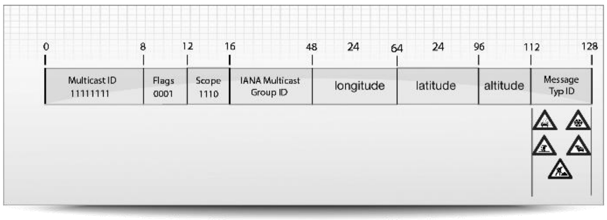


Figure 6-4. IPv6 header

6.1.3.2.4 Multicast routing (group-addressing)

Routers decide exclusively on information taken from the IP header as to which path to select to send a data packet. To process the geo-based IPv6 packets outlined above, it is not necessary for the router to have a basic understanding of geography. The decision whether or not to pass on a multicast packet can be broken down to the decision as to whether the calculated distance a to a given position $[B_2, L_2]$ is less than the predetermined distance c for filtering.

6.1.3.3 Conclusion

The procedure presented here for the generation of IPv6 multicast addresses allows the sending of messages limited to a defined geographical area. Based on the position indicated in the IPv6 multicast header, the area is defined autonomously both by the receiver and by the connected router. The encoding rule used is simple and easy to implement in the hardware.

The expansion of the mobile network LTE is being promoted and will cover the major transport routes relatively quickly within the next few years. Due to its bandwidth and its round-trip delay, LTE is also a promising candidate in terms of meeting time-critical requirements in C2C. As the required GPS and wireless modules are being employed in vehicles already and LTE is a field-tested technology as well, this solution will become increasingly interesting commercially.

The warnings can be centrally collected in a service centre and supplemented by existing information such as floating car data, traffic situation systems, etc. Thus the first vehicle to be equipped with this technology will benefit from it, without any additional expensive roadside unit infrastructure or market penetration. In particular the message addressing and routing technique researched and described here provides the opportunity to treat vehicular traffic as a "traffic sensor society" delivering real-time sensor data of different types over a wide range of map coverage. The potential further applications being researched will be introduced later in this thesis.

6.2 Weather services within the cloud using TMP

6.2.1 Motivation

Section 6.1 describes the idea of a native geo-based message addressing and IP based routing to and from vehicles in ideal terms. The underlying IPv6 infrastructure in terms of the described protocol extension is still not available. The process of obtaining such a request supported by a large traffic community of users will last for some years.

In order to use a geo-based addressing scheme for the purposes of this thesis, the Team Messaging Protocol (TMP) will be employed. This can be considered as a temporary solution until the native geo-based message addressing protocol is available. See Figure 6-6. TMP was designed within the TEAM[4] research project [20].

6.2.2 TEAM Messaging Protocol (TMP)

TMP (TEAM Messaging Protocol) is a communication protocol that enables the creation of opportunistic multicast groups between mobile and fixed traffic actors in a given geographic area. TMP is based on a client-server architecture where the server is aware of the geo-location of clients in its coverage zone (see Figure 6-5). The TMP client infrastructure hosted by the mobile traffic actor allows applications to exchange collaborative application messages and data. Simultaneously, information about the actor’s geo-location and its characteristics (type of vehicle, TMP-enabled applications, etc.) are periodically uploaded to the server, thus enabling effective mechanisms for actor, components and services discovery. Upon an actor’s request for instance, the server can create a multicast group of actors fulfilling given membership cri-

[4] TEAM: Tomorrows elastic adaptive mobility, https://www.collaborative-team.eu/

teria. These criteria are derived from a filter that specifies actor characteristics such as geographic range and location, actor type, actor is on the same street, actor is moving in the same or in the opposite direction, etc. TMP counts with a variety of options to create opportunistic groups, effectively providing powerful mechanisms to anonymously address traffic actors in a given geographic area and sharing a particular set of attributes. TMP was designed within the TEAM research project [60]

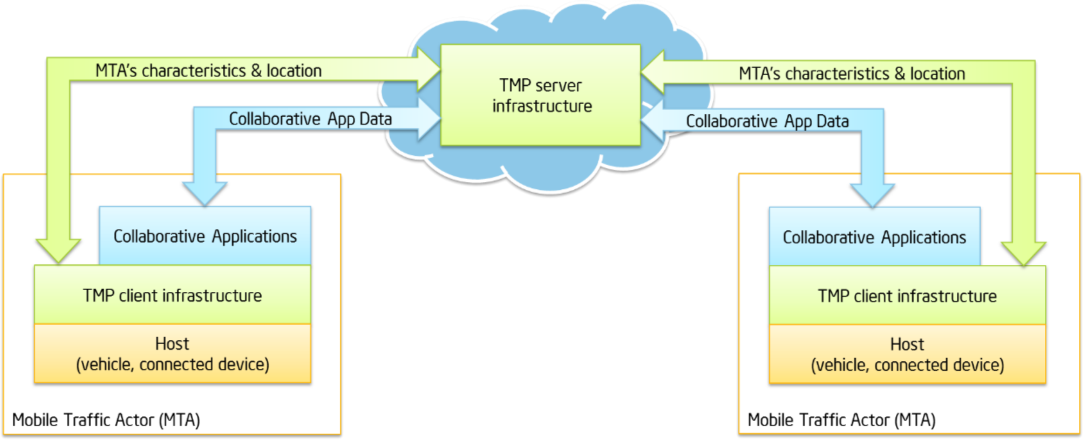

MTA's characteristics & location
Collaborative App Data
TMP server infrastructure
MTA's characteristics & location
Collaborative App Data
Collaborative Applications
TMP client infrastructure
Host (vehicle, connected device)
Mobile Traffic Actor (MTA)
Collaborative Applications
TMP client infrastructure
Host (vehicle, connected device)
Mobile Traffic Actor (MTA)

Figure 6-5. TMP is based on a client-server architecture, where the server is aware of the physical location of clients in its coverage zone

The TMP client API delivers a set of rich functions and event handlers to create collaborative applications that interact with other actors and the environment in a transparent and straightforward manner.

TMP makes use of so-called Local Access Servers (LAS) in order to provide optimized latency and data throughput within the server's corresponding coverage zone. Being a TCP/IP-based protocol, the underlying transport can be kept flexible. Currently TMP runs on 3G/LTE networks, delivering optimum results in regard to latency on a widespread mobile technology. However, other TCP/IP-based communication technologies can be used as well.

User privacy plays a central role in TMP: actor IDs are periodically renewed and all actor information is deleted from the server, so that neither traceability nor identification are possible. All exchanges are secure as well, providing fully encrypted communications and no direct interaction among users or between users and the server. TMP implements a self-managed security layer built upon the transport protocol (MQTT[5]) and is thus independent of all security features of the underlying logical and physical layers. TMP uses currently the Mosquitto

[5] http://mqtt.org, machine-to-machine (M2M) / "Internet of Things" connectivity protocol

MQTT[6] broker, which runs as a background application in the ITS subsystem. Some examples of dynamic group addressing are shown in Figure 6-6.

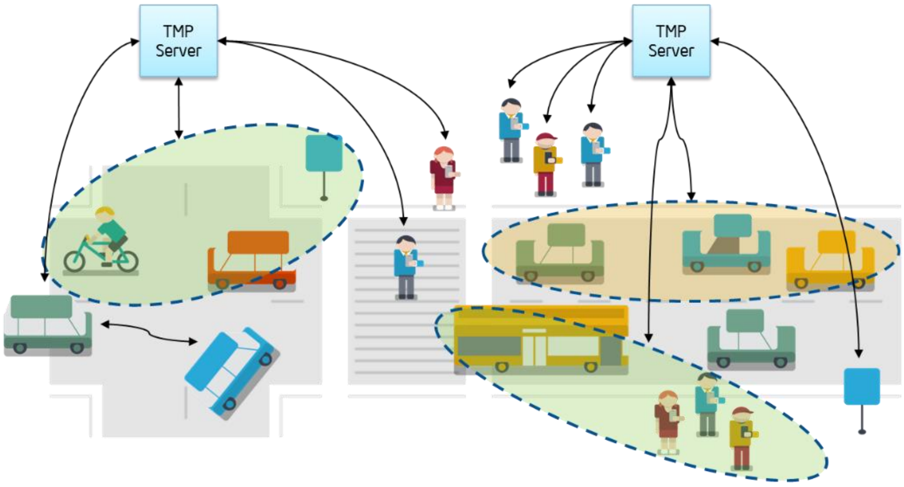


Figure 6-6. Examples of selective, heterogeneous, geo-specific, opportunistic groups enabled by TMP

6.2.3 Geo-based IPv6 multicast

Geo-based multicast refers to the location-dependent addressing of multiple network nodes. In particular IPv6 offers a sufficiently large address space to encode geographical coordinates directly into the IPv6 address. The geographical position is used for the dynamic generation of multicast addresses of a specific multicast group (see Figure 6-7). The sender generates the multicast address using its own current geographical position. Dependent on their own geographical position, recipients who belong to this specific multicast group can then quickly and effectively decide on the relevance and further processing of the message received.

Geo-based IPv6 multicast uses the multicast address space in the IPv6 packet header to encode geo-position and height information in combination with a message type ID for fast IPv6 packet identification and filtering.

LTE and other widespread cellular technologies constitute an optimum infrastructure for the transport of TCP/IP-based data, which geo-based IPv6 multicast can leverage to address vehicles with greater precision as that achieved by GPS. LTE and future cellular technologies offer the added advantage of low latency and large data throughputs as well as extensive geographical coverage, rendering the use of dedicated hardware for C2X communications deployed in the field, such e.g. roadside units, unnecessary.

[6] http://mosquitto.org, message queuing telemetry transport protocol

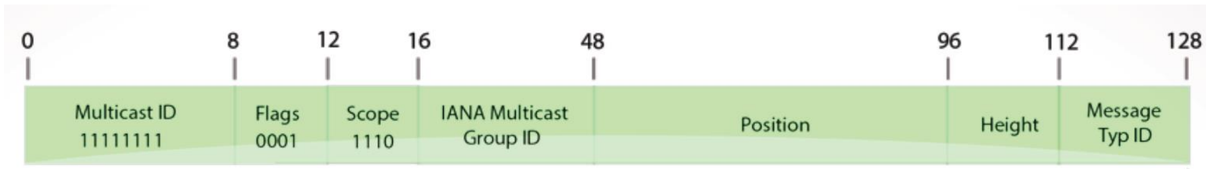


Figure 6-7. Geo-based IPv6 multicast header including geo-position and message type ID

6.2.4 Combining TMP and geo-based IPv6 multicast communications

Geo-based IPv6 solves the problem of geographical addressing and message delivery, whereas TMP provides the necessary logical infrastructure for a collaborative framework based on the creation of opportunistic groups. In this context, TMP uses IPv6 as the transport and distribution (addressing) layer over an appropriate IP-based physical communication technology such as LTE. This combination has the potential of eliminating TMP's server-side architecture as shown in Figure 6-8, at the cost of increasing traffic and logical complexity on the client side. With the appropriate configuration, network routers can deliver IPv6 packets to all actors in a given geographical zone, which in the TMP context corresponds to the Local Access Server coverage zone. Upon reception of an IPv6 packet containing a TMP message, the client decides with the help of the IPv6 header's ID type whether to discard or further process the message payload, which contains the TMP message itself. The client-side TMP infrastructure then analyses the TMP header and decides whether the TMP message is relevant for the actor or not.

Some changes to the current TMP protocol are necessary in order to support the distributed nature, i.e. non-centralized character, of the TMP LAS coverage zone now incorporated in the geo-based IPv6 addressing scheme, see Figure 6-8. For instance, an efficient mechanism for the creation of groups based on vehicle characteristics and their current position has to be devised, as well as the distribution mechanism of this information to all actors within the IPv6 addressing zone (compare Figure 6-5 and Figure 6-8).

Alternatively, suitable mechanisms within the geo-based IPv6 multicast addressing scheme could be introduced in order to address areas with configurable radius. For instance, instead of encoding height information, a radius around the geolocation specified by the longitude-latitude coordinates could be easily leveraged by TMP to create opportunistic groups, or any other application wishing to address all actors within a specific area. This would allow TMP actors within a group to interact with each other, using group membership to filter out existing actors in the same geographical zone but which should not belong to the group.

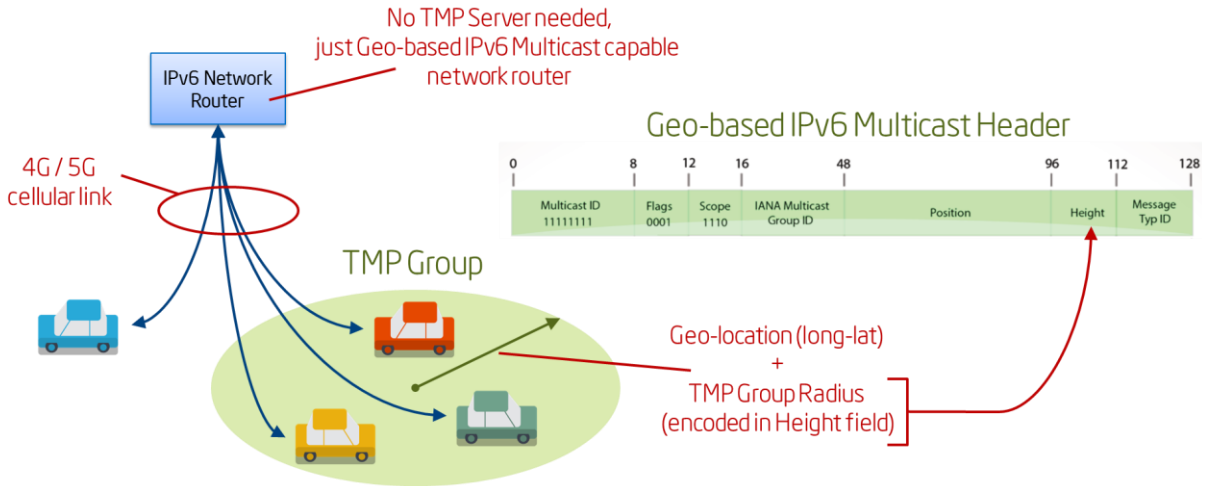


Figure 6-8. TMP can benefit from geo-based IPv6 multicast addressing simplifying the server-side infrastructure and by directly geo-addressing TMP groups

7 Massive Online Analysis (MOA) of the vehicular sensor data

As mentioned at the very beginning of this document, there are several ways to insert the weather conditions into our Drive-Guard system. In one of them, the vehicle recognizes the weather conditions of the environment with the help of sensors and relays it directly to our cloud-based server. Here, the determination of the weather conditions through the environment's measurement remains questionable and error-prone. This process should happen in a centralized place for the system and not in each vehicle separately in order to keep the process consistent. Therefore, the above-mentioned task should be split in two steps:

1. The vehicle sends appropriate environment measurements to the server
2. A job runs at specific time intervals and transforms the recently received information into a weather condition.

Regarding (1): The vehicle is capable of capturing some environmental parameters such as ambient temperature and rain intensity. This is implemented via an ADASRP (**A**dvanced **D**river **A**ssistance **S**ystem **R**esearch **P**latform) plugin. This plugin is very simple and straightforwardly follows the common structure described in the next section, so it will not be described in detail here. Its task consists in the extracting the two values (ambient temperature and rain intensity) from the CAN bus and sending them back depending on a specific period of time to our server via a simple HTTP POST request. The only thing to note is that the sending of the POST HTTP request is carried out with the help of boost::asio C++ library:

```
bool CBMWMOA::sendHTTPPost(CString url)
{
    boost::system::error_code ec;
       using namespace boost::asio;

       io_service svc;
       ip::tcp::socket sock(svc);
       sock.connect(ip::tcp::endpoint(ip::address_v4::from_string(DEFAULT_HOST),
       DEFAULT_PORT)); // http://drive-guard.net:55554

       // send request
       std::string request("POST " + url + " HTTP/1.1\r\n\r\n");
       sock.send(buffer(request));

       // read response
       std::string response;

       char buf[1024];
       size_t bytes_transferred = sock.receive(buffer(buf));
       response.append(buf, buf + bytes_transferred);
       sock.close();

       return response.find("HTTP/1.1 200 OK") != -1;
}
```

For each type of environment measurement the car can send, there is a responsible function in the server which accepts the respective ambient temperature's or rain intensity's request. These functions are declared in a**n MOA Controller** and their request mappings (API calls) are described as usual in the conf/routes file:

```
#--------------------- MOA API -------------------------
POST    /ambient_temperature
      controllers.MOAController.MOAAmbientTemperature(latitude: Double, longitude:
      Double, carId: String, temp: Double)
POST    /rain_intensity
      controllers.MOAController.MOARainIntensity(latitude: Double, longitude:
      Double, carId: String, intensity: Double)
```

The API is simply to be tested with a curl request like this (data part can be omitted as well):

```
Curl --request POST 'http://drive–guard.com:55554/rain_intensity?latitude=11.55&longitude=48.35&carId=FA14&intensity=3.4' -data ''
```

Please note that the vehicle is not able to transmit the type of the environment parameter in the request because it is necessary to first decide how to handle the different parameters in combination and therefore clients (vehicles) ought to be restricted in sending the data that they know/have at disposal. The client is restricted to send only the parameters with meaningful semantics. Also, an important aspect of the implementation of the MOA functionality is the lack of a generic function (god/master function) on the server side, handling all HTTP POST requests at once. Instead, there is a separate function for each request. The reason for this is that such a god function should contain a lot of boiler plate code representing the common IF-ELSE logic. In this particular case, it is avoided by using "separation of concerns" at the highest possible level of the system (the presentation level). This makes the code more robust, readable and easily maintainable.

The receiving of MOA data can be occasionally disabled or enabled. Since the MOA service runs in the web container and has no separately dedicated server, the server's behaviour is imitated with the help of a static Boolean flag. This flag is used later on in the interceptor[7] in order to determine whether the access to the whole MOA Controller has to be blocked. Further information regarding the interceptors can be obtained from the Play! Framework 2.X documentation. The following code snipped illustrates the function of the MOA controller:

```
public Action onRequest(Request request, Method actionMethod) {
   if( actionMethod.getDeclaringClass().equals(MOAController.class) ) {
      if(!controllers.Application.MOAServerRunning) {
```

[7] https://www.playframework.com/documentation/2.0/JavaInterceptors

```
            return new Action() {
               @Override
               public Result call(Context ctx) throws Throwable {
                  return badRequest();
               }
            };
         }
      }
      return super.onRequest(request, actionMethod);
}
```

Regarding (2): Suppose there are multiple sources of information about the weather/environmental conditions. It is clear that a sufficiently large cluster of such measurements is necessary, located close to each other, in order to make a reasonable conclusion about the weather conditions in that area. On the contrary, a small number of such environmental measurements can be considered insufficient for a meaningful prediction/analysis. In Figure 7-1 a) a planar surface with multiple environment measurements is illustrated, whereby each point represents the ambient temperature and the rain intensity values. The circles are the cluster of environmental measurements from which the weather conditions can be determined. Therefore, each of these groups will result in only a single weather condition of type **models.DriveGuardMessage**. In the particular situation on the image, three clustered **DriveGuardMessages** can be determined.

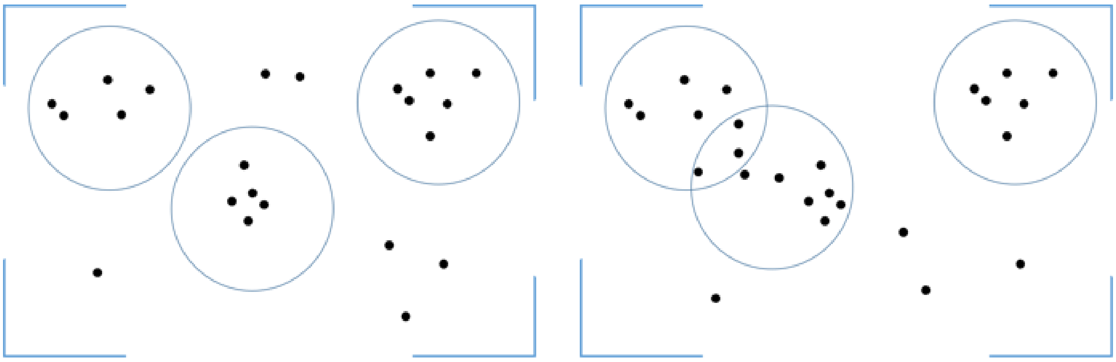


Figure 7-1. a, b - Conclusion of road weather in the area

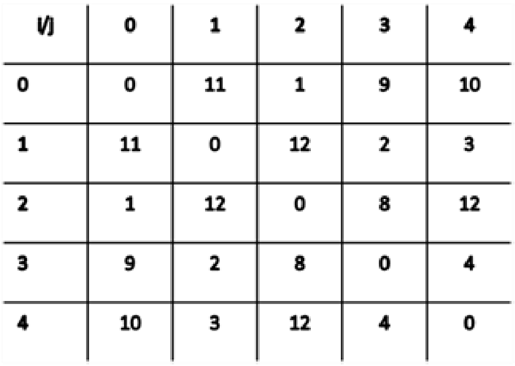
The naive determination of such groups of measurements (Figure 7-3 a) is a NP hard problem [61, 62], because it requires the consideration of every possible subset of such measurement and checking whether this subset matches a circle or not (complexity is O(2^n)) which in general is an infeasible solution, since there can be thousands of measurements. Therefore, a more sophisticated algorithm is required. In the following, a very simple algorithm is proposed and it is NOT assumed to be the fastest possible BUT it is good enough as a first step to artificially determine such groups/clusters in a feasible time. Care also has to be taken to ensure that the groups of points may overlap as shown in Figure 7-3 b). The solution is as follows:

In a sufficiently large-sized group, there is at least one element located approximately in the middle of this group. Therefore, this element can be considered as the centre of the circle in which the group fits. Despite not fully meeting the theoretical requirement, this idea turns out to be good enough (in terms of memory usage and performance) for practical implementation. Its algorithm consists of the following steps and runs at a worst case complexity of O(n^2) and an exact time complexity of O(n(n-1)/2).

The following algorithm will be used (to visualize the concept the measurement dislocation, illustrated in Figure 7-3 a):

1. Enumerate the points starting from 0. The order of data is one of the most important prerequisites for the application of artificial algorithms (see Figure 7-3 b).

2. Compute the distance matrix M between each 2 points (Figure 7-2). Please note that the matrix is symmetrical, only the part below its main diagonal needs to be computed. M[i][j] is the distance between the point with number *i* and the point with number *j*. Some of the distances are given in Figure 7-3 c).



I/J	0	1	2	3	4
0	0	11	1	9	10
1	11	0	12	2	3
2	1	12	0	8	12
3	9	2	8	0	4
4	10	3	12	4	0

Figure 7-2. Distance Matrix

3. Next, find all groups of points fitting in a circle with diameter D=5 and with size S at least S=2. Here, the trick is that it is highly probable that there is no point that lies exactly in the middle of the group (the so called centroid) and therefore the circle diameter is increased a little in order to compensate the resulting incorrectness. Increasing the diameter length by 25% means that D = 6.25.

   Improvement: The error correction depends on the group size. If there are thousands of points, the diameter length possibly needs to be increased a very little or even not at all, because the chance of a point lying approximately in the middle of the circle increases rapidly with the number of points.

4. Iterate over the rows of the matrix and create a bucket for the points with these indices, for which the distance is less or equal D/2. Actually, the elements in the row R having a distance less or equal D/2 have the semantic that they fit in a circle with a centre being the point with index R. In this particular case, the newly created buckets are the points in each circle from Figure 7-3 d) and namely <1, 3>, <1, 4>, <1, 3, 4>, <0, 2> and <0, 2> again.

5. Retain only the unique subset of points with the sufficient size S. The end result is <1, 3>, <1, 4>, <1, 3, 4>, <0, 2>. To the best of knowledge, there is no algorithm yet known solving the above described problem that is not NP hard.

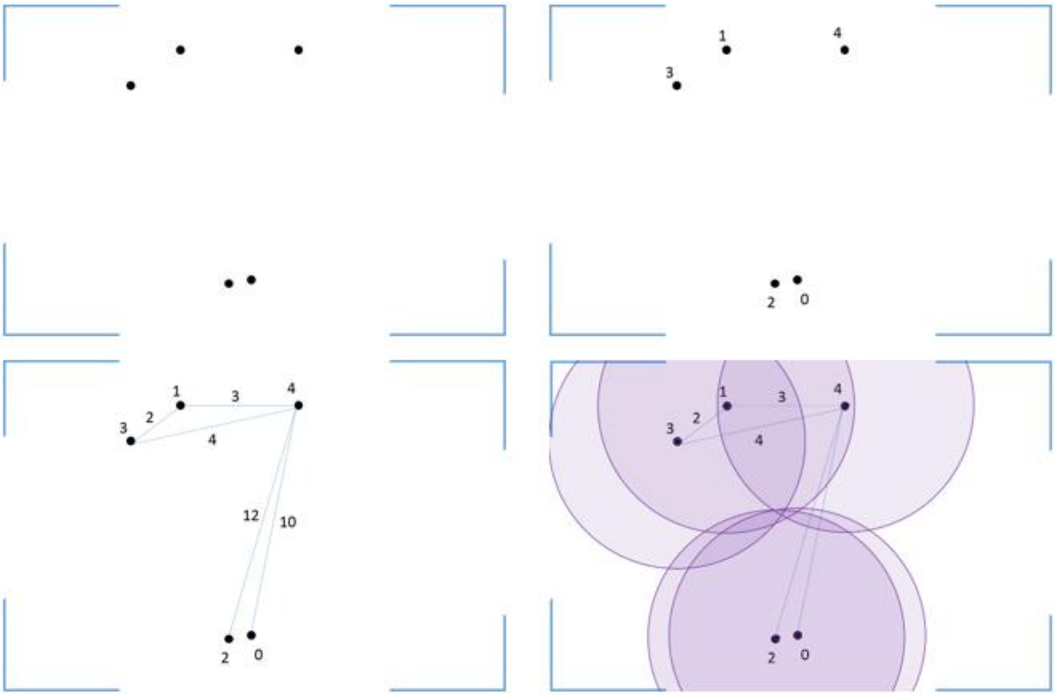


Figure 7-3. a, b, c, d - Road weather events grouping

7.1 Road weather derivation

For this research the road weather derivation was inspired by the conceptual basis of “The Vehicle Data Translator” VDT, published by the National Center for Atmospheric Research (NCAR) U.S. [63]. A three-level model is the basis for the concept of a reliable road weather data source system. The first level is responsible for vehicle sensor data acquisition, the second level for quality checks on the basis of auxiliary weather data sources and the third level for inference data mining to obtain more complex weather information mapped to particular road segments. The three layers are described in detail:

7.1.1 Vehicle sensor data acquisition

Taking into account that a huge number of future vehicles (vehicle-based probes) deliver millions of weather data tokens, it requires new methods and applications (algorithms) to manage and analyse them. A sensor data aggregator has to condense and translate the vast quantities of vehicular probe data into a unified and manageable format. Figure 9-27 illustrates the vehicle capturing a severe weather situation, sending the data to the TM system and propagating the vehicles sensor data into a corresponding spatial-temporal database. A map model is used for the underlying database scheme. The data is tagged with vehicle position, time of observation and direction. The vehicular probe data transmission frequency is dependent on road type (rural, urban, highway), speed, traffic situation and time. It can vary between five transmissions per second until 5 transmissions per minute, data transmission package size is estimated to be about 2 kB of data.

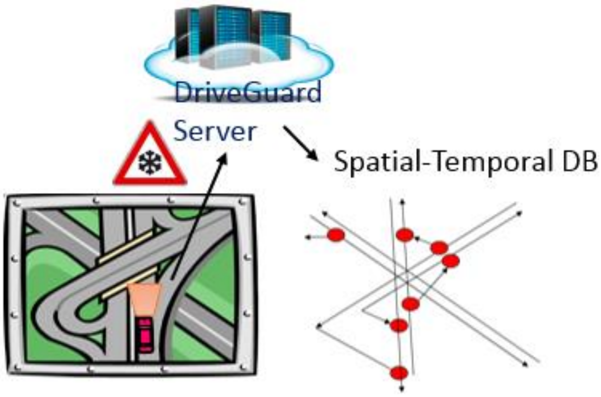


Figure 7-4. Vehicle sensor data acquisition and mapping to spatial-temporal DB

The decision in favour of a spatial-temporal database was deliberately made based on the fact that it includes an integrated access layer for location and time-based database queries. These functions are beneficial for GIS (Geographical Information System) operations such as a query to inquire all reported rain messages in a certain area. The time aspect of such a database systems ensures in this case that only data of a predefined age is obtained. Chapter 7 describes how such queries are designed and how to make a MySQL database combined with a particular framework “spatial” ready. Filtering, validation and qualitative inspection of sensor data is also simplified by spatial-temporal database techniques (see the chapters that follow).

7.1.2 Validation and qualitative inspection

Vehicle sensor data can be quite complex, particular when analysing or deriving near-range and mid-range weather and road conditions. Particular algorithms have to be put in place to filter and verify the sensor data prior to further processing. Filtering is applied for drop-outs of non-plausible sensor data (e.g. daylight status or rain in a tunnel). A subsequent validation of the vehicular weather data obtained is matched against:

- Data coming from ambient traffic.
- Road side unit weather observations.
- Public weather networks.
- Private weather networks (e.g. “Weather Underground”[8]).
- Satellite & radar weather data broadcasts.

The resulting purged and qualitatively improved sensor data will be the basis for the next steps to obtain expressive information about road weather situation and tendencies.

7.1.3 Weather data mining and dissemination to the cloud

The passed over validated and qualified data will be processed to obtain stochastic values for a given location (grid matrix points) and time interval (see Figure 7-3). The weather data mining will output two independent threads:

1. Processed “direct” data. This type of data is generated on basis of stochastic methods and mapped to a road matrix/grid segment (see Figure 7-3). Rain intensity is taken at a certain location on the map grid-matrix. The “direct” category includes weather observations of specific meteorological variables (e.g. daylight, temperature) that should

[8] http://www.wunderground.com/weather/api/

serve the TM as input for weather models and as data at high spatial and temporal resolution weather data for improved weather and hazardous consciousness.

2. Derived “indirect” data. For the analysis of these combined vehicular sensor and auxiliary weather data (coming from other data sources, such as “weather underground”) a considerably more complex algorithm scheme has to be applied. One example is the precipitation rate (derived data) computed from vehicle wiper status, rain sensor, speed, ABS, ESP and lights. The “indirect” category matches GIS information (e.g. time, location) and vehicle system status observations (e.g., windshield wiper status, ABS, ESP), which can be used in conjunction with other datasets to derive significant weather and road conditions [64].

An example how weather events (weather event types) can be obtained as single values or out of a combination of values by applying a decision matrix (containing some examples).

Table 1. Weather event detection rules [64]

Weather Event Type		**Weather Input Data**	
Computed Type	***Internal Code***	***Primary cause***	***Improvement***

Rain	11	Rain sensor readout = moderate rain	Wiper status = mid
			Humidity = high
			Speed = reduced
			Camera = mod. rain
			LIDAR = mod. rain

Heavy rain	12	Rain sensor readout = strong rain	Wiper status = high
			Humidity = high
			Speed = highly reduced
			Camera = severe rain
			LIDAR = severe rain

Icy road	15	Temp. < $3^{0\,C}$ Speed < 50 km/h Activity ABS, ESP	LIDAR = ice, snow

Degrading visibility	20	Speed < 50 km/h
		LIDAR = fog | rain
		Camera = fog | rain
			Lights status = on
			Fog light status = on

The event type “rain” as shown in Table 1 will be issued if the rain sensor read-out is set to “moderate rain” and the auxiliary improvement wiper status is set to “middle”, humidity is “high”, speed is reduced, camera and LIDAR sensor algorithms deliver “moderate rain”. The decision table and its combinatory can be improved based on best practice or even learned through certain approaches of artificial intelligence, machine-blended learning and decision theory (not the subject of this thesis). The theoretical concept introduced here will be researched and validated on the basis of some real live experiments using a research car and through simulation.

8 Vehicles speed profile as a key for efficiency and safety

8.1 Motivation

A stationary vehicle does not reflect any safety or eco-efficiency issues. But conversely, this means the following: as soon as the vehicle is in operation, the fuel consumption depends on such factors as route, traffic, driver, environmental and vehicle parameters [65]. Driving safety issues are dependent in particular on weather messages (rain, snow, ice, fog) based on this thesis. Both influenced impact groups will be researched in particular cases in later chapters of this document. The resulting effects for reduction of emissions and an increase in traffic safety will be taken into account through the parametrization of the "vehicles speed profile" presented here, as illustrated in Figure 8-1.

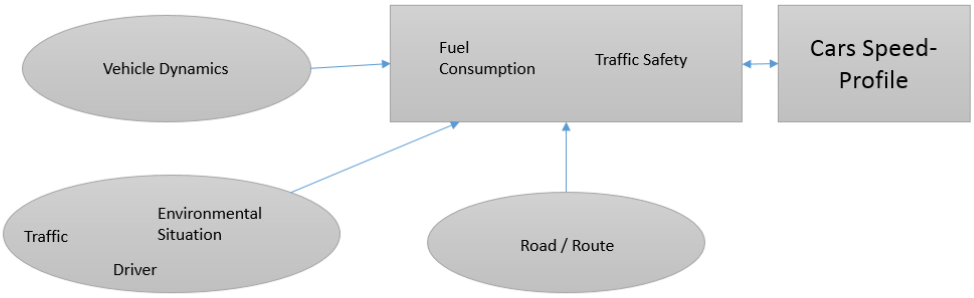


Figure 8-1. Driving parameters influencing fuel consumption and traffic safety. This affects the cars speed profile settings

8.2 Introduction

For green and safe driving, the primary route parameters as illustrated in Figure 8-2 are essential. Frequent changes of curvature may impact on driving safety in particular under degrading weather conditions (indicated as a preceding precipitation cloud ahead). Fewer available lanes on a highway will increase the risk of congestion and collision. Traffic calming measures including restricting the speeds within certain areas. On ADAS speed profile setting [65] such parameters will be predictively used in the calculation of the optimum approaching speed to such an area [66].

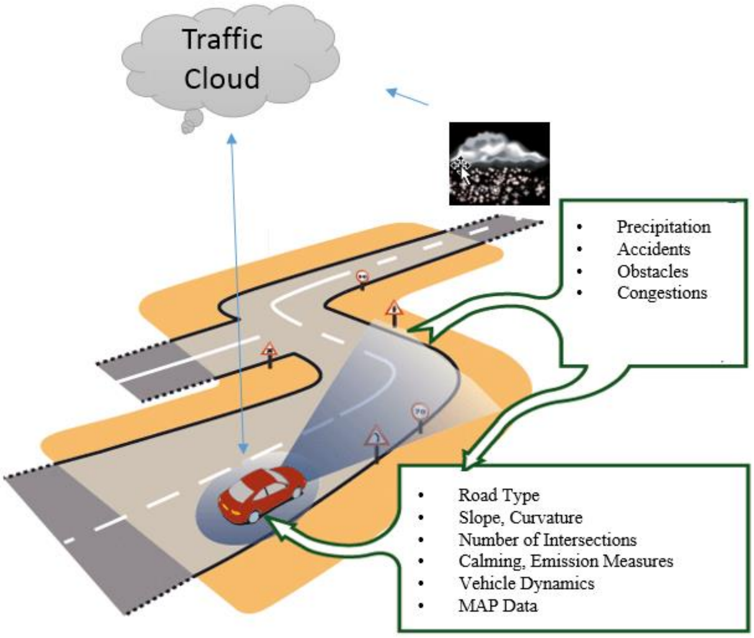


Figure 8-2. Green and safe driving - map data [67]

An example of a smart ADAS task of a cruise computer is introduced and illustrated in Figure 8-3. It's function ensures that the truck's dynamic speed setting is optimised to pass the hill under fuel efficiency and air pollution aspects. The truck ADAS systems needs position data to determine its exact map position and distance to the start of the mountain slope [68]. Based on the driver's attitude and the truck attributes, the approaching speed, climb and decline parameters are calculated based on a minimum fuel and emission target. The whole scenario is described as follows:

- Calculation of the appropriate approaching speed
- Incline – anticipate the incline ahead and equate the needed momentum
- Crest – anticipate the crest of the hill and begin decreasing speed
- Decline – anticipate and utilization of the trucks momentum to maintain optimal speed

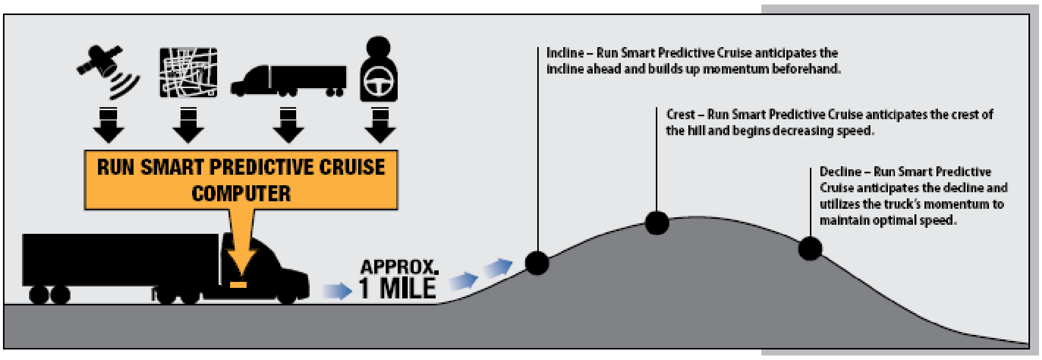

RUN SMART PREDICTIVE CRUISE COMPUTER
APPROX. 1 MILE
Incline – Run Smart Predictive Cruise anticipates the incline ahead and builds up momentum beforehand.
Crest – Run Smart Predictive Cruise anticipates the crest of the hill and begins decreasing speed.
Decline – Run Smart Predictive Cruise anticipates the decline and utilizes the truck's momentum to maintain optimal speed.

Figure 8-3. Digital map to enhance fuel economy [66]

The resulting effects on energy and emission savings of about 5% have been measured through a series of conducted tests [65]. This ADAS function should become an integral part of any future series vehicle to provide a contribution to reduce greenhouse gases.

The above example illustrates that a fuel efficiency effect can be obtained by means of an ADAS function. The availability of digital maps is essential for ADAS. Digital maps are capable of providing additional enriched map information along a planned route including information about slopes, curvature details, special obstacles or other impacts on driving. Figure 8-4 illustrates how a vehicle uses radar and LIDAR technique to scan the trajectory ahead. By supplemental use of digital maps, the existing view of a few hundred metres ahead is extended drastically. By definition, the map-based view ahead will be more or less infinite.

8.3 The electronic horizon

As described above, an advanced driver assistance systems (ADAS) requires various kinds of up to date information (see Figure 8-4) about the road the vehicle is driving on or is intending to drive on, in terms of an electronic horizon [69]. A method and system for providing an electronic horizon in an advanced driver assistance system architecture is defined as follows:

A method and system for use by driver assistance systems installed in a motor vehicle to continuously provide such systems with updated data about paths along roads onto which the motor vehicle can travel from a current position of the motor vehicle as the motor vehicle travels along roads. The method includes accessing a database that contains data that represents segments of roads and intersections of a road network located in a geographic region in which the motor vehicle is traveling and determining one or more paths along roads onto which the motor vehicle can travel from a current position of the motor vehicle. Each path is extended out to a threshold. Data representing each of the paths is provided in an organized data structure for use by the driver assistance systems [70].

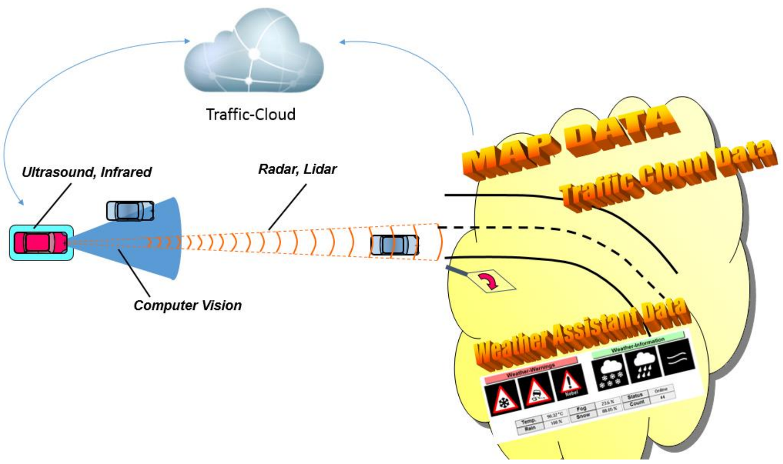
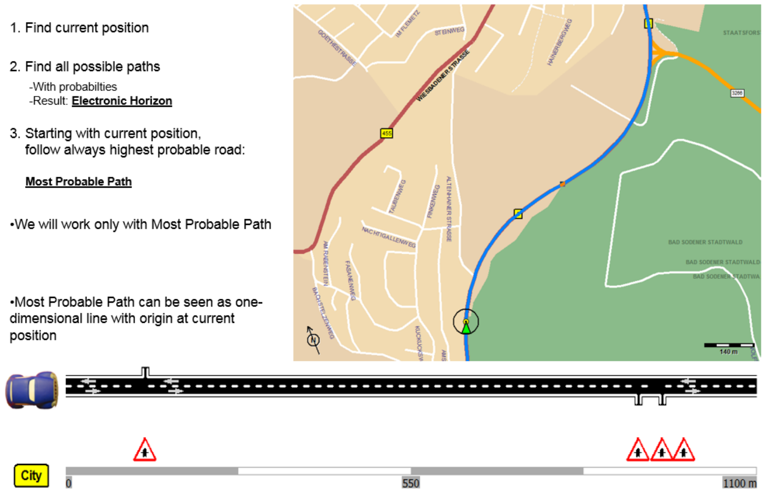


Figure 8-4. ADAS sensor maps - the Infinite Range Sensor [66, 67, 69]

8.4 The electronic horizon and the speed profile

The electronic horizon supports the vehicle with supplemental information of optimized driving strategies. In Figure 8-5 the navigation windows represents the optimized current routing [65]. For this calculation of the current position, all possible paths (considering their probabilities) are considered to resume the most probable path. In all cases, the most probable path can be treated as a one dimensional line beginning at the current position. Junctions, speed limits and all other traffic signs are considered for optimization of fuel efficiency and driving safety by influencing the speed profile setting of the ADAS systems [66].


1. Find current position
2. Find all possible paths
   - With probabilties
   - Result: **Electronic Horizon**
3. Starting with current position, follow always highest probable road:

   Most Probable Path

- We will work only with Most Probable Path
- Most Probable Path can be seen as one-dimensional line with origin at current position

Figure 8-5. Electronic horizon [66]

The architecture of the ADASRP (Advanced Driver Assisting System Research Platform) in combination with the BMW research car is outlined in Figure 8-6. The influence of MAP related data, the direction and position indicator and the weather-related information on the electronic horizon is shown [65]. The resulting speed profile is passed on as a parameter to the adaptive cruise controller (ACC) to set the cruising speed to the calculated optimum (fuel consumption and driving safety).

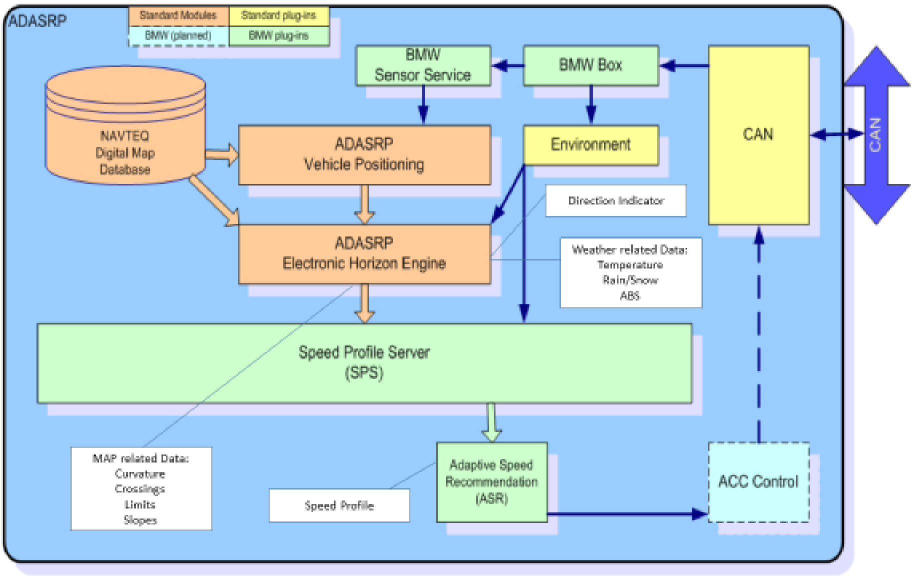
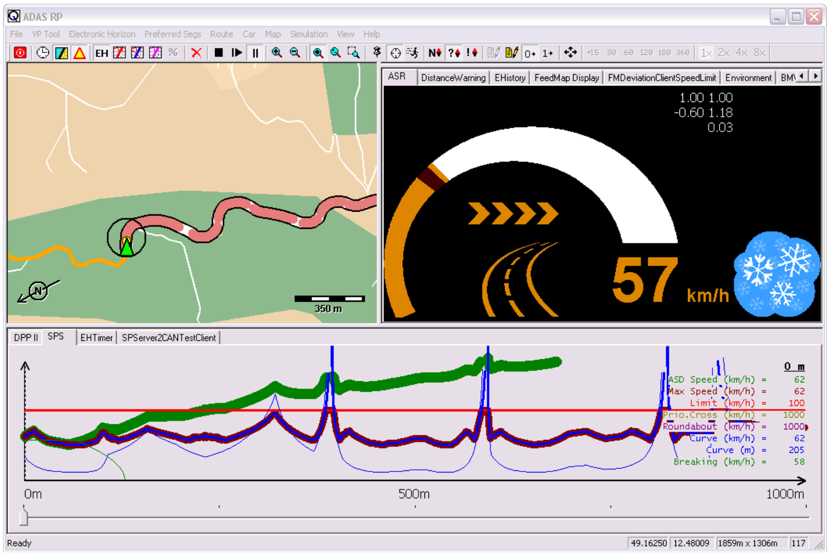

ADASRP
Standard Modules
Standard plug-ins
BMW (planned)
BMW plug-ins
NAVTEQ Digital Map Database
BMW Sensor Service
BMW Box
CAN
CAN
ADASRP Vehicle Positioning
Environment
Direction Indicator
ADASRP Electronic Horizon Engine
Weather related Data: Temperature Rain/Snow ABS
Speed Profile Server (SPS)
MAP related Data: Curvature Crossings Limits Slopes
Speed Profile
Adaptive Speed Recommendation (ASR)
ACC Control

Figure 8-6. Speed profile server and impact on ACC [66, 67]

Within the electronic horizon, digital map data such as road geometry, slopes and information of junctions are available [69]. Additionally, the weather assistant service may impact on the speed profile by passing upcoming restrictions to it. Figure 8-8 illustrates the changing speed limits on the vehicles trajectory. An evaluated and tested ADAS function (see Figure 8-7) installed in the BMW research car illustrates the benefits of the speed profile server [66].


ADAS RP
File VP Tool Electronic Horizon Preferred Segs Route Car Map Simulation View Help
ASR DistanceWarning EHistory FeedMap Display FMDeviationClientSpeedLimit Environment BMW
350 m
57 km/h
DPP II SPS EHTimer SPServer2CANTestClient
0m
500m
1000m
Ready

```
1.00 1.00
-0.60 1.18
0.03
```

```
                          0 m
     ASD Speed (km/h) =    62
     Max Speed (km/h) =    62
         Limit (km/h) =   100
    Prio.Cross (km/h) =  1000
    Roundabout (km/h) =  1000
         Curve (km/h) =    62
            Curve (m) =   205
     Breaking (km/h) =    58
```

```
49.16250 | 12.48009 | 1859m x 1306m | 117
```

Figure 8-7. Speed profile used within ADAS application [66]

The left-hand navigational map window reflects the current position of the vehicle and its upcoming change of curvature to the right. Additional information about road texture caused by weather influences such as ice and snow affects the speed profile setting. The resulting effect for the ACC is the adaptation to a secure new cruising speed and some visualizations for driver awareness of the degrading weather situation. Both aspects of driving security and eco-driving are effectively matched by the proposed strategy.

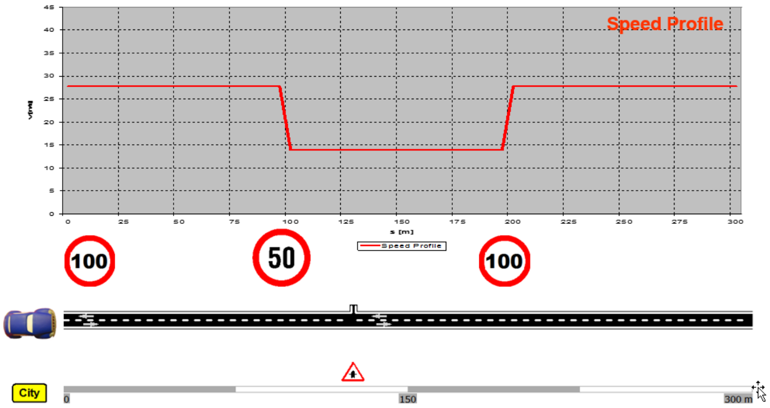


Figure 8-8. The electronic horizon and the speed profile [66]

8.5 The derivation of the speed profile

In the next section, the model behind the speed changes and the impact on the applicable speed profile is discussed in detail [65, 66]. The assumption is that the current speed of the research car is 90km/h and there is a 50km/h speed limit in 100 m ahead.

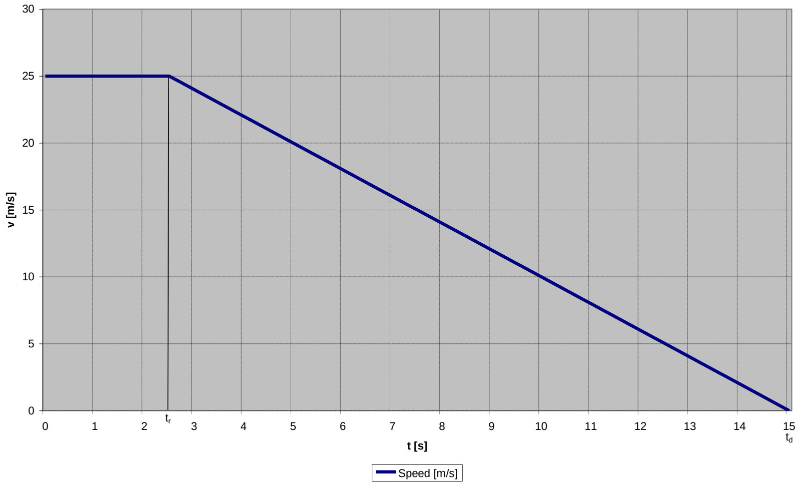


Figure 8-9. Deceleration to new target speed [66]

The question is raised as to whether the vehicle is still too fast to reach the new target speed after 100 m. Under the assumption that:

- Constant reaction time (t_r)
- Constant deceleration $(a, a \leq 0)$

- Known current speed (v_0)
- t_r = 2.5 s, a = -2 m/s^2, v_0 = 25 m/s = 90 km/h

The required time to stop is determined by the equation:

$$t_d = t_r - \frac{v_0}{a} \tag{8.1}$$

The corresponding braking speed as a function of time is defined as:

$$v(t) = \begin{cases} v_0 & 0 \le t \le t_r \\ v_0 + a(t - t_r) & t_r \le t \le t_d \end{cases} \tag{8.2}$$

The speed profile is a function of the path and thus the distance from the current location. The braking speed as a function of the distance is needed too.

The corresponding braking speed as a function of distance:

$$v(s) = \begin{cases} v_0 & 0 = s(0) \le s \le s(t_r) \\ \sqrt{2as - 2at_r v_0 + v_0^2} & s(t_r) \le s \le s(t_d) \end{cases} \tag{8.3}$$

The resulting graphical representation of the speed profile and the braking path is illustrated in Figure 8-10. As outlined, the yellow marking illustrates that the speed is still too high to achieve the target speed.

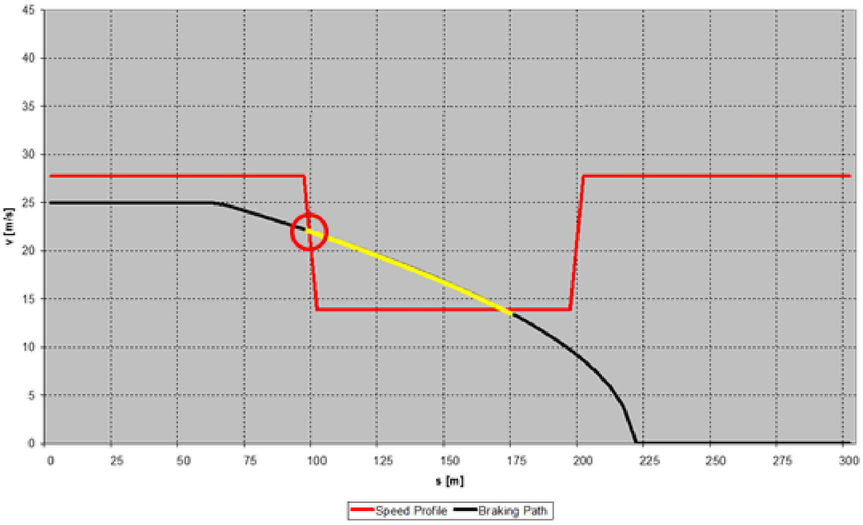


Figure 8-10. Speed profile and braking path [66]

In order to obtain a suitable braking path to just slow down in time over the distance to target, Figure 8-10 illustrates how to optimize and obtain $v_0(s, v_0)$, pointed out by the arrow. The braking path is shifted parallel left and with this speed profile the target speed is met at the crossing (see Figure 8-11).

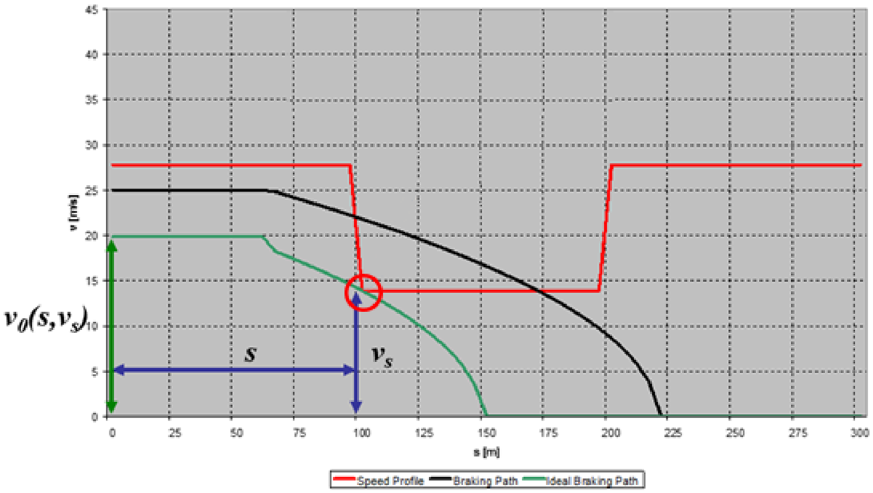


Figure 8-11. Ideal breaking path [66]

With $v(s) \to v_s$ and $v_0 \to v_0(s, v_s)$ the formula for the optimized braking path will be obtained as:

$$\begin{matrix} v(s) \to v_s \\ v_0 \to v_0(s, v_s) \end{matrix} \quad v_s = \begin{cases} v_0 & 0 = s(0) \le s \le s(t_r) \\ \sqrt{2as - 2at_r v_0(s, v_s) + v_0^2(s, v_s)} & s(t_r) \le s \le s(t_d) \end{cases} \tag{8.4}$$

$$v_0(s,v_s)=\begin{cases} v_s & 0=s(0)\le s\le s(t_r) \\ at_r+\sqrt{v_s^2+a^2t_r^2-2as} & s(t_r)\le s\le s(t_d) \end{cases} \tag{8.5}$$

The above formula of the braking path has been calculated as a function of the speed profile. In case the speed profile is given, the obtained function returns the optimum speed at any point on the most probable path (refer to Figure 8-12)

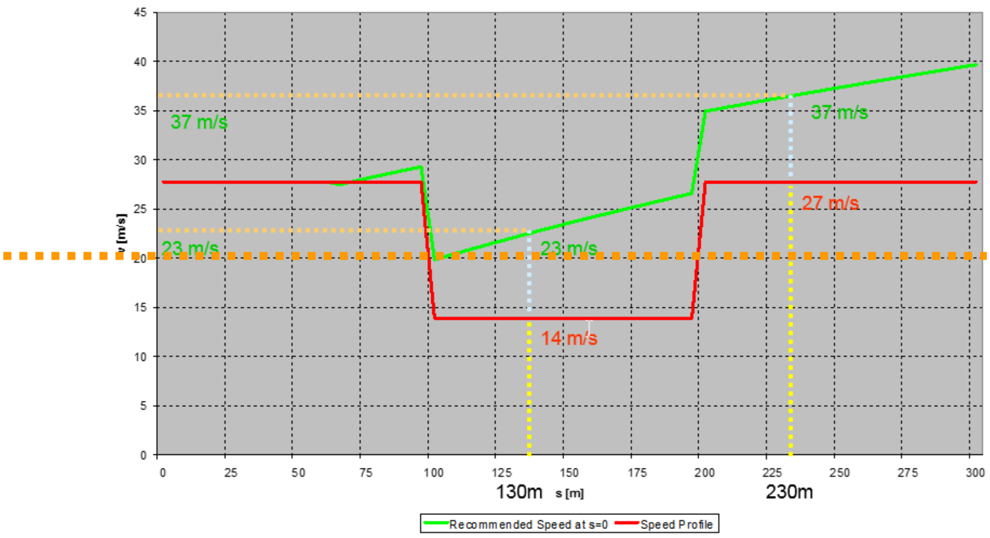


Figure 8-12. Current speed on most probable path [66]

The current speed v_0 at the dotted line in Figure 8-12 is:

$$v_0 \le v_0(s,v_s)\ \forall\ 0\le s \tag{8.6}$$

While the vehicle is driving on ACC or CACC [71] (Collaborative Adaptive Cruise Control) the dynamic adaptive speed is automatically passed to the controller to control the current optimised speed setting [72].

For evaluation, the speed limit warning in combination with the weather assistance function was installed in the research car. Legal speed limits either come from the navigational database or are captured by the vehicles vision sensors (camera). Weather data is provided by the weather assistant concept, already introduced in this research study. In order to adapt the speed profile to current weather conditions, the following (empirical) factors are set:

- Rain: Multiplication of the speed profile by 0.9 (technical assumption)
- Ice: Multiplication of the speed profile by 0.5 (technical assumption)

The factors are empirical assumptions, which have been obtained and approved based on the experiments performed on the research car.

8.6 Summary

The concept of the “speed profile” [66] has been evaluated using the experimental vehicle under different circumstances (see the following chapters). The outcome of the tests shows a reliable concept for many other in car applications referring to any kind of speed warning, alert and control system.

In particular in collaborative traffic, the concept will play an important role due the real-time exchange of C2X messages about current traffic hazards or traffic disturbances. In case of treating the car as a member of a vehicular sensor network, the granularity of data to determine a reliable speed profile will in fact improve. This is because the ambient traffic flow and traffic parameters such as weather and road conditions can be recognized continuously.

Future functions could be the junctions without traffic lights. On the basis of the remaining distance to the intersection and the current speed, the new optimum approaching speed can be derived for smooth and harmonized driving.

The overall effect of the speed profile concept is to increase traffic security and traffic efficiency in combination with applications introduced in this thesis. The measured effects on fuel consumption and greenhouse gas emissions will be set out in this thesis in the chapters that follow.

9 Applications for eco-friendly and safe driving

9.1 Active driving safety

9.1.1 Motivation

This chapter describes cooperative traffic safety features for vehicles with respect to their functions and technical structure. The requirements for vehicle infrastructure are identified together with mobile networks and central services in terms of performance and characteristics. The features can be used as apps on standard mobile devices, which interact with on-board systems by standardized interfaces. A proposal for the manufacturer-independent provision of automotive sensor data shall allow mobile devices easy access to the vehicle data and facilitate their transmission to central services.

9.1.2 Introduction

With further growth of traffic volume, methods for active road safety will become increasingly important. This can be particularly supported by powerful mobile technologies [73].

With the help of a wide variety of sensors modern vehicles collect extensive data such as outside temperature, vehicle speed, rainfall intensity, brake usage, acceleration, road condition according to ASP and ESP data, traffic density, adaptive cruise control, collision detection and prevention, number of passengers as well as current GPS position. These parameters and their position-dependent linkage and analysis allow for a whole new generation of assistant systems.

This document presents ideas for intelligent and networked driving assistant systems, which alert the driver in time to external events and conditions, thus leading to an increase in both safety and comfort.

9.1.3 Proposals for driving safety systems

In the following we discuss potential driving safety systems based on high performance mobile technologies assuming the future widespread availability of wireless technology LTE. The transmission of vehicle sensor data to the central framework and receipt of resultant data from the central databases is achieved as shown in Figure 9-1.

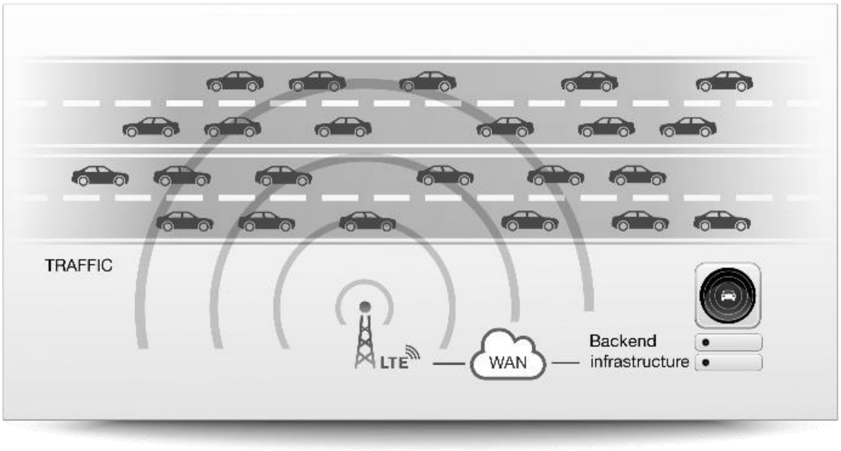
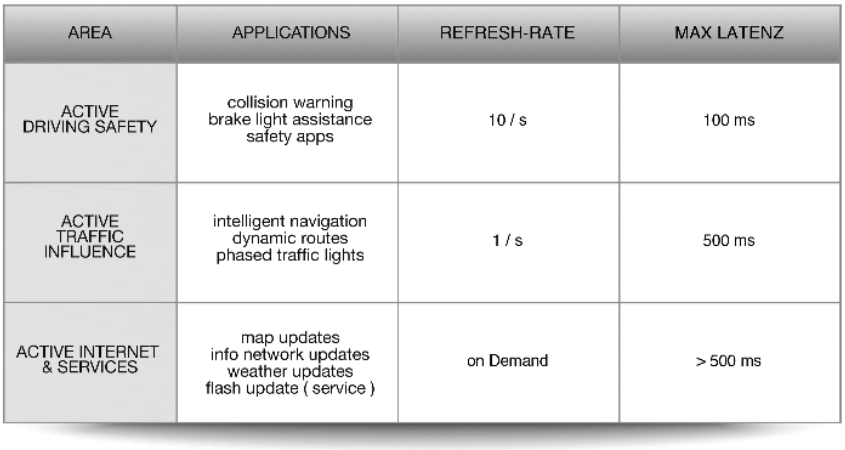


Figure 9-1. Communication base

9.1.3.1 Architecture

For all safety-relevant driving assistant features a high performance mobile technology is absolutely indispensable, as otherwise these features cannot be implemented Table 2. Safety Function and Mobile Communication Performance briefly classifies the requirements according to their scope of application and lists potential applications. In addition, it identifies the respective update rate (refresh rate) necessary to ensure each function and the respective maximum latency of information exchange with the central backend (Round Trip Time). LTE performance is essential for applications such as collision warning in tailbacks or accidents (brake assist).

Table 2. Safety Function and Mobile Communication Performance



AREA	APPLICATIONS	REFRESH-RATE	MAX LATENZ
ACTIVE DRIVING SAFETY	collision warning brake light assistance safety apps	10 / s	100 ms
ACTIVE TRAFFIC INFLUENCE	intelligent navigation dynamic routes phased traffic lights	1 / s	500 ms
ACTIVE INTERNET & SERVICES	map updates info network updates weather updates flash update (service)	on Demand	> 500 ms

In summary, the following criteria indicate the suitability of LTE in intelligent traffic systems (ITS), especially for active driving safety functions:

- Performance: High data rate in the DL and UL areas (100 Mbps, 50 Mbps) and low latency guarantee the transfer of time-critical traffic and vehicle sensor information.

- Availability: LTE will be available as a fully developed technology in the near future worldwide, at least along major traffic routes.

- Technology layer model: ITS use a fully developed technology base. Standardized availability renders proprietary infrastructure redundant.

- Expansion options (also geographical multicasting, see [52]). In the context of LTE further technical functions will arise in the medium term, hence targeted geo-based multicasting for ITS applications is both useful and desirable.

The framework for active driving safety is shown in Figure 9-2. It comprises the functionalities of data sources, communication infrastructure and central data management. Data sources are:

- Cooperative vehicles which communicate their sensor data and position to the central data management in the backend via LTE.

- Other vehicles which do not participate in cooperative driving safety but have activated mobile equipment on board. From the mobile net operator’s tracking the approximate position and velocity can be calculated and transmitted to the data management.

- Stationary sensors such as cameras, precipitation sensors and temperature sensors.

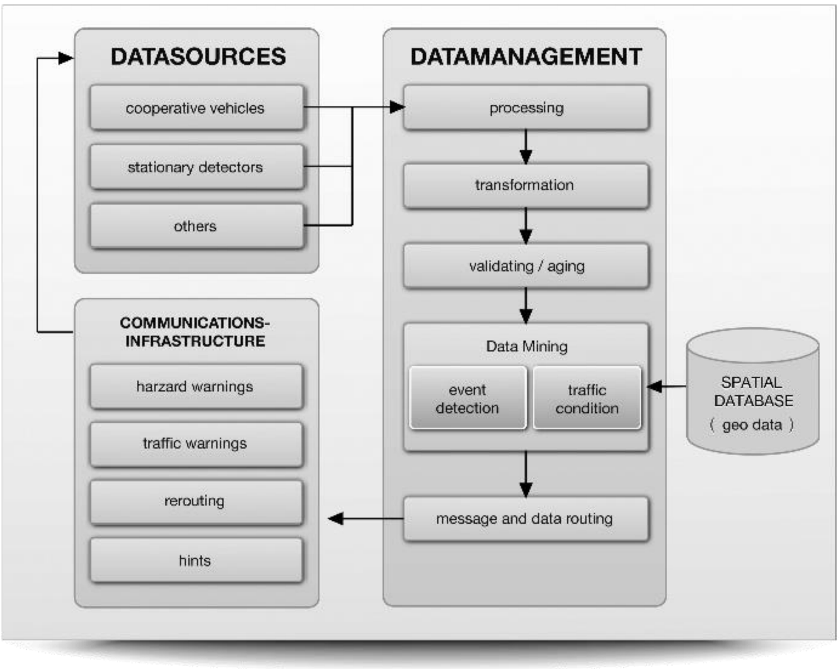


Figure 9-2. Architecture of driving safety

Car-to-Car Infrastructure Communication (C2CIC) occurs exclusively via mobile technologies. It will not require any proprietary hardware in vehicles or on roads. Car to Car Communication (C2CC) does not happen physically between the participants, but via C2CIC. The routing of messages addressed to the vehicles is carried out in the backend.

The architecture used employs the vehicle as a mobile sensor [74] for all the driving safety features presented, and thus as a supplier of physical data from its current environment. This data must be forwarded in de-facto real time from the mobile equipment to the processing backend and also be processed there in real time. Conventional storage in data bases and subsequent processing would be too slow, therefore a data streaming management System is employed [75].

Incoming sensor data and motion data are fed to a stream-oriented data mining system to calculate the various driving safety functions in real time. Powerful solutions for data mining systems are InfoSphere Streams or Massive Online Analysis (MOA), MOA being the preferred solution for this project.

Position tracking of vehicles and identification of vehicle clusters to be addressed are facilitated through a spatial database. Spatial databases are especially suitable for use in applications where objects are to be dealt with in relation to their geographical location (road map). There are now numerous providers of relational databases with add-ons for spatial addressing and processing which means that standard products can be used. Simple map matching algorithms are being applied to associate reported positions with the current map position. The implementation of the V-APP (Drive Guard) featured below uses the calculation of the shortest distance between reported position and the closest map segment. A range of possible methods is shown in [76].

9.1.3.2 Cooperative Collision Warning

The Cooperative Collision Warning provides alerts and assists when approaching other vehicles. This is especially useful in situations with limited view (narrow curves, blind corners, hilltops, or when visibility is blocked by the traffic ahead, e.g. trucks) and helps to prevent rear-end collisions in the event of a tailback or sudden braking. Rear-end collisions due to tailbacks or sudden emergency braking in particular account for a substantial share of total reported accidents, so a Cooperative Collision Warning function could significantly reduce the number of such accidents [77]. Radio-based communication among the vehicles is essential here, as is an accurate GPS position. This function can be based on optical recognition of brake lights of the traffic ahead, but only when there is a clear view of the vehicles involved.

The Cooperative Collision Warning System based on mobile technology (Figure 9-3) is not limited by such sight restrictions. It operates as follows:

- Braking behaviour is communicated to the backend according to its strength and current position. Detected braking of the traffic ahead is sent as a message to the backend.
- In backend processing the traffic to the rear is analysed in real-time conditions based on the incoming stream (Data Stream Mining).
- Addressing and distribution of monitored hazard information and its severity is subsequently distributed to the traffic behind.

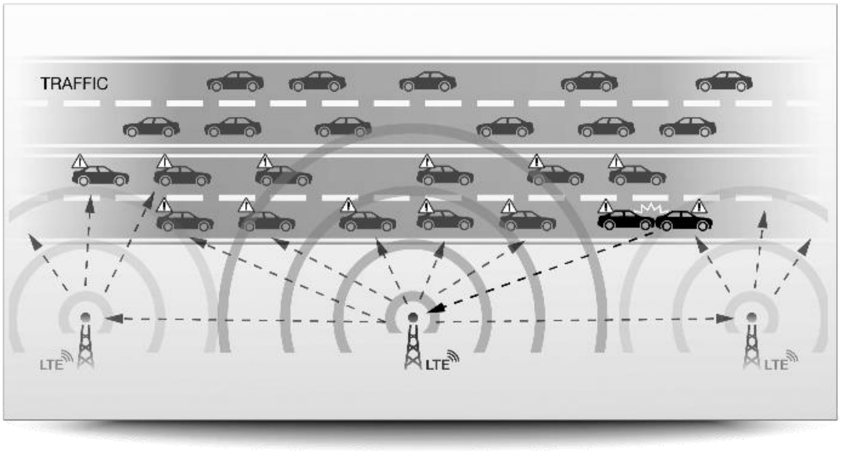


Figure 9-3. Collision recognition and warning

The backend system has a special task. Its server infrastructure needs to be built as a cluster to ensure the necessary response times (Table 2). Furthermore, it needs to identify the various participants in the geographic vicinity of the braking vehicle or of the obstacle and send the danger warnings immediately (see also [78]).

9.1.3.3 Cooperative hazard area warning

Warnings about black ice, aquaplaning risk, animals on the road and any other unexpected obstacles are further aspects of active safety. Here again, communication and tracking technologies are necessary to communicate or obtain information on hazards and dangers.

Similar to Cooperative Collision Warning, a logical persistent connection to the backend is needed. As a Cooperative Hazard Area Warning does not describe a particular event in the vehicle but instead derives hazards from a variety of outside parameters, a multitude of sensor data (see Figure 9-4) must be correlated and combined with the current position of the vehicle and then sent to central data processing. Depending on the outcome of the evaluation in the backend, a warning is sent as shown below.

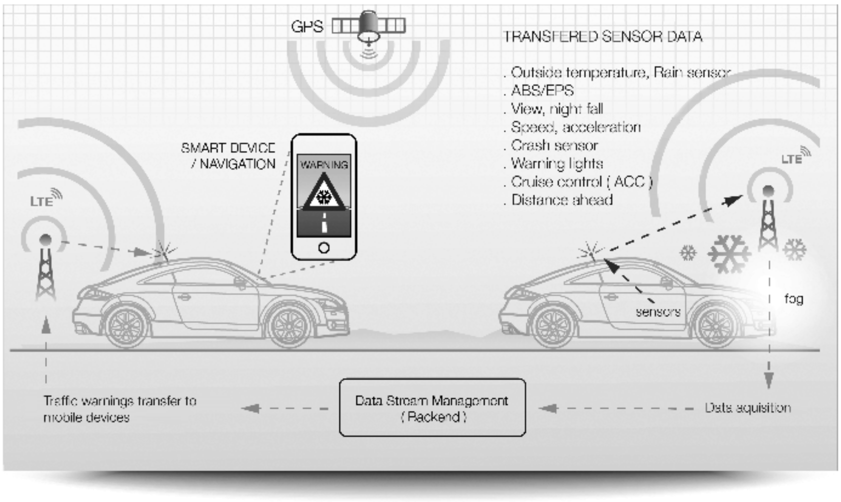


Figure 9-4. Function of hazard area warning

The function of the black ice warning shall be used to demonstrate Cooperative Hazard Area Warning. Figure 9-5 shows how in road conditions with limited or blocked view, a timely warning generated by the technologies described above can increase the level of safety for the traffic behind the ‘reporting’ vehicle.

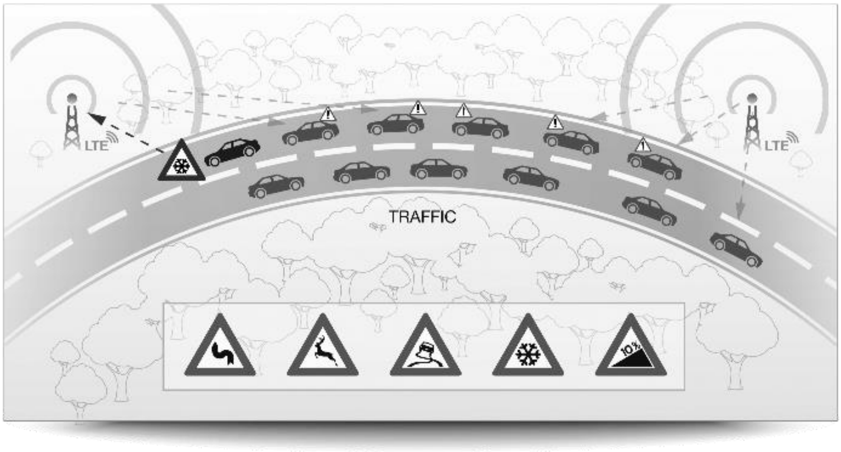


Figure 9-5. Hazard area warning

The black ice warning needs the sensor data for outside temperature, precipitation, windshield wipers, ABS/ESP, light levels and velocity. The data, along with the current position, is cyclically transmitted in real time to the backend [79, 80]. Based on the data received and including geo-data (road data), the analysis function of the backend system provides a road conditions summary. In order to guarantee reliable information, the data of vehicles that travelled the route before also needs to be correlated. Weather information from stationary weather sensors verify the weather conditions that have been identified and render them more precise. These meteorologically valuable results are passed on to weather services as well as other social networks.

In addition, with the data streaming received the backend system can independently predict black ice. The following functions are algorithmically used in the analysis system:

Current positions of information sources are correlated with the geo map [76]. In order to do so the data points which were received as a stream are fed to the DSMS and subsequently to the Data Stream Mining.

Analysis of various physical and meteorological data points (temperature, precipitation, visibility and light level). An AI system learns tendencies for the respective region and combines them with the data from stationary weather services, so that a precise prediction for the area of travel can be provided.

Subsequently this newly generated black ice prediction is distributed to the road users throughout the affected region which will result in a lower risk of accidents.

9.1.3.4 Intelligent routing and navigation

Today autonomous GPS navigation in vehicles, depending on their technology level, is adjusted dynamically to disturbances by traffic information (TMC). The elapsed time between this information and the correlating responses ranges from between 30 to 60 minutes. Nevertheless, through cooperative communication exchange among vehicles traffic delays can be automatically detected, reported and used for cooperative routing and improved navigation. The reduced latency allows smarter and more efficient response to traffic density and flow, succession of green lights, road closures, construction sites, various obstacles, changing lanes and parking situation.

The V-APP (Drive Guard) demonstrated below will also provide this functionality. Dynamic navigation will in future be available - both integrated in mobile devices and in-car technology. It can receive cooperative warnings through appropriate APIs and in turn send visual, acoustic and mechanical signals (e.g. a vibration of the steering wheel). External systems such as the mobile service offered by Inrix [81] for navigation, traffic telematics and services can also subscribe to Cooperative Driving Safety Features via an appropriate business model, and thus provide dynamic road safety data to their clients as shown in Figure 9-6.

Figure 9-6. Routing und navigation [81]

9.1.4 Powerful standardized communication technology - a prerequisite for intelligent road safety functions systems

The critical requirement for such safety-increasing assistive functions is robust communication technologies with low latency and high bandwidth such as LTE (Long Term Evolution as UMTS, 3G successor technology) and WLAN 802.11p. Their suitability has already been investigated in several research projects [82, 83]. The identification and addressing of road user groups (a group of vehicles in the same place and time) required for C2C (car to car) or C2X (Car to Car & Infrastructure) communication has been outlined there and is used as the technical basis here (see also [84]). For the generation and sharing of sensor data described above a communication interface for LTE and Wi-Fi and Bluetooth to 802.11p AUTOSAR Basic (AUTomotive Open System ARchitecture) [85] as shown in Figure 9-7 is presented.

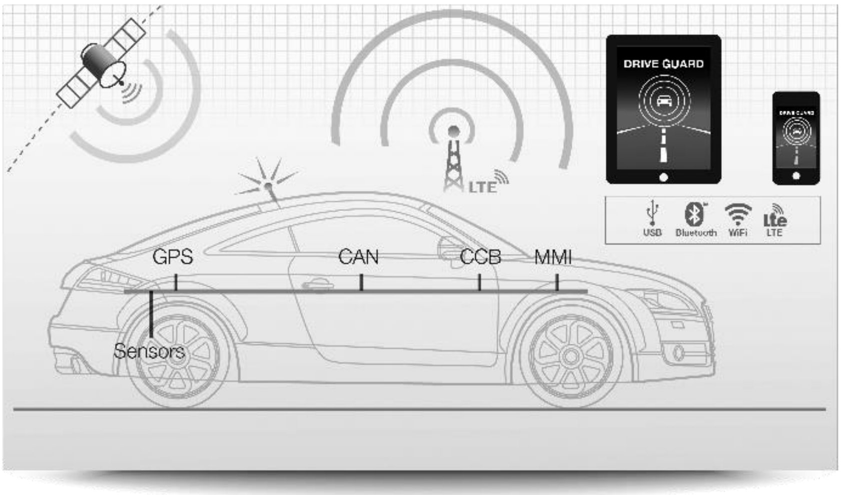


Figure 9-7. Car connect box communication interface

The vehicle can communicate its sensor data (see C.) to the backend system and exchange results via a mobile network interface. Currently the sensors are networked with their control devices in the vehicle by on-board CAN bus systems. Via an OBD (On Board Diagnosis) interface only limited data relevant for traffic safety applications can be collated.

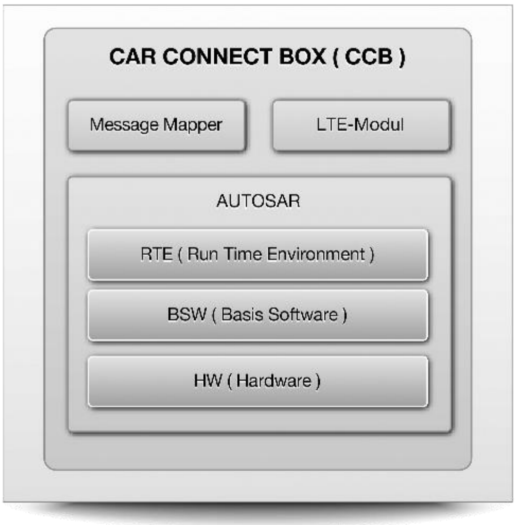


Figure 9-8. Car connect box

The Car Connect Box (CCB) shown in Figure 9-8 serves as a standardized, manufacturer-independent interface for all future vehicle models. The implementation of software services is based on the AUTOSAR standard. AUTOSAR enables standardized development of ECU software. The communication tasks in the electrical system get separated from the actual logistics of the ECU, thus creating flexibility for the integration of the CCB in vehicles from various manufacturers.

The illustrated Message Mapper component includes firewall, encryption and security features to prevent unauthorized access to vehicle data. The LTE module provides the mobile connection with a fallback to 3G technology if LTE is not available. In the vehicle itself mobile devices can be connected via USB, WLAN and Bluetooth.

9.1.5 Traffic safety application for mobile devices (V-APP)

To prove feasibility we will demonstrate a program V-APP (traffic safety application Drive Guard, see Figure 9-9) in which some of the above mentioned features are realized based on existing mobile networks (3G, LTE).

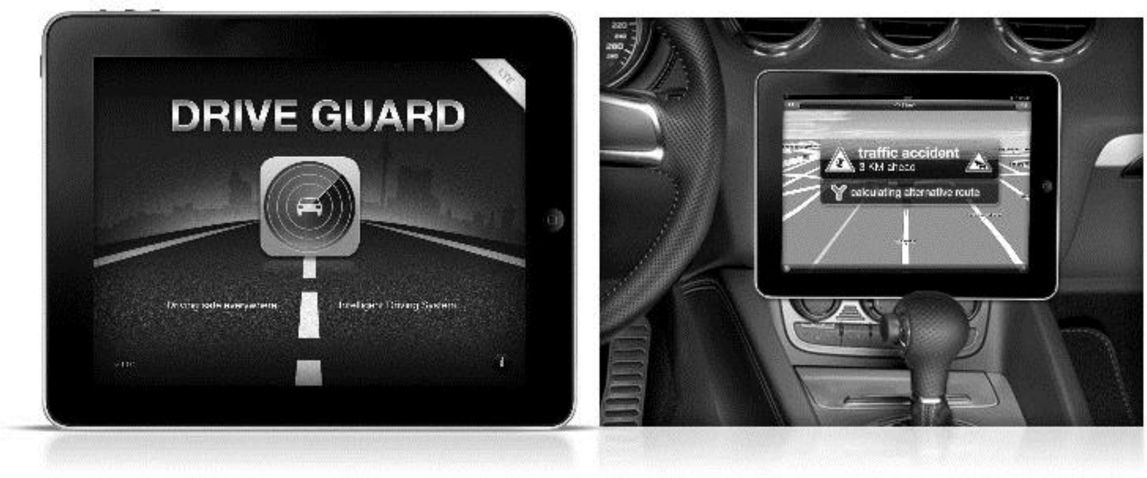


Figure 9-9. V-APP Drive-Guard

The app is started via the user interface shown on the left. During active operation you will see a warning in the context menu on the right-hand side. As described in 2.3, the information from cooperative driving safety functions can also be displayed on navigation devices made by other manufacturers. An exchange of their manual event recording with the backend via APIs should also be possible. Figure 9-10 shows the possibilities of event recording [81].

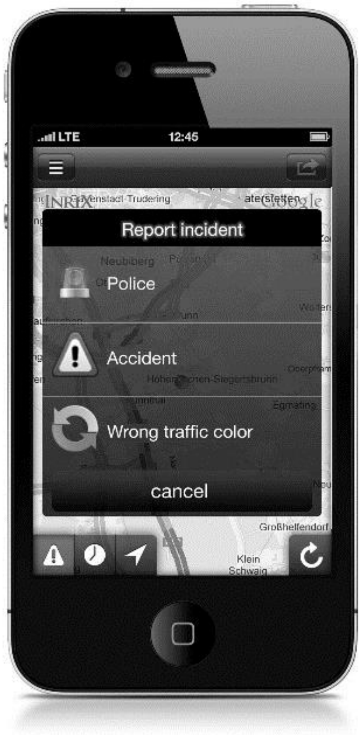


Figure 9-10. Inrix APP with event recording

This illustrates the interaction of mobile devices with the central services of the driving safety functions. The above mentioned vehicle data points are communicated to the backend system via mobile technology. There the location-dependent data is assessed and provided in appropriate granularity to the V-APP (Drive Guard), (also see A.). Some driving safety applications (see B. and C.) are presented below, based on their communication with the driver.

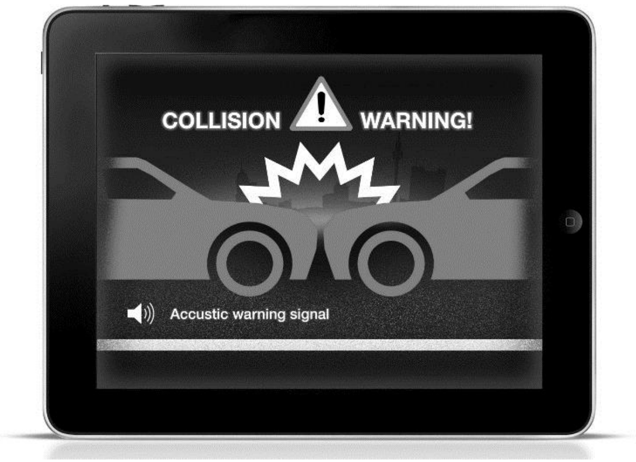
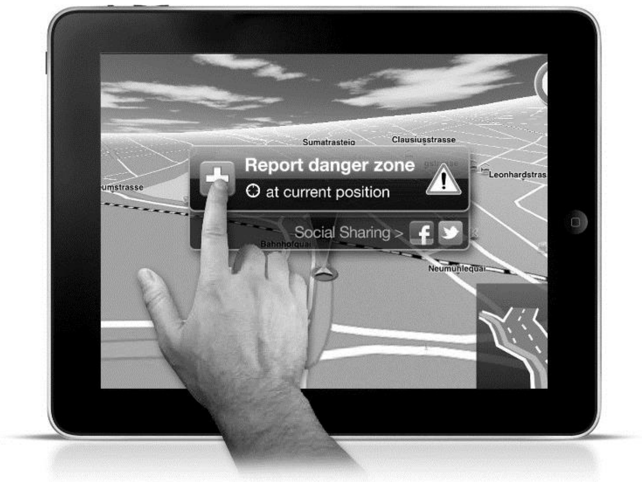


Figure 9-11. Imminent collision warning

For example, the Cooperative Collision Warning here issues an alert to a vehicle ahead braking abruptly by means of an acoustic, visual and mechanical warning (such as a vibration of the steering wheel) as illustrated in Figure 9-11.

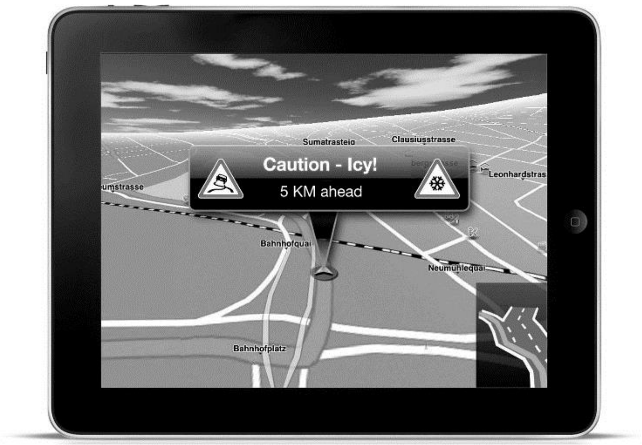
In the future, assistive braking and evasion manoeuvres could be triggered by direct interventions in the vehicle actuator. In addition to the Cooperative Hazard Warning there is the possibility of a short dialogue recording of a hazard warning for the driver as shown in Figure 9-12, because not every warning is automatically detected and promptly deducted from sensor data



Report danger zone
at current position
Social Sharing >

Figure 9-12. Manual hazard warning

According to their type of subscription service, traffic communities / forum or third party routing and navigation systems manufacturers can automatically be supplied with this data from the central backend in order to provide their users with the most current traffic information possible.



Caution - Icy!
5 KM ahead

Figure 9-13. Weather hazard warning

The warning to the driver (Figure 9-13) is automatically displayed on the device to allow plenty of reaction time given the distance to the point of danger. This information serves primarily to increase driving safety and reduce the risk of accidents. Collated data can also be used to support the routing of the on-board navigation. Interfaces between the driving safety features shown here and the on-board navigation systems allow the exchange of messages. The hazard warnings and traffic reports received thus also permit the calculation of alternative routes.

9.1.6 Conclusion

In recent years car manufacturers have developed various functions that aim to increase driving safety and have significantly contributed to statistically reducing the number of accidents [77]. Among these functions are ABS, ESP or lane change assistant. All of these functions operate autonomously, i.e. based on the sensor data from the vehicle itself. By also making use of mobile technology, cooperative driving safety functions today help identify critical situations and warn the driver in time through interaction of the car's on-board functions or coordination with the back end.

In addition, the vehicle serves as a mobile sensor itself and provides current data on its surroundings, which in combination with static sensors draw a far more accurate picture of the current traffic, road and weather conditions. The primary prerequisite for widespread use is manufacturer-independent standardized access to the on-board vehicle sensors and connection to a central coordinating backend via a powerful LTE mobile network. Our suggestion for standardization is the Car Connect Box discussed above.

By increasing the use of the safety enhancing features presented here, the overall ease and comfort of general navigation and re-routing can be increased while overall vehicle accident rates will certainly be reduced. Particularly in connection with the use of mobile devices there is no dependence on specific manufacturers.

9.2 Green, eco-friendly and economical driving

A novel approach to the enhancement of Cooperative ACC by deriving real-time weather information will be the chosen as an example to illustrate eco-friendly and eco-driving [86].

9.2.1 Motivation

The concept of Cooperative Adaptive Cruise Control (CACC) leverages wireless communication between vehicles in order to harmonize their cruising speed and thereby to gain a positive impact on fuel consumption and CO_2 emission. In this research, a novel approach of incorporating real-time weather information into the CACC [71] functionality will be introduced. Local weather changes are detected in a cooperative manner by sharing sensor data between vehicles, scanning the local vehicles environment for the current weather situation and retrieving location-based weather data from public services or road operators. This information can thus be used to anticipatory adapt the speed of the vehicles in relevant areas, according to the expected impact on driving safety and efficiency. In this way, a considerable increase in safety can be achieved, particularly in the case of severe weather conditions such as fog, heavy rain or blizzards. Additionally, since approaching traffic can slow down early enough, fuel consumption and the probability of arising traffic congestions can be reduced and so CO_2 emissions are minimized. In order to bring this concept to the cockpit, we present a weather assistant function with an integrated HMI which was realised in a prototype vehicle.

9.2.2 Introduction

Transportation is one of the major sources of pollutants and greenhouse gas emissions. In 2009, about 24% of the overall greenhouse gas emissions in the European Union was associated with the transport sector, mainly from road transportation [87]. A lot of research and development is currently being invested in this area. Reducing fuel consumption and encouraging environmentally friendly driving is becoming an increasingly important issue in order to minimize the carbon gas emissions. The BMW Group, as a leading premium car manufacturer, has reduced fuel consumption and CO_2 emissions of its vehicle range with its 'Efficient Dynamics' strategy to a greater extent than any other car-manufacturer.

Cooperative Adaptive Cruise Control (CACC) [10, 71] is one of the most relevant applications aiming to decrease fuel consumption and emissions in the future. CACC extends the well-known Adaptive Cruise Control system (ACC) [88], which is commonly based on forward-looking radar, with a wireless communication link with one or more preceding vehicles. In this way, string instabilities [88] in platoons are reduced and so, as compared to regular ACC, the impact on traffic throughput and consumption/emissions can be significantly improved. Since weather influences driving safety and routing, it needs to be considered in navigation systems. In this way, pollution factors, energy consumption and CO_2 reduction are addressed; the driver’s interaction with online weather information becomes increasingly important. In order to use CACC and leverage online weather information, a weather assistant is introduced in this paper. This weather assistant enhances CACC functionality by taking the current weather situation into account, thereby striving to further reduce fuel consumption and CO_2 emissions.

The work presented in this paper was done in the context of the TEAM (Tomorrow’s Elastic and Adaptive Mobility) project [60]. TEAM offers BMW and its partners the opportunity to share opinions, requirements and concepts regarding the field of eco driving to reduce CO_2 emissions. In order to prove the underlying concepts of this work, a reference implementation was carried out using a BMW-related test carrier that was used within the project. The mission of the TEAM project is to make travellers and infrastructure act as a TEAM, adapting to each other and to the situation so as to create optimum mobility conditions at all times. It leads to more (eco-) efficient, flexible and safer traffic. TEAM started in 2012 and will end in 2016. It is co-funded by DG CONNECT, European Commission within the 7th Framework program. TEAM essentials are that a TEAM is motivated and shares individual and common goals while cooperating through communication. Such a TEAM produces better results than the sum of its individual players and is therefore rewarded, both individually and globally. The TEAM project will refine and extend the outcomes of several ITS showcase projects such as CVIS, SAFESPOT and AIDE. In conjunction to DRIVE C2X it deploys the enhanced cloud-based Local Dynamic Map (LDM++), V2X [89] on 802.11p communication and collaborative GUI's for applications in different kind of vehicles. Within TEAM situational speed applications considering the current and predicted weather will be developed to minimize the CO2 emission and fuel consumption.

9.2.3 Related work

Efficient data acquisition technologies and reliable algorithms for its interpretation have been investigated in many other international projects such as TRACKSS [90]. A summary of technologies at different level of details involved into collaborative transportation systems is provided in [91].

As weather plays a crucial role in the safety and efficiency of transportation systems, research has also been carried out in this particular direction. An example of the importance of awareness of the current weather situation for transportation and conceptual development of mobile weather monitoring systems is shown in [92] and [93]. A brief overview of existing international programs in the field of road and weather information systems can be found in [94]. A description of comprehensive real-time weather notification systems using a combination of in-vehicle sensors and infrastructure sensors is provided in [95–99].

Safety and CO_2 efficiency of transportation systems is a global challenge. Therefore standardization and control over technological development in this area is important. Drive C2X project is one of the EU projects dedicated to the comprehensive assessment of existing cooperative transportation systems, the creation of testing environments and the promotion of cooperative driving [100, 101].

In order to acquire safety-relevant information on the road, different methods can be used: physically based and visual methods. Examples of different sensor technologies for acquisition of weather-related data on the road can be found in [102–104]. Traditionally measured parameters are: temperature, traction, speed, acceleration, etc. Exploiting visual information collected by the vehicle itself is not new, but in many cases it is still challenging. Changing illumination or visibility parameters reduce the usability of information measured. A possible solution to this problem is sensor fusion of computer vision information and data from other external sensors [105, 106]. Another kind of fusion, the combination of fixed infrastructure and mobile sensors, has also been shown to deliver reliable results [107, 108].

9.2.4 The effect of the weather assistant on CO_2 reduction in combination with CACC

The advantages of "Collaborative Adaptive Cruise Control" (CACC) are the harmonization of traffic flow, improved driving efficiency and safety. Utilizing online weather information together with CACC will even increase these effects. Let us assume that while vehicles cruise there are incoming predictive weather warnings prior to approaching the affected region. The vehicle's weather recognition system enforces the routing and speed profile to be recalculated and optimised for best distance and timing to destination. This leads to the desired reduction in energy consumption and CO_2 emissions, while CACC adopts the new setting for the onward cruise.

The idea of merging CACC and real-time weather information has been demonstrated by means of demonstrating concept implementation using a dedicated test vehicle. This vehicle was developed to act as a comprehensive test and evaluation platform for the TEAM project. The vehicle (see Figure 3-1) is equipped with various environmental sensors such as LIDAR, cameras, ultrasonic devices and IR sensors. It will also enable map database suppliers to develop new methods of dynamic data acquisition, dynamically extending data formats and obtaining live feedback.

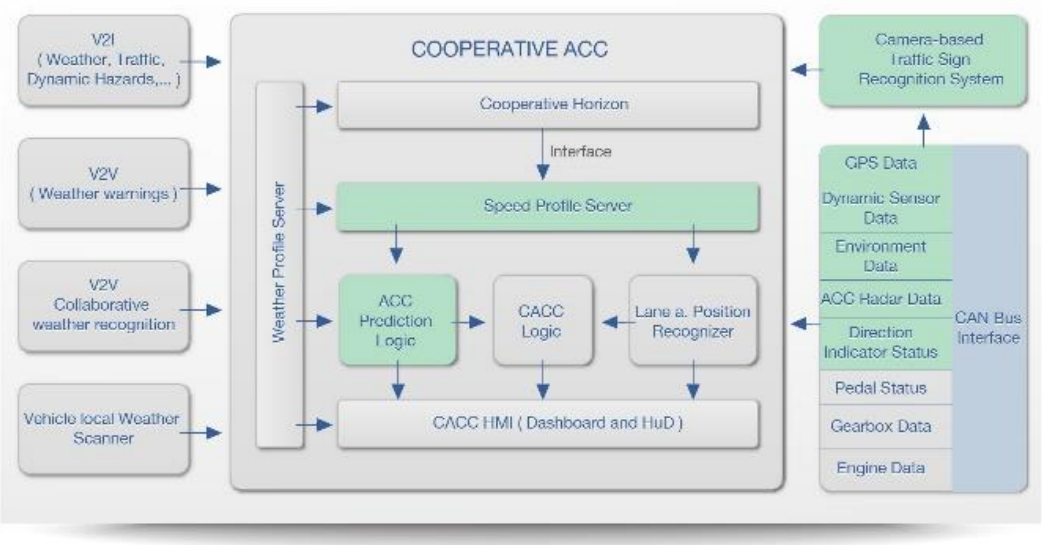


Figure 9-14. Structure of CACC within TEAM

Figure 9-14, depicts the structure of the TEAM-CACC application, which is deployed in the test vehicle. This structure aims to improve CACC by leveraging online weather information. In the following, the concept of this structure will be described with reference to its main components and their related functional aspects.

In order to obtain the positive effects of CACC, mainly CO2 reduction through harmonisation of traffic flow, several collaborative aspects have to be addressed. For this purpose, V2V and V2I communication [89] is leveraged which enables sharing of information with other vehicles. The weather information in the current vicinity of a vehicle and its planned routing (see local weather in Figure 9-14) plays an important role. This combination of traffic flow and weather (aggregated within the weather server in Figure 9-14) leads to optimisation of the vehicle's speed profile and therefore a sustainable reduction of emissions. Weather conditions such as meteorological visibility and road traction factor directly affect velocity and safety. Awareness of these parameters improves velocity profile planning (see: weather profile server Figure 9-14) and helps avoid unnecessary hard braking as well as preventing collisions and skidding out of the lane. As a result, energy consumption and emission are reduced, and driving efficiency and safety are enhanced.

Location-based weather information can be obtained from road operators or special service providers. Collaborative techniques allowing a more precise determination of near-range weather and its severe influences are available. Figure 9-14 illustrates the effect of collaborative weather detection on CACC [71]. Weather data consists of event, location, time of observation and time of expiry. Such collaborative data is collected and processed on the weather profile server. The weather profile server issues messages and alerts to achieve the following:

- Tuning of the vehicles speed profile (through speed profile server)
- Modification of the ACC speed setting
- Influencing ACC behavior to take into account near range weather effects (through ACC prediction logic)
- Rerouting illustrated on the dynamic map (Cooperative Horizon)

Communication with the driver through visual effects and icons (CACC HMI)

9.2.5 Architectural design of the weather assistant infrastructure and applications

The critical aspect for the effectiveness of the weather assistant application to enhance driving safety is to obtain current and accurate weather data. Again, communication and tracking technologies are necessary to obtain or communicate information on weather warnings and threats. For this purpose, a logical persistent connection to the backend is needed, since a cooperative weather warning does not describe a particular event but derives hazards from a variety of outside parameter. A multitude of sensor data must be correlated and combined with the current vehicle position (FCD, Floating Car Data [109]) and sent to central data processing (see Figure 9-15). Depending in the evaluation in the backend, a dedicated warning is issued

In order to issue such a weather warning, relevant sensor data is required. This sensor data includes the outside temperature, precipitation, current status of the windshield wipers, current status of the anti-locking system or the electronic stability control system (ESP) respectively, the light levels and the current velocity of the vehicle. Along with the current position of the vehicle, this is transmitted to the backend at regular intervals. Based on the weather warning received and including the corresponding geo-data (road data), the analysis function of the backend system provides a summary of the current road conditions derived from the data mentioned. In order to produce reliable information, it is also necessary to correlate the data of vehicles that travelled the route before, utilize weather information provided by local fixed weather sensors (Figure 9-15) and verify the weather conditions that have been identified and re-evaluate the results. These meteorologically valuable results can be passed on to weather services as well as to social networks.

Complementary to a specific weather warning information, the data streaming attached can be used by the backend to independently predict hazards or changing weather conditions. The following functions are algorithmically used in the analysis system:

- Current positions of information sources are correlated with a geo map. In order to do this, the data points which were received as a stream are fed to the data mining of the backend system.

- Analysis of various physical and meteorological parameters such as temperature, precipitation, visibility and light level. An AI system learns tendencies for the respective region and combines them with the data from stationary weather services, so a precise pre-diction for the area of travel can be provided.

- Subsequently this newly generated weather prediction is distributed to road users throughout the affected region, which will result in a lower risk of accidents.

Figure 9-15 describes from left to right the origin of weather data beginning with floating car data (from car sensors or determined by the car's on board weather scanner). The next source of weather data is data from public authority's weather systems or road operators. The backend system does the forecasting and alerting described above and distributes the appropriate data to the affected vehicles within the depicted geo region to provide accurate weather data to their applications.

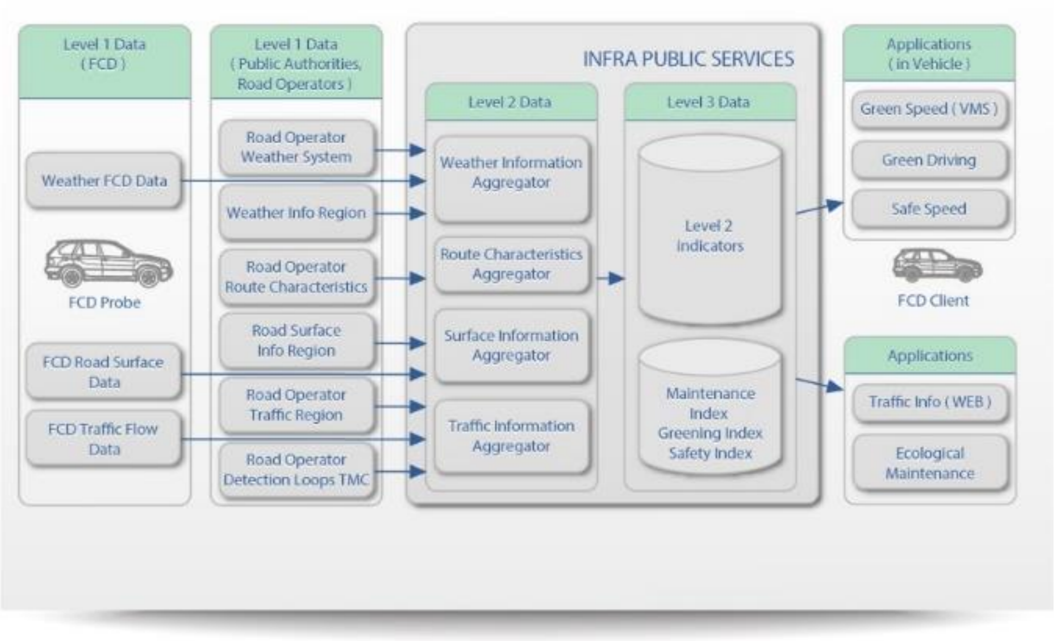


Figure 9-15. Weather warning in the cooperative horizon within the HuD

A particular focus is set to gain local weather data in the vehicle’s current vicinity such as limited visibility (fog, dust), rain, snow, slippery road and severe icing. The above mentioned weather prediction system considers single physical aspects such as temperature, rain or ESP/ABS activity.

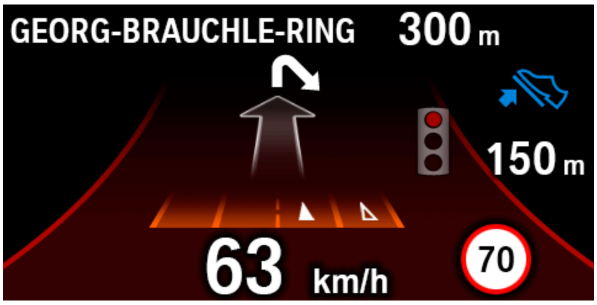


Figure 9-16. BMW HuD with V2X information in the cooperative horizon

As a result a weather tendency is evaluated and issued as a prediction. Through on board sensors such as LIDAR, cameras, ultrasonic devices and IR sensors, the vehicle (see Figure 3-1) is able to derive current weather conditions [110]. Recognizing collaborative messages of the same type from preceding road users, the picture can be estimated and predicted even more accurately [111, 112]. Technologies such as sensor fusion, weather determination based on sensor data and collaborative information acquisition are used to merge the incoming data to produce a significant result.

The cooperative horizon and the Head-Up Display (HuD) are used to provide advice about the optimised vehicle's collaborative settings (routing, power, navigation) for minimized CO_2 emissions. Appropriate messages will appear in case of particular weather warnings to achieve more driving security as a secondary result.

9.2.6 HMI Concept and user acceptance

In order to interact with our prototype application and evaluate its usability, we integrated the weather assistance into the HMI of the test vehicle described earlier. The experiments we conducted were designed to evaluate user acceptance of advanced and predictive green driving support systems and HMI concepts under recognition of life weather data. As shown in Figure 9-16, the driver is informed about the traffic lights at the next intersection [113, 114]. As an easy to capture message, the road is displayed in “red” while the eco-drive is not optimum. The influence on recommended approaching speed will be calculated automatically. Minimization of CO_2 happens in the background and results only affect the current journey.

In the event of a severe weather impact alert, messages are issued as illustrated in Figure 9-17. The blue coloured road indicates an optimum setting for eco-driving. Pop-Up windows appear on the screen if urgent weather incidents require particular attention. Weather impact automatically influences the calculation of best cruising speed and optimum routing to destination as well (see above).

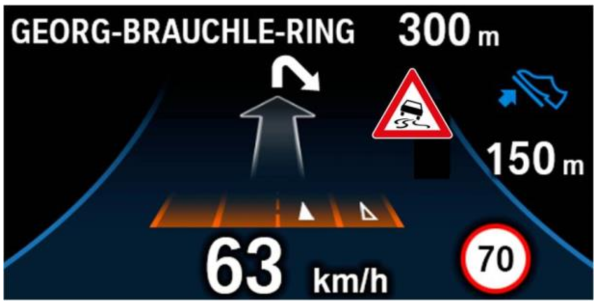


Figure 9-17. Weather warning in the cooperative horizon within the HuD

Since anticipatory and predictive driving offers great energy-saving potential, it is an important aspect to involve the driver in the loop and make him/her understand what the system is doing during travel. In the first prototyped application, a set-up was created to determine a point in time at which the driver is most prepared to receive predictive driving advice taking into account real-time weather data. Apart from the timing, the system's interaction with the driver represents another fundamental aspect offering potential for further investigation and improvement with regard to environment-friendly driving. As a consequence, a second driving simulator prototype application focusing on user acceptance of a new and innovative HMI concept [115] created by BMW was tested. This approach was used in the context of the ecoDriver Project [113]. It also aims to provide the driver with support for environment-friendly driving [115], thereby reducing energy consumption and emissions.

9.2.7 Conclusion

In this thesis we have described a way to reduce CO_2 emissions by combining CACC and leveraging of real-time weather information. This is achieved using an intelligent weather assistant system which enables detection of local weather changes in a cooperative manner by sharing sensor data between vehicles, scanning the local vehicles environment for the current weather situation and retrieving location-based weather data from public services or road operators. Information derived from the data is used to adapt the speed of the vehicles in relevant areas on an anticipatory basis, according to the expected impact on driving safety and efficiency. In order to prove the underlying concept of this work, a reference implementation was carried out using a BMW-related testing vehicle. As a result, the proposed architectural design and its related HMI presentation were verified. Furthermore, we were able to determine a point in time at which the driver is most prepared to receive predictive driving advice taking into account real-time weather data.

Future plans are to further investigate the impact of leveraging additional information, derived in a cooperative manner, in order to improve the safety-enhancing and CO_2-reducing effect of CACC. The outcome of this research shows the need for well-engineered V2X communication services for environment-friendly in-vehicle applications and predictive efficiency driving strategies with optimised longitudinal control, as well as multimodal and perspective HMI functionalities. BMW will demonstrate the complete chain of the CACC application with the Weather Assistant to reduce CO_2 emissions. The results will be taken into account in the development of the EfficientDynamics and ConnectedDrive strategy to achieve fuel reduction. The strategy includes:

- Engine start stop function and intelligent energy management
- Brake energy regeneration and gear shift indicator
- Improved engines and power train
- Learning electronic horizon
- Predictive efficiency driving strategies with optimised longitudinal control
- Multimodal and perspective HMI functionalities
- Intelligent ACC with special control strategies and Car2X communication
- Weather information connected to the navigation device

9.2.8 Acknowledgement

This chapter of work was supported by BMW and the European Commission under TEAM, a large scale integrated project forming part of the FP7-ICT geared towards cooperative systems for energy-efficient and sustainable mobility. The authors would like to thank all partners within TEAM for their cooperation and valuable contribution.

9.3 Collaborative vehicles and smart traffic lights

9.3.1 Motivation

The growing economic interlocking of European countries as well as cities leads to a strong dependency of citizens' prosperity on an efficient and safe transport system. The White Paper “Roadmap to a Single European Transport Area” [116] therefore sets out a competitive and resource-efficient transportation system and sets the goal as "increased transport and support mobility while meeting the 60% emission reduction target".

In the light of these preconditions and the arising technologies in the field of communication, information science and positioning, this particular application intends to provide a contribution to more efficient, safer and hence more sustainable mobility. These steps lead towards the vision of a transportation system with (semi-) autonomous vehicles - where the classical concept of traffic lights can be overcome and a self-regulating system [117] with cloud-based infrastructure can be established.

Again, the topic of this thesis "The Vehicle as Mobile Sensor in a Collaborative Network" plays an important in capturing the traffic light status as illustrated in Figure 9-18 and demonstrates how the vehicle communicates with "smart traffic lights" before crossing the intersection.

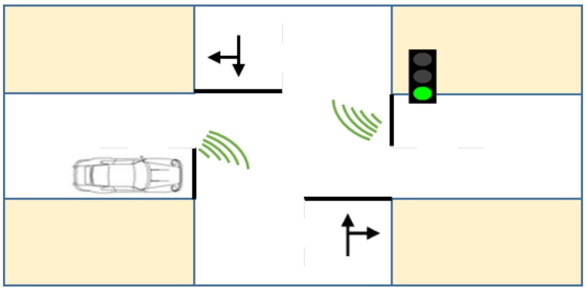


Figure 9-18. The collaborative vehicle senses the traffic light

Beyond this, the concept of “smart traffic lights” is approaching the performance and functionality of intersection-free road segments.

9.3.2 Introduction

As in the context of urban traffic flow, during periods of peak hours or unexpected events, road intersections are generally the principle cause of bottlenecks, a new generation of local cooperative traffic control systems with bi-directional information exchange is the main focus of this research. The bi-directional information exchange is based on a double feedback loop approach and will therefore enable the mutual adaptation of the behaviour of the vehicle and parameters of the installed infrastructure, such as approaching speed, optimisation of power setting and traffic light timing respectively. The type of vehicle is the main concern; hence, customised solutions for public transport, electric traction and heavy duty vehicles will be enabled. Due to this, the strengthening of multimodality and a more detailed consideration of environmental aspects can be achieved. Moreover the cooperation of vehicles and infrastructure enables new safety features such as detection of red light runners, which can enrich the functionalities developed in a possible future project [118].

9.3.3 Application concept

The Drive-Guard Traffic Light research application's primary aim is to reduce or even completely eliminate idling time at red light stops while driving in city conditions. The research has the potential not just to enhance the driver’s comfort and satisfaction of driving, but also provides fundamental benefits to society such as a reduction in fuel consumption and CO_2 emissions. This is due to the fact that it is exactly the moments when the vehicle rapidly reduces or increases its velocity that result in the highest fuel consumption and release of CO_2 emissions.

For this purpose, every traffic light should share its current status (signalisation by coloured lights) to a server as well as information about the time period remaining until the traffic lights status switches to another state.

The idea behind this is that on the server side, this information about the traffic lights' states will be merged and analysed in order to make a meaningful and accurate prediction as to how the vehicle's speed (see speed profile 8.4) should be changed (increased or decreased) to meet the above requirements. One might ask (1) why this information is not analysed directly by the vehicle itself and not on the server side or (2) why the traffic lights do not communicate directly with the car as in the common car-to-traffic-light infrastructure but use a server instead? The answers are very simple (this can also be seen as a reason for using a cloud-based server in automotive industry):

1. The server-side decision process can guarantee that we can extend the software’s logic or even fix bugs without the need to reinstall the software in the car. The car remains unaware of software updates we make. In general, this is a critical point for each system, where software updates on the client side are expensive or economically unreliable (decoupling of the software updates from the client).

2. First of all, the common car-to-traffic-light infrastructure is able to transmit a message only within a radius of 0.1 to 0.5 km, which can be temporally insufficient to perform the speed adjustment process. Secondly, the complete elimination of idling time at red light stops might involve the consideration of the next few traffic lights and not only the next one. Of course in the scope of this project the focus is only set on usage of information from a single traffic light. However, the algorithms are extended transparently for the user at any point in time due to the reasons set out in the first bullet point (1).

The idea shown in Figure 9-19 illustrates that the traffic light state is either directly transmitted to the vehicle via V2I communication, then the optimised approaching speed (harmonised and eco-friendly) is calculated by the vehicle's computer. Alternatively the traffic light states are known to the Drive-Guard Cloud and on the basis of the vehicle's location, the new advised approaching speed is communicated via V2I communication to the vehicle's speed profile server to adapt to the new recommended speed setting.

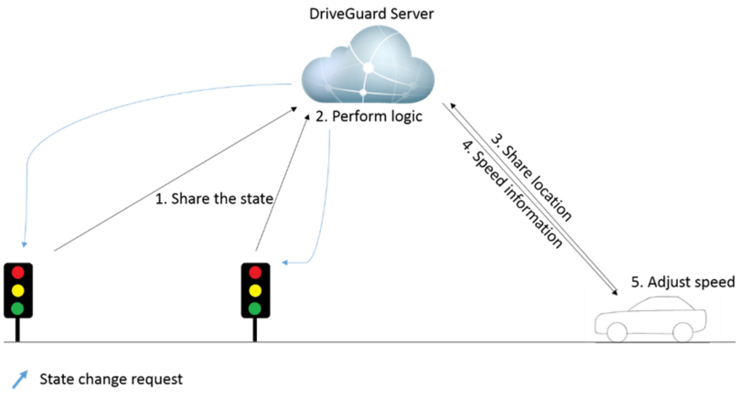


Figure 9-19. Collaborative vehicles and smart trafic lights

In addition, the server can push notifications to the traffic lights to update their state to obtain an optimum harmonised traffic flow [119] based on eco-friendliness and fuel consumption.

9.3.4 Smart traffic light application logic and algorithms

The presumption of the developed algorithm is standalone driving on the road (without any traffic or other obstacles), well-known curvature of the route as well as the only one traffic light relevant for the experiment. This idea was inspired by concepts introduced within a scientific publication about “Fluent coordination of autonomous vehicles at intersections” [120]. The idea described was used with minor adjustments to obtain an automatic approach for reduction of idling time at red light stops. The basic idea is that the time for reaching the intersection (traffic light) is considered and if exactly at this moment the traffic light is at state “red”, it is decided whether to decelerate or accelerate, depending on which action would lead to a smoother traffic flow (minor speed changes).

For simplicity and consistency the notations of the publication above [120] are used. The following cases are considered separately:

1. Decelerate (See Figure 9-20)

   The arrival time at the traffic light and the respective times t_1 and t_2 have to be calculated. It is necessary to decelerate in case $t_1 > t_2$, otherwise acceleration has to be checked (2). In this cast T_1 and T_2 have the semantics as in Figure 9-20 as well. The parameter σ in this case represents the duration of the RED state (T_{RED}) of the traffic light. The deceleration (change of the speed) has to be computed by equation (9.2) analogue to the referenced publication. Of course it is necessary to bear in mind that the current speed will be recomputed at every time step (every time the current location is shared with the Drive-Guard server e.g. 1 seconds) and therefore to use this algorithm in real traffic cases as well.

   Since the paper differentiates the current, desired and maximal allowed speeds, as a first implementation the car’s velocity is the desired one and the applied speed always remains within the permitted speed limits. It is beneficial because it eliminates the first part of the integral from an equation (9.2).

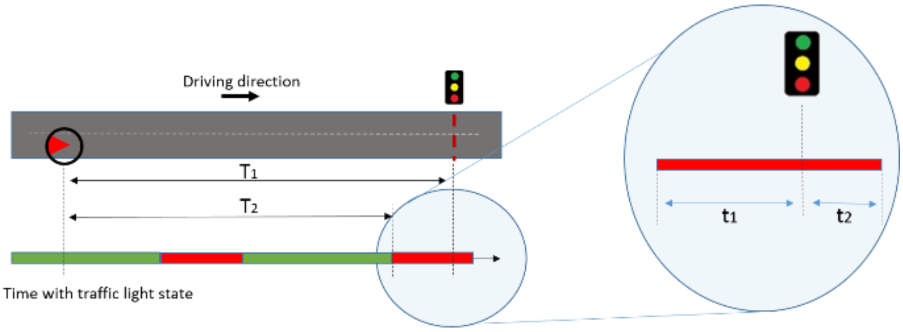


Figure 9-20. Decelerate due to insufficient timing

2. Acceleration (Figure 9-21)

   If $t_1 < t_2$, T_1 and T_2 have the semantics from Figure 9-21. The value of the σ parameter remains the duration of the RED state (T_{RED}) of the traffic light but in the equation (9.2) the function β (equation (9.1)) has to be computed according the second case. Everything remains unchanged as described above.

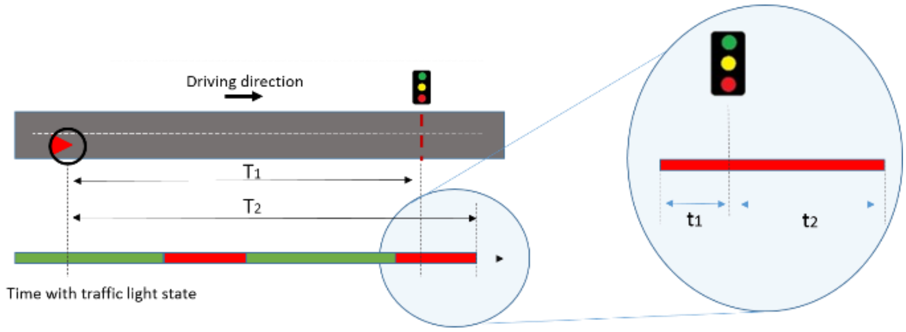


Figure 9-21. Accelerate due to sufficient timing

3. No action required

If T_1 is in a state of green light, no further action is required.

The following equations (9.1) and (9.2) have been explained above and are the basis for the implementation of the algorithms to calculate the respective approaching speeds to the intersection. The speed thus obtained is used in the traffic and vehicle simulator and the experimental car to perform the research experiments for validation of the ideas.

$$\beta(\tau_i, \tau_j) = \begin{cases} \frac{1}{\sigma}\log(\frac{\sigma}{\tau_i - \tau_j})\,v_i & 0 < \tau_i - \tau_j < \sigma \\ \frac{-1}{\sigma}\log(\frac{-\sigma}{\tau_i - \tau_j})\,v_i & -\sigma < \tau_i - \tau_j < 0 \\ 0 & |\tau_i - \tau_j| > 0 \end{cases} \tag{9.1}$$

$$dV_1 = \int_0^{T_2+\sigma} -\frac{1}{2}\lambda_1(v_1 - v_{d1}) - \lambda_2\frac{1}{\sigma}\log\left(\frac{\sigma}{\tau_2 - \tau_1}\right)dt \tag{9.2}$$

9.3.5 Testing, simulation and validation

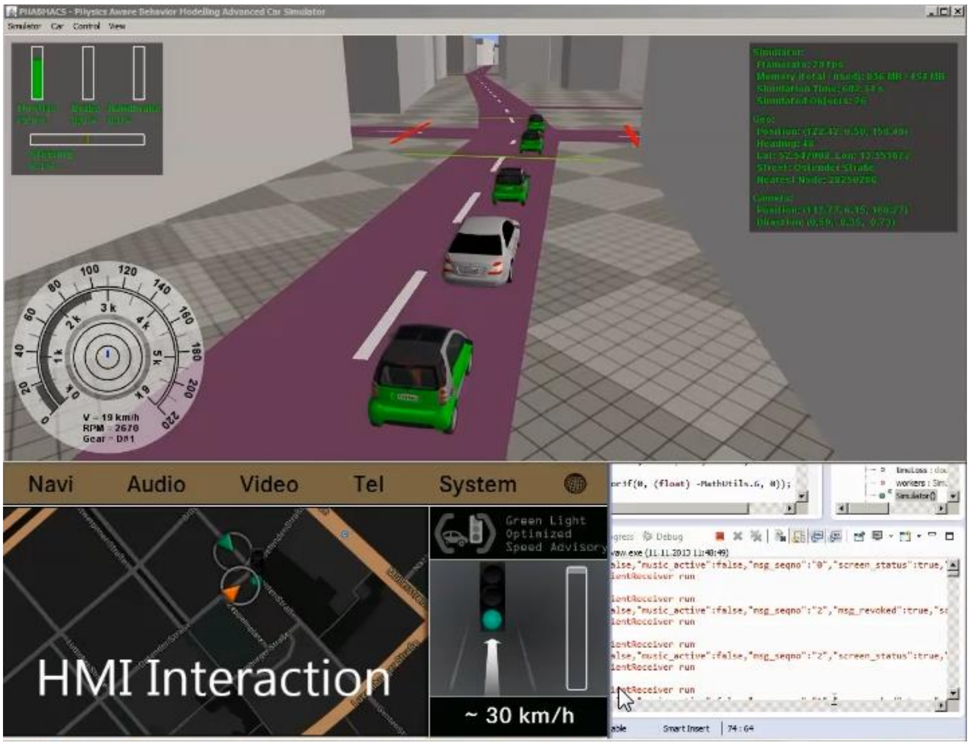
The experiments and validations of the concepts were conducted on the basis of the PHABMACS[9] [121] vehicle and traffic simulator and the BMW research car (see Figure 3-1).

[9] PHABMACS: **P**hysics **A**ware **B**ehavior **M**odelling **A**dvanced **C**ar **S**imulator - A vehicle and traffic simulator for research purposes provided especially for this research from Fraunhofer FOKUS.

In both validation methods, the algorithms to control and set the best-suited approaching speed can be implemented within the corresponding plugin.

9.3.5.1 PHABMACS simulation

The Physics Aware Behaviour Modelling Advanced Car Simulator (PHABMACS) provides a research-utilisable framework for testing advanced driver assistance systems within a simulated environment. In this particular case, vehicles and traffic lights at intersections are simulated with advanced simulation models that provide them with realistic driving dynamics in normal traffic situations. Tested ADAS (Advanced Driver Assistance Systems) can utilise simulated sensor data (Traffic Lights on V2I communication) as well as control simulated vehicles by using virtual actuators (the speed algorithm influences the speed profile setting, see 8.4) [121].



```
orif(0, (float) -MathUtils.G, 0));
```

```
.exe (11.11.2013 11:48:49)
alse,"music_active":false,"msg_seqno":"0","screen_status":true,'
ientReceiver run

ientReceiver run
alse,"music_active":false,"msg_seqno":"2","msg_revoked":true,"s
ientReceiver run

ientReceiver run
alse,"music_active":false,"msg_seqno":"2","screen_status":true,'
ientReceiver run

ientReceiver run
```

Figure 9-22. Smart traffic light on vehicle simulator

The illustration in Figure 9-22 shows in detail the experiment performed to measure fuel economy in combination with CO2 efficiency. The urban environment is focused on for evaluation by importing a German metropolitan city map into PHABMACS. Only the first vehicle represents the target for green light optimisation to harmonise the traffic flow. In the upper window, the traffic light (set to green), the power setting and some navigational details are presented. The lower mid-window shows the proposed algorithm to calculate recommended speed and the prediction of the traffic light when passing the intersection. Auxiliary two lower windows (left and right-hand side) provide a bird's-eye view and there is a trace window for debugging. The tests were executed in a series of about 100 separate test and validation cases. The traffic itself and the traffic light status were generated randomly. The particular algorithm (described above) to determine the optimum speed setting for respective harmonised junction crossing controlled the vehicle's speed profile, where parameters such as driven routing, current speed, fuel consumption and CO_2 emission were traced into a SQLite[10] database for subsequent analysis of the experiment [122].

The trace data obtained on SQLite basis provides input for an MATLAB analysis function which was developed especially for this purpose.

The following strategy was chosen to perform the trip data analysis (regarding the PHABMACS simulator and the subsequent real-life vehicle test trips following the next chapter):

1. Analysis of distribution for the considered variables in each vehicle and set-up (same car, drivers/simulator, system) to evaluate the significance of reduction and outlier analysis.

2. Comparison of result internally with each vehicle to highlight significant differences between systems/vehicles (e.g. haptic/non haptic, petrol/diesel/electric ...).

3. Similar comparisons can be carried out using on naturalistic tests too, but the significance of results must be evaluated.

4. Identification of a mean value (mean for trip or for driver/simulator or for vehicle - to be defined) for each configuration: type of vehicle/fuel/system/HMI for each vehicle.

    - To be evaluated if different vehicles with same configuration are considered (especially if they have different % of reduction).

    - To be evaluated if they also include some percentiles beyond mean values.

Output of the trip analysis:

- Global distribution of mean values of configuration and related statistics (plus some statistics on absolute reduction to be evaluated)

[10] SQLite is an open source software library that implements a self-contained, server less, configuration, transactional SQL database engine. https://sqlite.org.

- Mean percentage of reduction (also median)
- Range of reduction across configuration (percentiles)
- Percentage of configurations with saving
- Percentage of configuration with significant saving

The following Figure 9-23 represents the analysis (performed on basis of the above method) of 100 trip traces data without “Smart Traffic Lights” illustrated as “Baseline” and at “Smart Traffic Lights” presented as “Smart Traffic Light”. The final effect on fuel consumption leads to about a 22% saving. This underlines the importance and potential of the reduction of fuel consumption and any related reduction of CO_2 emissions.

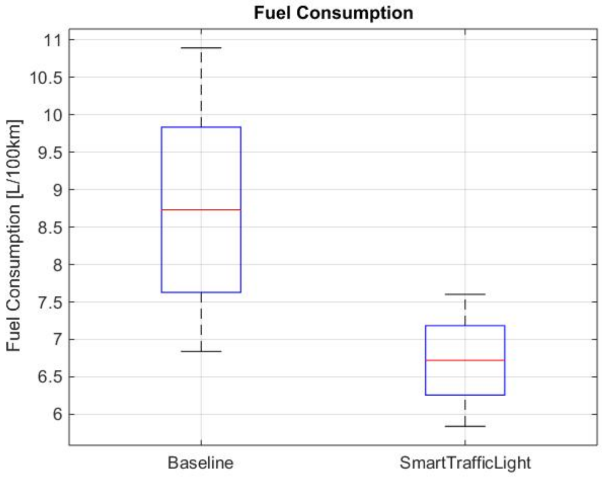


Figure 9-23. Fuel Consumption at smart traffic lights (Phabmacs)

9.3.5.2 Real live experiment on the BMW research car

The test pad for the vehicle collaborating in real life with WLAN 802.11p-equipped traffic lights was the city of Helmond, Netherlands. It is an ideal environment to conduct experiments to research ADAS functionality in combination with road operating infrastructure such as traffic lights. The experiments were performed on the BMW research car (see Figure 3-1) described in chapter 3.

The validation of the developed and researched concept was primarily focused on the following technical vehicle and infrastructure components:

- Vehicle-to-infrastructure communication (V2I, C2I) to maintain the vehicle – traffic light communication.

- Read-only of traffic light status messages and actions derived therefrom (recommended speed setting).

- Calculation and control of the vehicles recommended speed setting to obtain the optimum approaching/cruising speed to optimise harmonised traffic flow and fuel consumption.

For the validation itself the following conditions were chosen:

- Number of runs: 10 without and 10 with the system.
- With coasting mode, enabling driver to use the remaining kinetic energy of the moving car to approach the targeted intersection in the most economical way.

- Using core technologies such as: HMI (Head-Up Display and dashboard), messages, map, navigation, driving support, cooperative horizon.

- Data logger to obtain cruising data for analysis of the validation trips (SQLite).

- Multiple drive on the selected traffic lights test route in Helmond. The starting point is a bus stop on the east side of Helmond. The route crosses Helmond from east to west, then takes a U-turn west of Helmond to finish at the starting position, after crossing the city in the opposite direction. The total distance is about 14km (7 km each).

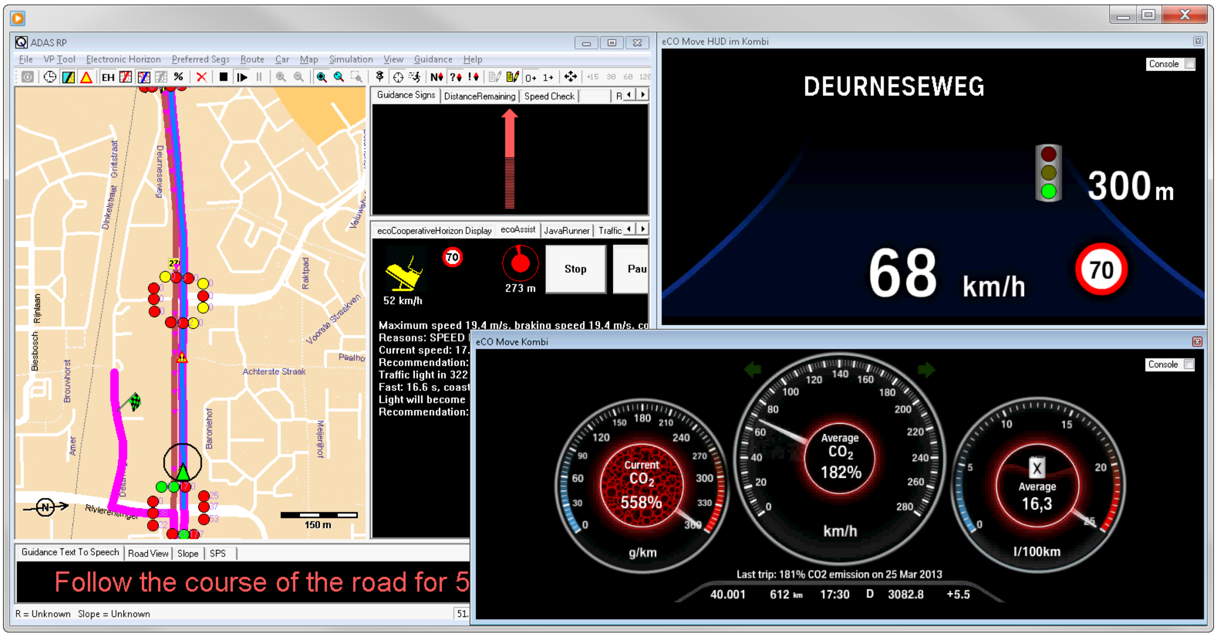
- Traffic density was similar on all runs. Weather conditions were about 20 degrees with normal weather and road conditions.

The number of stops in this urban run through Helmond is influenced by the cooperative traffic lights sending both their time to change and also a recommended speed and indeed there are fewer stops as expected. On this information, the driver can approach the traffic lights in an economically and ecologically optimised way, sometimes avoiding a full stop and waiting time at a red light intersections.

The screenshots in Figure 9-24 show the situation where the traffic light broadcasts its real time status and schedule, the vehicles on-board algorithms calculate the recommended strategy and thus the optimised recommended speed setting. For this computation position, current speed and distance to the intersection and the traffic light schedule has to be taken into account.

Recommended speed and distance to the traffic light and its status to be expected on arrival is visualized to the driver via the HuD. The HuD appears on a blue path horizon as an indicator of the most efficient way of driving while a red horizon indicates high emissions.

In Figure 9-24, different screens are displayed. The main ADASRP window (**A**dvanced **D**river **A**ssistance **S**ystem **R**esearch **P**latform) illustrates in its map view at the left-hand side the navigational horizon including the next traffic lights and the current status. In addition, some smaller surrounding windows with research support information such as turn advisor, or some technical details viewing the output of the control algorithms to compute the optimised speed are available in this view. In the right-hand upper window, the predicted traffic light status when approaching at the recommended speed is illustrated. In combination with the ACC (**A**daptive **C**ruise **C**ontrol) function, the recommended speed will automatically be passed to the speed profile server to automatically adapt and control the car's new cruising speed.


ADAS RP
File VP Tool Electronic Horizon Preferred Segs Route Car Map Simulation View Guidance Help
Guidance Signs DistanceRemaining Speed Check
ecoCooperativeHorizon Display ecoAssist JavaRunner Traffic
70
52 km/h
273 m
Stop
Pau
Guidance Text To Speech Road View Slope SPS
R = Unknown Slope = Unknown
150 m
eCO Move HUD im Kombi
Console
DEURNESEWEG
300 m
68 km/h
70
eCO Move Kombi
Current CO2 558%
g/km
Average CO2 182%
km/h
Average 16,3
l/100km
40.001 612 km 17:30 D 3082.8 +5.5

Maximum speed 19.4 m/s. braking speed 19.4 m/s. co
Reasons: SPEED
Current speed: 17.
Recommendation:
Traffic light in 322
Fast: 16.6 s, coast
Light will become
Recommendation:

Follow the course of the road for 5

Last trip: 181% CO2 emission on 25 Mar 2013

Figure 9-24. Cooperating vehicle and traffic light

At the centre of the dashboard, the normal speed and the average CO_2 consumption are displayed. The current CO_2 consumption is shown on the left-hand side and the fuel consumption on the right-hand side (in this accelerating case – showing up *red* for high energy consumption).

The dashboard further shows the red background due to the “Eco Mode” not being activated. Below the speed, the last trip emission is shown for comparison if the driver is driving the same route. The other two modes are designed equally but with the difference that the driver gets the indication whether he is driving in neutral or red mode (non-Eco mode). All the other vehicle icons are fully integrated. When the driver drives in the coasting mode, a coasting sign appears in the form of a blue sailing ship to inform the driver.

The same strategy for trip data analysis introduced in chapter 9.3.5.1 is also applied to the SQLite test data set obtained. Figure 9-25 illustrates the result of the test sequence. Similar to the outcome of the PHABMACS simulation (see oben), the graphic representation also shows the reduction of fuel consumption. The results differ only slightly.

About 10 trip traces were conducted without “Smart Traffic Lights” to obtain urban real life traffic data in terms of fuel consumption and CO_2 emissions. The “Smart Traffic Light” application was then set in action to execute the test under the research application including the special control algorithms.

The final effect on fuel consumption was about a 19% saving. This underlines the importance and potential of the reduction of fuel consumption and any related reduction of CO_2 emissions, particularly proved by this experiment.

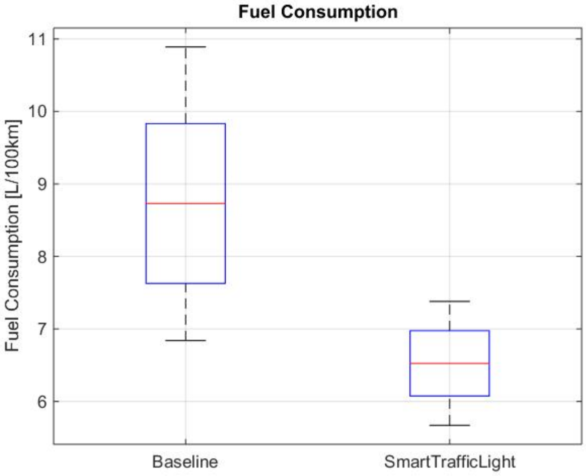


Figure 9-25. Fuel consumption with smart traffic lights (BMW car)

9.3.6 Conclusion

The fuel and CO_2 reduction of the application was achieved using the Eco driving functionality with respect to the coasting mode incl. the newly developed Eco Driving HMI.

The average fuel consumption and CO_2 improvements achieved with the Eco drive system in the PHABMACS simulation and the real life Helmond test route is about 25% with a minimum of 16.45% and a maximum of 31.95%.

Future applications are commonly based on the usage of existing and the generation of new data which makes advanced data mining and fusion functionalities inevitable. New approaches can be developed aiming to enable innovative cooperative services and augment their reliability, scalability and robustness. New data will comprise highly accurate positioning data leading to new functionalities e.g. by making lane-specific positioning possible or changing traction according to the presence of virtually fenced areas of city centres.

9.4 Drive-Guard ADAS weather assistant application

9.4.1 Motivation

Drive-Guard-s[11] major idea is to enhance the security of (semi-) autonomously driving vehicles (or those that use ACC). It alerts road users about changes in the weather conditions, enabling automated speed adjustment without involving the driver's interaction. Vehicles act as sensors in vehicular networks, private weather networks or road operators, collecting their weather recognitions in cloud-based service backends for further road weather data analysis. In case of weather hazards or degrading weather conditions, particular alert messages are broadcast to the vehicular community to enhance driving safety [123].

9.4.2 Introduction and technical base

The Drive-Guard’s related weather conditions are grouped in three different categories and handled differently in technical terms.

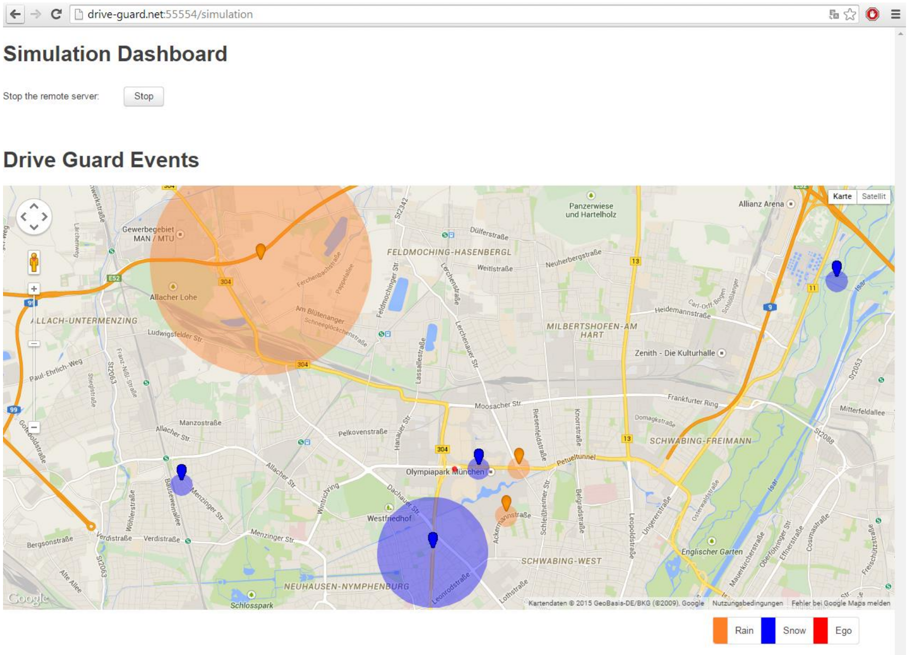
In addition, every weather condition has a range/coverage of leverage, not only spatial but also a temporal. The weather conditions are invalidated after a period of time (e.g. 30 min). All type of weather information is merged to one data set which represents the most accurate weather situation in a given area. Figure 9-26 illustrates special weather conditions defined in Munich with different spatial coverage and range.

1. **Manually:** The user can define a special weather condition on web-based GUI (http://drive-guard.net:55554/index) especially developed for the purpose as shown in Figure 9-26, or via a single click on a native Android Application while driving.

[11] Drive-Guard: Comprehensive ADAS Application with Web-UI, Smart-Device App, Cloud-Services and Vehicle Integration. http://drive-guard.net:55554/index

2. **Automatically:** The vehicle's sensors are able to recognise degrading weather conditions and send them back to our server in order to co-operate the other driver’s comfort and security. The transmission of information is carried out with the help of a LTE mobile network connector or alternatively the WLAN 802.11p NEC Linkbird connector.

3. **Public information:** Importing the information supplied by private weather stations and available in the cloud (see Figure 9-26).



drive-guard.net:55554/simulation

Simulation Dashboard

Stop the remote server: Stop

Drive Guard Events

🟧 Rain 🟦 Snow 🟥 Ego

Figure 9-26. Drive-Guard simulation dashboard

Each traffic member communicates with the Drive-Guard server via a TCP socket (see Figure 9-27). In this case, sockets are considered as the most beneficial communication type, because they do not introduce any traffic overhead (e.g. request Meta information, HTTP headers, etc.) and transmit only a minimal chunk of information. Of course, the total amount of information transmitted between the server and the clients strongly depends on the communication protocol used in the scope of the application.

An additional requirement is that the server and the client software may be implemented in different programming languages and run on heterogeneous platforms. Here the server is written in Java and uploaded on a Linux machine in contrast to the vehicle's client, which is a C++ application and runs on Windows PC. In order to resolve the above-mentioned issues, the ProtoBuf[12] technique is used to serialise the protocol messages.

[12] ProToBuf: https://developers.google.com/protocol-buffers/docs/overview

On the other hand, the project Drive-Guard protocols task contains the protocol message definitions themselves as well as their automatically generated implementation.

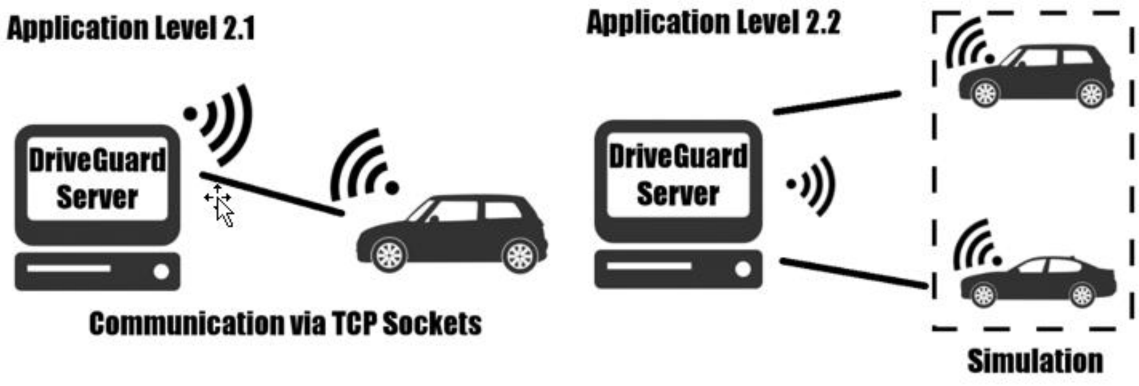


Figure 9-27. a) Desired real world scenario b) Current simulation scenario

An example of how to generate the Java protocol using “protoc” (ProtoBuf compiler) is shown as illustrated below:

- Clone the Drive-Guard protocol project from BitBucket
- Go to the root folder of the project
- Ensure you installed the “protoc” and added it to the system PATH
- To create the protocol source files, type in the cmd \terminal the command “protoc -I=.\resources --java_out=.\src .\resources\dg-weather.proto”
- To use the protocol in other projects without depending on a Drive-Guard protocol project, we need to generate the jar. Ensure you have installed a maven2. Type in the cmd/terminal “mvn install” again
- The jar file is generated under <path>\target\ DriveGuardProtocols-0.0.1-SNAPSHOT.jar and you can reference it in other projects

In order to generate a common traffic flow and to test the robustness of our application, the PHABMACS will be used and integrated (see Figure 9-27 b).

9.4.3 Drive-Guard simulation

The Drive-Guard simulation is based on the PHABMACS software. The following steps are necessary to integrate the simulator into the individual Drive-Guard research software system.

9.4.3.1 Software

Development platform is Eclipse and JDK 1.7. To start the development and integration the following steps have to be performed:

- Download the simulator project from Bitbucket
- Import the recently cloned project in Eclipse. (Unpack Phabmacs_Release to a folder and create new project by: “Import->General->Existing Projects into Workspace” browse to the unpacked folder -> finish. Make sure all libs and Java are bound to your Eclipse project correctly)
- Runnable examples can be found in de.objective.phabmacs.wetterassistent. They mostly end with the suffix “Starter”. In order to start, just click on the run button. Another way to start the application is directly from the .bat file which basically sets the classpath with its all dependencies and starts the main class. An example of such .bat files can be found in the main folder of the project and has the content:

  Example of the .bat:

  ```
  set PHABMACS_CP=^
  ./libs/DriveGuardProtocols-0.0.1-SNAPSHOT.jar;^
  ./libs/phabmacs-2.0.4-SNAPSHOT.jar;^
  ./libs/sp2.fac.vdpapi-2.3.1.jar;^
  ./libs/AppleJavaExtensions.jar;^
  ./libs/commons-math3-3.2.jar;^
  ./libs/jbullet.jar;^
  ./libs/jinput.jar;^
  ./libs/jollada.jar;^
  ./libs/lwjgl.jar;^
  ./libs/lwjgl_util.jar;^
  ./libs/tinylog.jar;^
  ./libs/vecmath.jar;^
  ./bin
  java -Xmx1024m -classpath %PHABMACS_CP%
  de.objective.phabmacs.wetterassistent.WetterassistentStarter
  ```

9.4.3.2 PHABMACS Handling

Driving

W,a,s,d on the keyboard are mapped to steering, brake, and throttle. M and n can be used for changing gears between forward, neutral, backward. You can also use a gaming steering wheel with pedals (Logitech is bound, others can be added in the Java code). For this purpose, use the menu inside the simulator window while the simulator is running. “control->steering wheel”.

Camera view

The camera can be switched between steady and chasing view via menu “view->camera”. Zoom: mouse wheel, rotate: left click + mouse move, pan. (only in chasing mode) right click + mouse move.

Performance

gfxEngine.props -> set glFsaa=4 to “0” to disable anti aliasing

gfxEngine.props -> set glVersion=3 to “2” to disable shadows

9.4.3.3 PHABMACS customisation to the needs of Drive-Guard

Maps

An exported map from Open-Street-Map (OSM) can be directly integrated into the simulator. To export a map, go to “http://www.openstreetmap.org/export”. A recently exported OSM can be added to the simulation by the following lines of code:

```
// load the map
OsmMap map = new OsmMap("maps/muenchen_angepasst.osm");
map.buildMap();
```

In order to mock up more complex and specialised driving scenarios, anyone can modify the OSM infrastructure with the help of a tool called *JOSM* (Java-OpenStreetMap-Editor). Please note that modifying of the infrastructure includes changing the lane number and event adding of a new street as well.

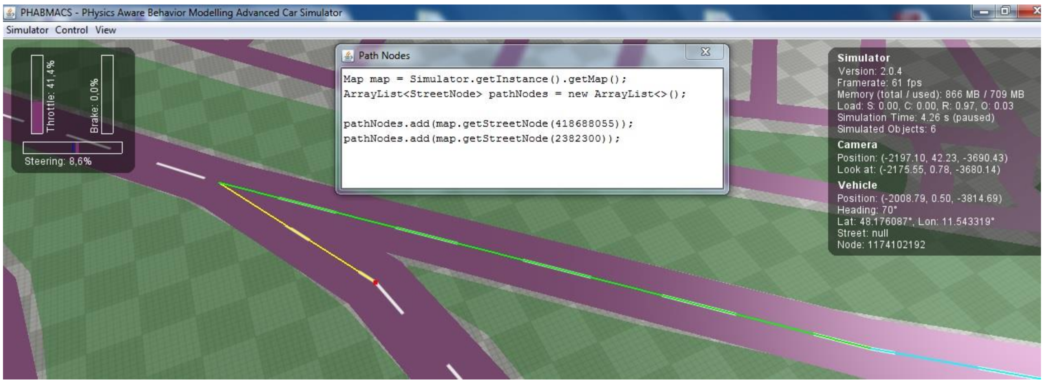
Driver types

There are mainly three types of drivers within the PHABMACS V2.05 simulation:

- ***RouteDriver*** follows a stop points describing its driving route
- ***InputDriver*** can be controlled by a user using the keyboard’s arrows
- ***CityDriver*** is a traffic light aware driver’s implementation, performing the main task of correct driving within the city boundaries

Street infrastructure

As mentioned above, the user is able to specify the route of the vehicle manually. This is done by using the so called street nodes and street edges of the street infrastructure. Actually, taking a closer look, it is obvious that the OSM describes its street curvatures upon them both. The street is divided in parts, whereby the parts at curves are smaller in order to be able to precisely describe the path. In Figure 9-28, the red point is a street node (2 nodes in a part – by one on both sides) and the street edge is the green/yellow line (along every part). For more details regarding this topic, see Figure 9-28.



```
Map map = Simulator.getInstance().getMap();
ArrayList<StreetNode> pathNodes = new ArrayList<>();

pathNodes.add(map.getStreetNode(418688055));
pathNodes.add(map.getStreetNode(2382300));
```

```
Simulator
Version: 2.0.4
Framerate: 61 fps
Memory (total / used): 866 MB / 709 MB
Load: S: 0.00, C: 0.00, R: 0.97, O: 0.03
Simulation Time: 4.26 s (paused)
Simulated Objects: 6
Camera
Position: (-2197.10, 42.23, -3690.43)
Look at: (-2175.55, 0.78, -3680.14)
Vehicle
Position: (-2008.79, 0.50, -3814.69)
Heading: 70°
Lat: 48.176087°, Lon: 11.543319°
Street: null
Node: 1174102192
```

Figure 9-28. PHABMACS Street Nodes and Open Street Map

To add a route to the ***RouteDriver***, there is an option named "Pick Street Nodes" directly in the simulator. The menu "Pick Street Nodes" is found under "View" → "Overlays". Simply choose some street nodes where the route should pass. Copy the path nodes from the extra window and add these to the code as illustrated below:

Example for the "Mittlerer Ring, Munich":

```
private final ArrayList<StreetNode> mittlererRing = new ArrayList<>();

// create the route "Mittlerer Ring"
createMittlererRingRoute(map);

private void createMittlererRingRoute(OsmMap map) {
    this.mittlererRing.add(map.getStreetNode(1174101563));
    this.mittlererRing.add(map.getStreetNode(472164));
    this.mittlererRing.add(map.getStreetNode(2940935732l));
    this.mittlererRing.add(map.getStreetNode(429833206));
    this.mittlererRing.add(map.getStreetNode(2963232106l));
    this.mittlererRing.add(map.getStreetNode(1616721169));
    this.mittlererRing.add(map.getStreetNode(564149));
    this.mittlererRing.add(map.getStreetNode(29039642));
    this.mittlererRing.add(map.getStreetNode(2388715717l));
    this.mittlererRing.add(map.getStreetNode(3205485428l));
    this.mittlererRing.add(map.getStreetNode(1262146256));
    this.mittlererRing.add(map.getStreetNode(2779899419l));
    this.mittlererRing.add(map.getStreetNode(20925331));
    this.mittlererRing.add(map.getStreetNode(1460310634));
```

```
        this.mittlererRing.add(map.getStreetNode(27510728));
        this.mittlererRing.add(map.getStreetNode(643899398));
        this.mittlererRing.add(map.getStreetNode(472164));
    }
```

The street nodes are created by import of the OSM map.

Connection to the communication server – RemoteAwareDriver

There are several types of drivers implemented in the PHABMACS 2.05 simulation as described above. They all perform some specific task. From this point of view something similar to semantics is needed - the goal is to provide a driver type, which is able to connect to the communication server, to read and write to it and adjust the vehicle's speed on request. In order to achieve these objectives, the **RemoteAwareDriver** as an extension of the **CityDriver** has been implemented.

```
public class RemoteAwareDriver extends CityDriver implements
             DriveGuardProtocolListener
```

DriveGuardProtocolListener provides methods to define what should happen when the RemoteAwareDriver receives a WeatherCondition or GeoLocation protocol message on the socket - or in other words defines the driver's behaviour on protocol message receive. Since the listener is called within the **RemoteAwareDriver** class itself and not asynchronously from outside, its practical importance is only to improve the readability and understandability of the **RemoteAwareDriver**:

```
    onWeatherConditionReceived(WeatherCondition msg)
    onGeoLocationReceived(GeoLocation msg)
```

Take care to ensure that the communication server can deliver only information about the weather condition so that the second listener's method remains unimplemented.

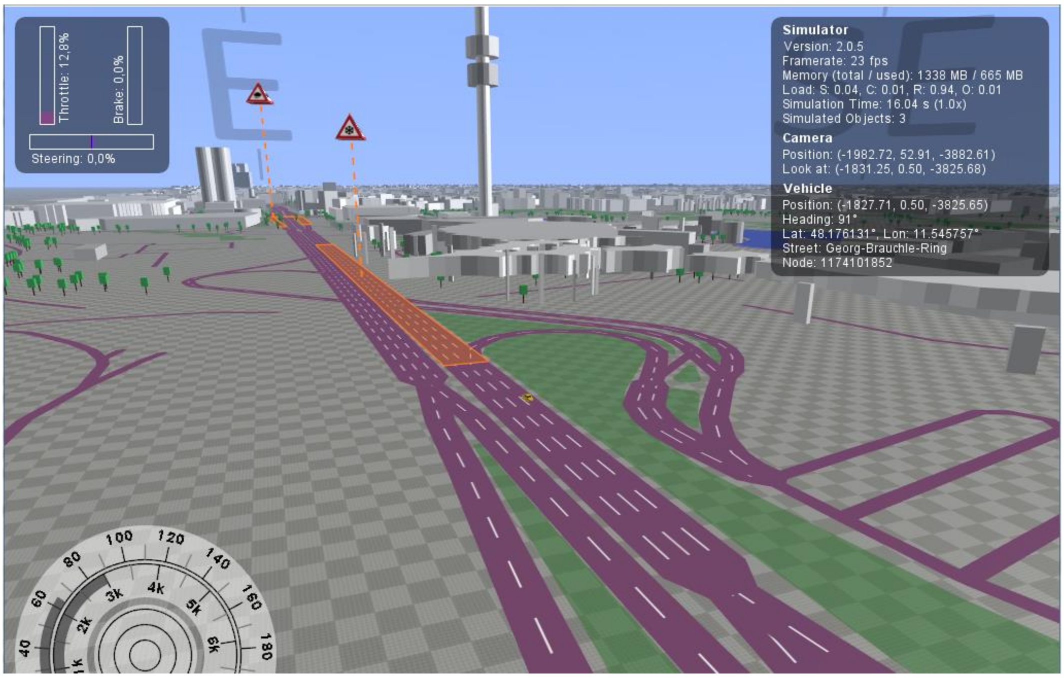
As mentioned many times above, communication is realised via TCP sockets. The most important point is that the RemoteAwareDriver starts threads for reading and writing to the server in order to keep the main execution thread free and prevent it from blocking.

Visualisation of a weather element on the road

In order to get a more precise notion about where a vehicle should slow down and increase its velocity, an orange polygon is visualised on the street as well as a traffic sign indicating the type of the weather condition (see Figure 9-29). Since this feature is very useful when developing and testing innovative systems and because there is no Java documentation, the reader will find a brief summary of how to use it below:

```
PolygonEventArea area =
   PolygonEventArea.onStreet(nodes.get(0), edges, 1, 430, 0, 3);

EventIndicator event = new EventIndicator(EventType.ICE, area);

map.addEventIndicator(event);
```

The first argument of the static method PolygonEventArea.onStreet is the street node, where the orange polygon should start from. The second element is the street edge which it should cover. The third argument is unclear but works perfectly with the value of 1. The fourth argument is the length of the polygon in metres. It is useful to fill all of the street edges to pass but only the first N (e.g. 430) metres of them. The last two arguments specify the width of the polygon and have the semantic of “from lane number” and “until lane number respectively”. The lanes have ascending numbers from 0 as lane furthest to the right. In this case, all 4 required lanes have to be coloured – from lane 0 (furthest to the right) to lane 3 (furthest to the left).



```
Simulator
Version: 2.0.5
Framerate: 23 fps
Memory (total / used): 1338 MB / 665 MB
Load: S: 0.04, C: 0.01, R: 0.94, O: 0.01
Simulation Time: 16.04 s (1.0x)
Simulated Objects: 3
Camera
Position: (-1982.72, 52.91, -3882.61)
Look at: (-1831.25, 0.50, -3825.68)
Vehicle
Position: (-1827.71, 0.50, -3825.65)
Heading: 91°
Lat: 48.176131°, Lon: 11.545757°
Street: Georg-Brauchle-Ring
Node: 1174101852
```

Figure 9-29. Visualization of a weather element on the road

On the other hand, the **EventIndicator** specifies the type of traffic sign floating above the road. Possible values are only **EventType.ICE** and **EventType.ROAD_WORKS**. They are hard-coded and cannot be changed. This means that the drawing of the traffic sign looks in ./assets/textures for one of both images with names roadworks.png or ice.png.

A sample code snippet of the drawing of the first polygon is shown below:

```
RouteFinder f = new RouteFinder();
StreetNode start = map.getStreetNode(1174101852l);
f.findRoute(start, map.getStreetNode(2940935730l));
ArrayList<StreetNode> nodes = f.getRoute();
ArrayList<StreetEdge> edges = new ArrayList<>();

for (int i = 0; i < nodes.size() - 1; i++) {
      edges.add(nodes.get(i).getEdgeTo(nodes.get(i+1)));
}

PolygonEventArea area =
   PolygonEventArea.onStreet(nodes.get(0), edges, 1, 430, 0, 3);
EventIndicator event = new EventIndicator(EventType.ICE, area);
map.addEventIndicator(event);
```

9.4.4 Drive-Guard android app

Description

The application is used to send the GPS location and the event to the Drive-Guard web application. The **LocationListener (android.location.LocationListener)** acquires the geo position (longitude and latitude) using the Android GPS API. This API provides access to the system location services. These services allow the application to obtain periodic updates of the device's geographical location. Additionally manual input of the geo position is possible.

For the further development of the application, Eclipse with installed ADT plugin or Android Studio can be used.

Front-end

To edit the layout go to **driveguard_client1 -> app -> src -> res -> layout -> activity_main.xml**. To render the image buttons on the screen, dynamic rendering is used: go to **driveguard_client1-> app-> src-> de.objective -> driveguard-> GPSTracker-> ImageAdapter**.

Back-end

To send a POST request to the server the next URLs should be used:

- **POST** http://drive-guard.net:55554/addRain
- **POST** http://drive-guard.net:55554/addSnow

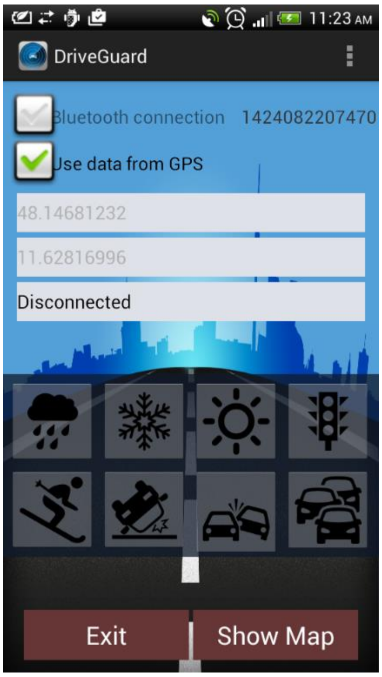
The POST request parameters are as follows:

- Latitude: String
- Longitude: String

The sent requirement is implemented on the **Asynctask** thread because if an application attempts to perform a networking operation on its main thread **NetworkOnMainThreadException** arises. This happens due to the fact that the webserver is taking too much time to respond and causes the main thread to become unresponsive. To avoid this, the **Asynctask** thread is used. A code snippet for Drive-Guard's event messaging is shown below:

```java
private class DriveGuardEventSending extends AsyncTask <String, Void, Integer> {

    @Override
    protected Integer doInBackground(String... args) {
        int code = 200;
        try {
            String url = args[0];
            HttpClient client = new DefaultHttpClient();
            HttpPost post = new HttpPost(url);
            List<NameValuePair> urlParameters = new ArrayList<NameValuePair>();
            String LatitudeSend = MainActivity.latEditText.getText().toString();
            String LongitudeSend = MainActivity.lonEditText.getText().toString();

            urlParameters.add(new BasicNameValuePair("Latitude", LatitudeSend));
            urlParameters.add(new BasicNameValuePair("Longitude", LongitudeSend));
            post.setEntity(new UrlEncodedFormEntity(urlParameters));

            HttpResponse response = client.execute(post);

            if(response.getStatusLine().getStatusCode() != 200) {
                System.out.println("Error!"); // TODO retry sending
            }
            code = response.getStatusLine().getStatusCode();
        } catch (Exception e) {
            System.out.println();
        }
        return code;
    }

}
```

The main screen of the Drive-Guard Android app is presented in Figure 9-30. Typical message type buttons are arranged on the screen below the longitude/latitude and connection status fields. The show map button makes it possible to switch to the map view for navigational and orientation purposes.



11:23 AM

DriveGuard

□ Bluetooth connection 1424082207470

☑ Use data from GPS

48.14681232

11.62816996

Disconnected

Figure 9-30. Grive-Guard android app

9.4.5 Drive-Guard web-application and communication server

Software requirements

Java Play! Framework 2.1.0, JDK 1.7, MySQL Workbench 6.2, putty, Eclipse. To start with development, please follow the instructions:

- Download the Drive-Guard-server project from BitBucket
- Go to the root folder of the recently cloned project in the terminal/cmd
- Type “git eclipse” to make the project importable
- Import the project in Eclipse
- To start the project type ”git play” in the cmd/terminal

Additional software needed:

- Download and install MySQL Workbench (my personal preference is version 5.X but feel free to use 6.2+)
- Maven
- Git

- Protoc[13]

Remark: Ensure, that the Play! Framework[14] is property installed beforehand.

Drive-Guard Server Architecture

The Drive-Guard server is a Play! Framework 2.1.0 Application, written in Java and compiled on jdk 1.7. Please note that the current Play! Framework version 2.3.x (in this number v2.1.0 also) contains a bug in combination with jdk 1.8 and therefore does not operate properly.

The common play application follows the Model-View-Controller (MVC) architectural pattern (see Figure 9-31). This pattern splits the application into the following separate components:

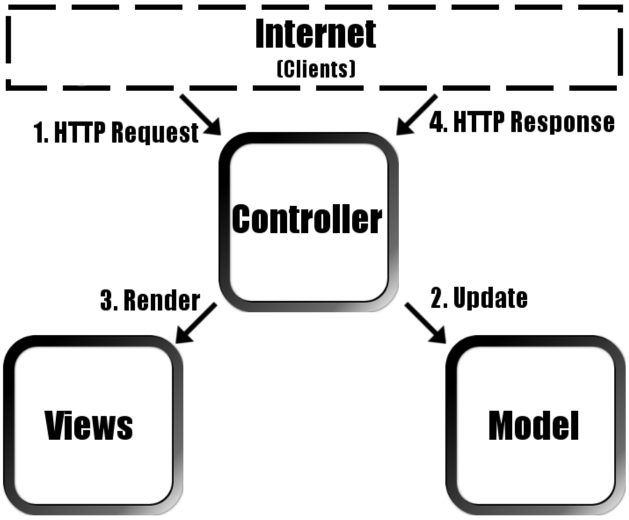


Figure 9-31. Common Play! Framework application structure

- The Model is the domain-specific representation of the information on which the application operates. Domain logic adds ‘meaning’ to raw data (e.g., checking if a weather condition is already expired/still valid). Most applications use a persistent storage mechanism such as a database to store data. MVC does not specifically mention how this information to be stored and later accessed (do not provide an access layer), because it is understood to be underneath, or encapsulated by, the Model.

- The View renders the model into a form suitable for interactions, typically called a (graphical) user interface. Multiple views can exist for a single model according to the

[13] https://developers.google.com/protocol-buffers/docs/overview

[14] https://www.playframework.com/documentation/2.0/Installing

purpose it is used for. Play! Framework v2.1.0 uses a Scala template engine, responsible for HTML generation. Supplementary types are XML, JSON or even some cases where the view can be expressed in binary form, e.g. dynamically rendered chart diagrams.

- The Controller responds to events (typically user actions and in a web application mainly HTTP requests) and processes them. This means that the Controller module is the part where the business logic performs. The common scenario is that Controller listens for HTTP requests, extracts relevant data from the ‘event’, such as query string parameters, request headers, etc. and applies changes on the underlying model objects.

However, the Controller’s function - mapping the HTTP request - is not the most suitable part for executing the application’s business logic because these are hard to mock up and therefore entirely untestable. A much better option would be to split the Play! Controller responsibilities inti two distinct tiers – the first one responsible for the GUI presentation to the clients and the other one responsible for performing of business-specific operations. Such a split can be made also within the Play! Model in order to extract the more complex model specific operations and to enable us to use the model classes as a POJOs. This ensures that our model classes are detached from any kind of vendor specific technologies. The Drive-Guard server architecture is designed following the above-mentioned good practices and the recommendations and has the following structure (see Figure 9-32).

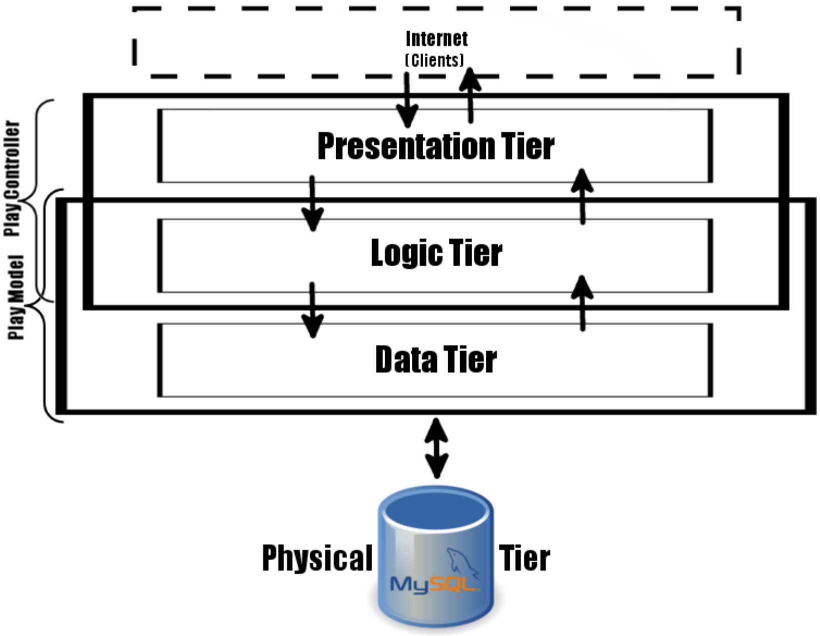


Figure 9-32. Improved Drive-Guard server architecture

The Tier-alike architectural pattern allows every layer to be exchanged without having a negative influence on the rest of the application layers. For instance, the problem to perform a high number of spatial operations and therefore, it makes sense to use directly a database/storage server, implementing internally the spatial functionality (e.g. Apache Solr(link), MsSQL Server and not MySQL like now). In practice, the only think needed to adjust is the methods in the last Data Tier, direct connecting to the physical tier.

Web application server management

The Drive-Guard web application is located physically on a machine with IP 78.46.1.87. The global public DNS resolves this IP to the host drive-guard.net. Any change on the machine can be performed directly via a ssh command by using the login credentials:

Host: drive-guard.net
Port: 222

Once logged onto the machine, the folder Drive-Guard server appears, which contains the project itself. For simplicity, in the home directory there are two executable shell scripts with the names **start-drive-guard-server.sh** and **stop-drive-guard-server.sh**, starting and stopping the web application respectively as their names imply.

In rare cases, the **start-drive-guard-server.sh** may result in an error due to incorrect order in which the Scala and Java files are compiled. In this event, all that needs to be done is to start the script again. Note that the start script does not stop the application in advance so it needs to be stopped manually.

The web application is accessible through the browser by invoking http://drive-guard.net:55554/index. It is necessary to call “**git pull**” in order to update the web application.

9.4.5.1 Web application implementation

The detailed architecture shown in Figure 9-32 will be described below in more detail beginning with the presentation tier, thereafter the logic tier and last but not least the data tier on a MySQL database system.

Presentation tier

The presentation tier is composed of the Play! Framework controllers (it wraps the servlet functionality or more precisely it delegates the servlet requests into a higher level of abstraction) and is therefore not highly sophisticated. There is currently only one controller in the web application called Application and its request mapping is described by specification in the conf/routes file. The file conf/routes can be considered as a natural API description: (<- API)

```
•  # Starts the DriveGuard communication server
   POST   /start                      controllers.Application.start()
•  # Stops the DriveGuard communication server
   POST   /stop                       controllers.Application.stop()
•  # Inserts into a database a RAIN weather condition with a default coverage
    of 150 meters ATTENTION: This API call requires the string parameters "latitude" and "longitude" in the request body
   POST   /addRain             controllers.Application.addRainEvent()
•  # Inserts into a database a SNOW weather condition with a default coverage
    of 150 meters ATTENTION: This API call requires the string parameters "latitude" and "longitude" in the request body
   POST   /addSnow         controllers.Application.addSnowEvent()
•  # Returns a json list of all DriveGuardMessage available in the database
   GET /events          controllers.Application.events()
•  # Downloads the video demo file
   GET /demoUrl         controllers.Application.downloadDemo()
•  # Disable a job if it is enabled, or enabled it in case the job is disabled
   POST  /changejobsetting      controllers.Application.changeJobSetting(job:
   String)
•  # Runs once a job by its id from the database
   POST  /runjob        controllers.Application.runJob(job: Long)
•  # Deletes a weather condition by id of the marker pinned on the google map
   GET /deletemarker    controllers.Application.deleteMarker(id: Long)
```

Logic tier. The business logic

The business logic of the web application is located in the logic tier and separated in encapsulated services. Each service holds a cluster of functions performing similar logic. Every service is implemented by an interface, where all of its functions are declared, and a concrete implementation of this interface. This coding style gives the ability to test the application more easily and more effectively in terms of unit or integration testing, because for each interface a Stub/Mock implementation can be provided. This guarantees that the test does not instantiate the entire application, avoids side effects and runs more quickly. In addition, the separation of services into an interface and a concrete implementation enhances the entire readability, maintenance and exchangeability of functionality.

Currently the two following services exist in the Drive-Guard web application:

- **DriveguardEventService** – performs operations with **DriveGuardMessage** objects, in this number the spatial/geo search for special weather conditions by a given geo location. It should not to be forgotten that the weather conditions have different coverage and come from various sources. So it is also possible that the **DriveguardEventService** considers multiple weather conditions from the database in order to determine the one, which should be sent back to the client (vehicle). Currently every weather condition has a criticality value (CLEAR = 0, RAIN = 100, SNOW = 1000) and **DriveguardEventService** answers exactly with the weather condition, having the highest criticality. (← **Weather Evaluation**)

Despite the fact that the geo search logic can be directly implemented in the controllers, the usage of a service enables exposure of this information in the presentation layer without breaking the main concept of layer architecture (a layer can interact only with the direct neighbour's layers).

- **WeatherUndergroundService** – collects information about the special weather conditions from stationary personal weather stations (PWS). The idea is to diversify the sources of information in order to make a more reliable and precise decision. This is the place where a connection with the WeatherUndeground[15] API is realised. The WeatherUndeground API works with the help of simple HTTP GET methods and returns JSON or XML responses. Since manipulating a text-based response directly is often hard, error prone and time consuming, it is preferable to transform it into a Java object. This can be accomplished by using a JAXB[16]. The main idea is to describe the structure of the XML response once and store it in a .xsd[17] file. The .xsd file itself should be located in conf/xsd. A Maven2 plugin generates the JAXB classes which allows XML binding (just to type "mvn install" in the cmd/terminal of the project's root directory). Figure 9-33 shows a sequence diagram of WeatherUndeground API's workflow.

[15] http://www.wunderground.com/weather/api/

[16] https://jaxb.java.net/

[17] http://www.freeformatter.com/xsd-generator.html

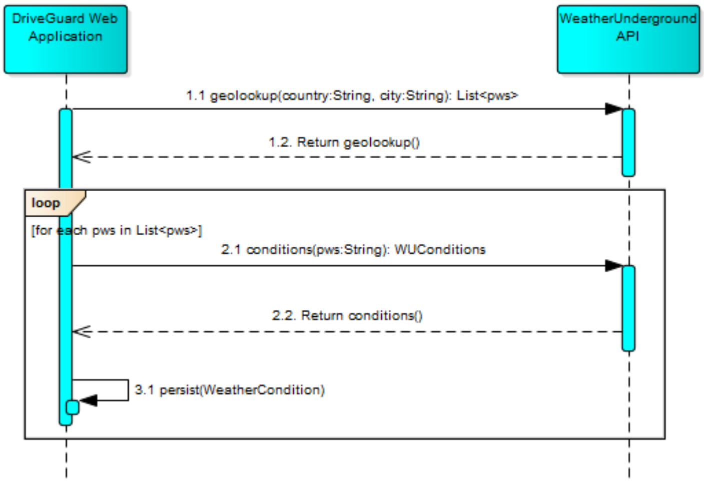

DriveGuard Web Application
WeatherUnderground API
1.1 geolookup(country:String, city:String): List<pws>
1.2. Return geolookup()
loop
[for each pws in List<pws>]
2.1 conditions(pws:String): WUConditions
2.2. Return conditions()
3.1 persist(WeatherCondition)

Figure 9-33. WeatherUnderground API workflow

Data tier

The data tier holds only the **DriveGuardMessage** and **JobsSettings** models.

The **DriveGuardMessage** represents a special weather condition with its type (SNOW, RAIN, CLEAR = NONE), the geo position of its origin, the insertion date and more meta information. **DriveGuardMessage** indeed is the place where the spatial search is realised. It is implemented with a simple HQL (Hibernate Query Language) statement, computing the earth surface distance from a geo location to the centre of each weather condition and selecting only those with a sufficiently small distance: (← **Spatial Search**)

```
public static List<DriveGuardMessage> findNearby(
             double distance, double originLatitude, double originLongitude
) {
      String HQL = String.format("(
      6371 * 1000 * Acos(Sin(radians(latitude)) * Sin(radians(%.6f)) + Cos(ra-
      dians(latitude)) * Cos(radians(%.6f)) * Cos(radians(%.6f) - radians(longi-
      tude)))
      ) < %.3f",
      originLatitude, originLatitude, originLongitude, distance);
   return DriveGuardMessage.find.where(HQL).findList();
}
```

The **JobsSettings** is described in the Jobs section.

Jobs (periodical task)

The Drive-Guard application requires the periodical execution of specific operations (so- called "jobs") in order to ensure the correct execution of the entire system. Examples of such periodical operations are the insertion of information from WeatherUnderground API or the invalidation of expired weather conditions. The Play! Framework 2.X integrated in scheduler is used for this purpose, which is based on the Akka[18] toolkit. On an application start, the Akka decides which jobs should be registered in the system and t which time interval they should be executed, but dynamically determines if to do this or not. This gives the flexibility to stop and start different jobs without the need to restart the entire application.

The database the job settings are kept for this purpose (a qualified name of the job class itself, the time interval and a status weather to execute it or not). Therefore, if a new job is developed, everything needs to be persisted correctly to the database. It can be also checked on the http://drive-guard.net:55554/simulation.

The job itself is a simple class which extends **tasks.jobs.DGJob**. It is not bound to any specific location in the project, since the information of its location is provided via its qualified name and stored in the database. However, in order to ensure a project's structure consistency it is strongly recommended to locate it in **tasks.jobs** package.

Remark: After the introduction of a new job, it is necessary to restart the web server so that the Akka toolkit registers it on application start.

Communication server

As mentioned in section 9.4.5, every traffic member connects to a server via TCP/IP socket and listens on port 55555. Please note that this server is different from the web application, but also runs in the context of the Play! Framework web container. There exist a number of possible ways to implement server with the desired functionality. The most naïve approach is to listen in the main thread for client connections and to dedicate a separate thread to each new one. It is obvious that at specific point of time the server will explode due to the high number of simultaneously running threads. So it would be a better idea to limit the possible client thread number but to ensure robustness of the Drive-Guard system. A thread pool in this case seems to be the most meaningful implementation. In the scope of the Drive-Guard project, we implemented a customised thread pool - **tasks.pools.DGThreadPool**. It has two very important characteristics in contrast to the standard thread pool implemented in Java 5+:

- Provides means for the interruption of infinite threads in a secure manner, where the user can decide at which point in time and how to stop a thread. This avoids all problems which may result from the standard interruption of the thread – an operation to be half-ended. (check the class **tasks.DGPoolTask**)

[18] http://akka.io

- Restarting of the thread pool, which is not supported in the Java standard.

Recommendation: Communication server to be implemented not using a thread pool but using a round-robin scheduler. Of course it is necessary to take into account the pros and cons of both techniques (thread pool vs. round-robin scheduler).

Security

Security is provided by the native Play! Framework authentication mechanism[19].

9.4.6 ADASRP vehicle plug-in

9.4.6.1 Introduction

ADASRP[20] is an **A**dvances **D**river **A**ssistance **S**ystem **R**esearch **P**latform for research, development and testing of new Apps for driver assistance systems of various types of applications. The ADASRP Research Platform v2011 is used for this research, installed and currently running on a 32-bits Windows PC in the trunk of the vehicle. Every single ADAS is actually realised as a plugin for the RP and therefore should extend the CEHPlugIn. The ADASRP computer is directly connected to the CAN bus of the vehicle, where different messages flow in order to notify the hardware components of a change of the state and even to physically control them (Figure 9-34). Since CAN messages are just a byte frame and therefore are very hard to be interpreted by a developer, there is a plugin called MSGDecoder which translates them into a higher level of abstraction suitable for further development.

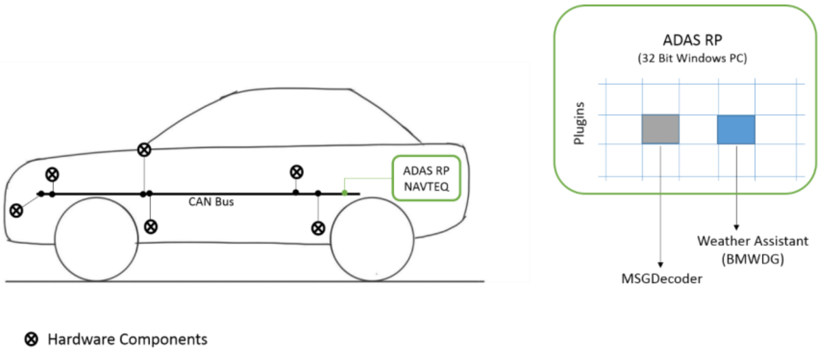


Figure 9-34. Controlling the vehicle by means of the ADASRP plug-in

[19] https://www.playframework.com/documentation/2.1.0/JavaGuide4

[20] http://www.here.com

9.4.6.2 Software setup

The following software prerequisites have to be fulfilled to develop and deploy any new plugins for the research vehicle: Visual Studio v2010, boost library and the ADASRP installation. To start with development, the following instructions must be followed:

- Copy from the Objective HardDrive (\\OS-FILE\Entwicklung\München\Wetter Assistent) the ADASRP.Installation, and BMW plugins for ADASRP. The folder is named “BMW”.
- ADASRP.Installation creates a folder NAVTEQ in the selected location of installation.
- Copy the BMW ADASRP plugins in the folder <path>\NAVTEQ\ADASRP \ADASRP.PlugIns.
- Install VisualStudio v2010. Some external dependencies are compiled on v2010 and therefore adhere to this version as well. Additionally MFC which comes with Service-Pack1 (SP1), so install SP1 for VS2010 as well.
- Compile boost library and place it somewhere on your local machine. Note that the compilation should be carried out in version 2010 as well (VS=10). The compiled version can be also found in the Objective HardDrive (\\OS-FILE\Entwicklung\München\Wetter Assistent) under “boost”.
- Add into your environment the variable with name ADASRP and value <path>\NAVTEQ\ADASRP\ADASRP.2011.
- Go to <path>\NAVTEQ\ADASRP\ADASRP.PlugIns\BMW and open the VS10 solution file.
- Adjust the declarations of boost to your project settings:
  - Right click -> Properties -> C/C++ -> General
  - Add <path to boost> in Additional Include Directories
- Adjust the path to boost used by the linker:
  - Right click -> Properties -> Linker -> General
  - Add <path to boost>\stage\lib in Additional Library Directories

- Rebuild the BMWDG/BMWMOA project.

Attention: After building the BMWDG plugin in release mode, in the folder <path>\NAVTEQ\ADASRP\ADASRP.2011\bin\Win32\ADASRP are generated 4 files with the same name (1 dll, lib file, etc), which will be used from the ADASRP to visualise and start the plugin itself later on. The ADASRP application executable file is ADASRP.exe (the file to start the application from) in the same folder, but it cannot be started right away because it requires an input configuration file (.ini). For this purpose you can create a shortcut for the ADASRP.exe and adjust its “Goal” field/value in the properties section, e.g. "C:\Program Files (x86)\NAVTEQ\ADASRP\ ADASRP.2011\bin\Win32\ADASRP\ADASRP.exe "C:\Users\<current-user>\Desktop\ ADASRP.BMW.ini.

Attention has to be paid to the plug-in section:

Table 3. ADASRP vehicle plug-in setting

[xPlugIns0] 0=EHPIAHLog 1=EHPIAHAttributes 2=EHPIStats 3=EHPIAHTrafficSign 4=EHPICanvas 5=IClientMonitor	[PlugIns0] [PlugIns1] [PlugIns2] 0=BMWDG [PlugIns3] 0=EHPIVPSensorDisplay

Pay particular attention to the [PlugIns2] section, where each plugin name should exactly match the name of the .dll file (the plugin you want to start) from the <path>\NAVTEQ\ADASRP \ADASRP.2011\bin\Win32\ADASRP folder. In the concrete case, the .dll file matches the name of the project (BMWDG) and therefore in [PlugIns2] same name has to be used as well.

9.4.6.3 Drive-Guard weather assistant plug-in

The Drive-Guard weather assistant plug-in (named **BMWDG**) looks like the illustrations of a few most important code snippets:

```
class CBMWDG : public CEHPlugIn, public BMW::MSGDecoder
{

public: // Constructor/Destructor
   CBMWDG(const Context& context);
   virtual ~CBMWDG();


public: // Listening flags
   virtual bool isAHServiceListener()      { return false; };
   virtual bool isVPServiceListener()      { return false; };
   virtual bool isSensorServiceListener()  { return false; };
   virtual bool isRoutingServiceListener() { return false; };
public: // CEHPlugIn overridden methods
   virtual void onUpdate(Sint32 lHint, void* pHint);


public: // Overridden BMW::MSGDecoder
    virtual void on(const BMW::NAV_GPS1&     msg);
    virtual void on(const BMW::DISP_LDM_1&   msg);
    virtual void on(const BMW::RCOG_TRSG&    msg);
    virtual void on(const BMW::OP_CCTR&      msg);

private:
   void sendCAN();
   // Other worker methods come here …

private: // Data members
   WeatherAssistantModule*        waModule;
   StateMachine*           stateMachine;
   CloudInformation::FlorisMessage    mFlorisMessage;
   // Other data members come here …

   CButton              m_btnWAOn;
   // Other MFC elements …

};
```

The functions overridden from the BMW::MSGDecoder determine the behaviour of the plugin when a specific message is received from the CAN bus. These functions are used as they are. The MSGDecoder is used as a black box, because this research was not directly involved in the process of CAN message (re)marshalling and interpretation. However, in terms of sending a specific message to the other hardware components via the CAN bus of the vehicle, there is no automation procedure available which does everything instead. The definition of the function sendCAN() shows how to send a CAN message, increasing or decreasing the ACC speed. All other CAN messages can be sent the same way. The idea behind this is fairly simple and proceeds in three steps:

1. Initialise the required CAN message

2. Convert it to a byte frame

3. Send it

```
void CBMWDG::sendCAN()
{
    int speedTargetKMh = …, speedACCSetKMh = … // Some logic comes here …

    // ----------- 1. Initialize the CAN message -----------
    BMW::OP_CCTR_SF msg;
    msg.QU_SER_OP_CCTR.set(0x22); // SZL=0x21, MFL = 0x22
    msg.OP_CCTR_OPEL_6.set(BMW::OP_CCTR_OPEL_6_NEUTRAL);
    msg.OP_CCTR_OPEL_5.set(BMW::OP_CCTR_OPEL_5_NEUTRAL);
    msg.OP_CCTR_OPEL_4.set(BMW::OP_CCTR_OPEL_4_NEUTRAL);
    msg.OP_CCTR_OPEL_3.set(BMW::OP_CCTR_OPEL_3_NEUTRAL);
    msg.OP_CCTR_OPEL_2.set(BMW::OP_CCTR_OPEL_2_NEUTRAL);
msg.OP_CCTR_OPEL_1.set(BMW::OP_CCTR_OPEL_1_NEUTRAL);
msg.OP_CCTR_DUMMY.set(0);

if  (speedTargetKMh == -1                    )  /* do nothing */;
   else if (speedTargetKMh == BMW::ACC_OFF          )
          msg.OP_CCTR_OPEL_1.set(BMW::OP_CCTR_OPEL_1_ACTIVE);     // Off
else if (speedTargetKMh == BMW::ACC_STANDBY      )
          msg.OP_CCTR_OPEL_1.set(BMW::OP_CCTR_OPEL_1_ACTIVE);     // Off
   else if (speedTargetKMh == BMW::ACC_ACTIVATE     )
          msg.OP_CCTR_OPEL_1.set(BMW::OP_CCTR_OPEL_1_ACTIVE);     // Off
   else if (speedTargetKMh == BMW::ACC_RESUME       )
          msg.OP_CCTR_OPEL_2.set(BMW::OP_CCTR_OPEL_2_ACTIVE);     // Resume
   else if (speedTargetKMh >= speedACCSetKMh + 10  )
          msg.OP_CCTR_OPEL_3.set(BMW::OP_CCTR_OPEL_3_FORWARD2);  // +10
   else if (speedTargetKMh >  speedACCSetKMh    )
          msg.OP_CCTR_OPEL_3.set(BMW::OP_CCTR_OPEL_3_FORWARD);   // + 1
   else if (speedTargetKMh <= speedACCSetKMh - 10  )
          msg.OP_CCTR_OPEL_3.set(BMW::OP_CCTR_OPEL_3_BACKWARD2); // -10
   else if (speedTargetKMh <  speedACCSetKMh    )
          msg.OP_CCTR_OPEL_3.set(BMW::OP_CCTR_OPEL_3_BACKWARD);  // - 1

// Add meta information to the CAN msg here if needed …

// ----------- 2. Convert the CAN message to byte frame -----------
NTCAN::MsgTX msgTX;
msg.convert(msgTX);

// ----------- 3. Send the CAN message via the CAN Bus of the -----------
context.container->updateAllPlugIns(HINT_CAN_TX, &msgTX);

};
```

There are 3 main use cases, the Drive-Guard weather assistant plugin must fulfill:

- The vehicle drives faster than the maximum safety speed into an area with a special weather condition (A, Figure 9-35). In this case the vehicle slows down its velocity and restore it once it leaves (A).

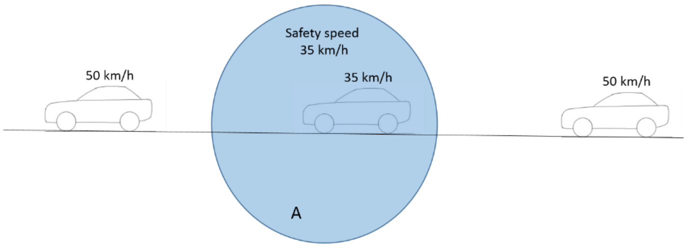


Figure 9-35. Event conditions and faster driving

- The vehicle drives slower than the maximum safety speed into an area with a special weather condition (A, Figure 9-36). In this case no actions should be performed.

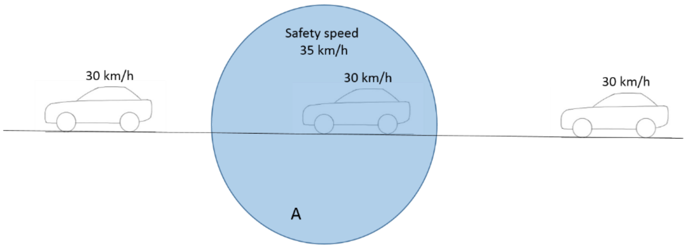


Figure 9-36. Event conditions an slower driving

- The vehicle drives faster/slower than the maximum safety speed into an area with a special weather condition (A, Figure 9-37). The driver manually adjust the ACC speed. When the vehicle goes out of A, its original speed should NOT be restored.

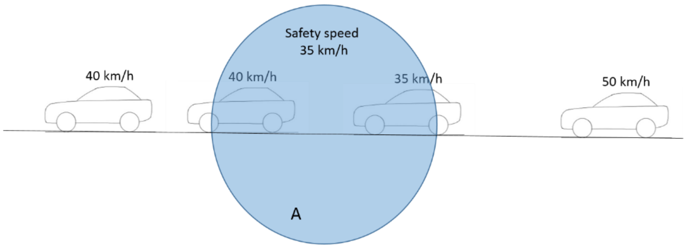


Figure 9-37. Event conditions and manual speed setting

In order to keep track of the information if the vehicle drives in an area with a weather condition or not, the vehicle needs to share its location for a specific period of time. Fortunately, there is a standard CAN massage holding the GPS position. In response, the server sends the current safety speed and the type of the weather condition, if any. According to the information received, it changes the state of the car and sends it via the CAN bus to all connected components to notify them of the change. It is their responsibility to react accordingly.

Since there are many sources which influence the speed of the car (weather condition or driver's manual interaction), the implementation of the plugin should use a state machine. The transitions between the individual states can be seen in Figure 9-38.

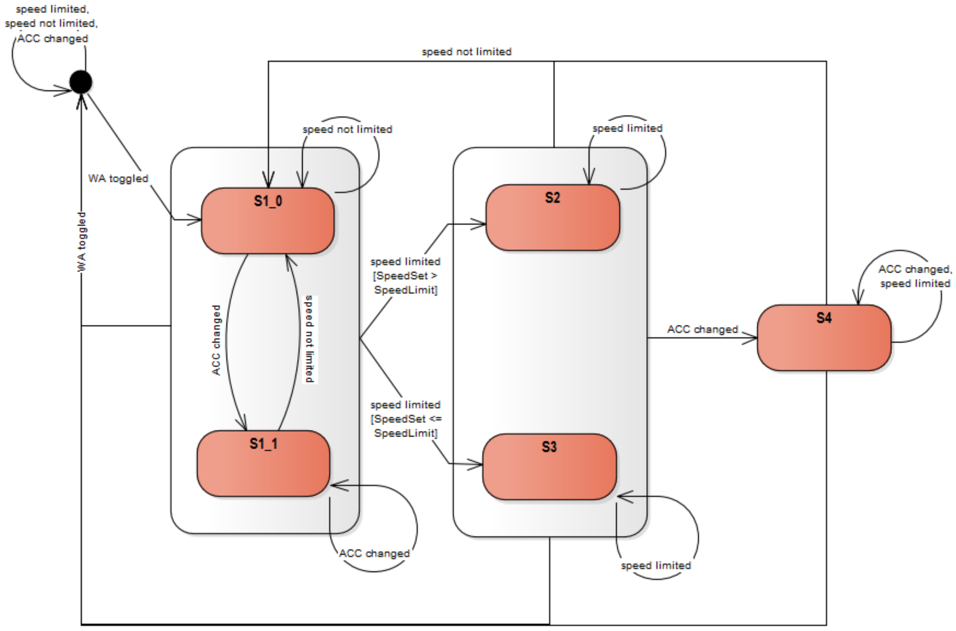


Figure 9-38. Weather assistant states for the vehicle plug-in

There are six possible states with the following semantics:

- Initial state (S0). The Drive-Guard ADAS is offline.
- S1_0. The Drive-Guard ADAS is on. There is no weather condition and therefore no safety speed.
- S1_1. The Drive-Guard ADAS is on. There is no weather condition and therefore no safety speed, but the driver has now changed the speed of ACC manually.
- S2. The Drive-Guard ADAS is on. The vehicle just entered an area with a special weather condition and obtained from the server its type and safety speed. The safety speed is set lower than the current speed. The vehicle should slow down. any further weather condition will have no direct impact on the vehicle’s state.
- S3. The Drive-Guard ADAS is on. The vehicle just entered an area with a special weather condition and obtained from the server its type and safety speed. The safety speed is NOT set lower than the current speed. The vehicle should not perform any additional actions. Every further weather condition has no direct impact on the vehicle’s state.

- S4. The Drive-Guard ADAS is on. The driver just changed the currently set ACC speed while driving within an area with a special weather condition. Any further weather condition received from the server will be ignored.

Implementation of the state machine

A central point in the state machine design is how to present the individual states and transactions. In order to keep the states consistent, an abstract class state has to be provided which should extend every concrete state of the machine (see Figure 9-39). It defines the core functionality like:

- virtual DG_STATE_TYPES getStateType() {} - realises the mapping between the actual state class and its symbolic representation.
- virtual State* execute(DG_STATE_TYPES, signed int, void*) {} - does a transition from the current state to the state, given as a first argument. The second and the third arguments are used to provide optional information to be used in the transition process e.g. what is the exact safety speed, how much is the change of ACC speed by manual interaction, etc. The second argument determines the type of information passed as void* in the third argument and is used in order to cast it correctly.

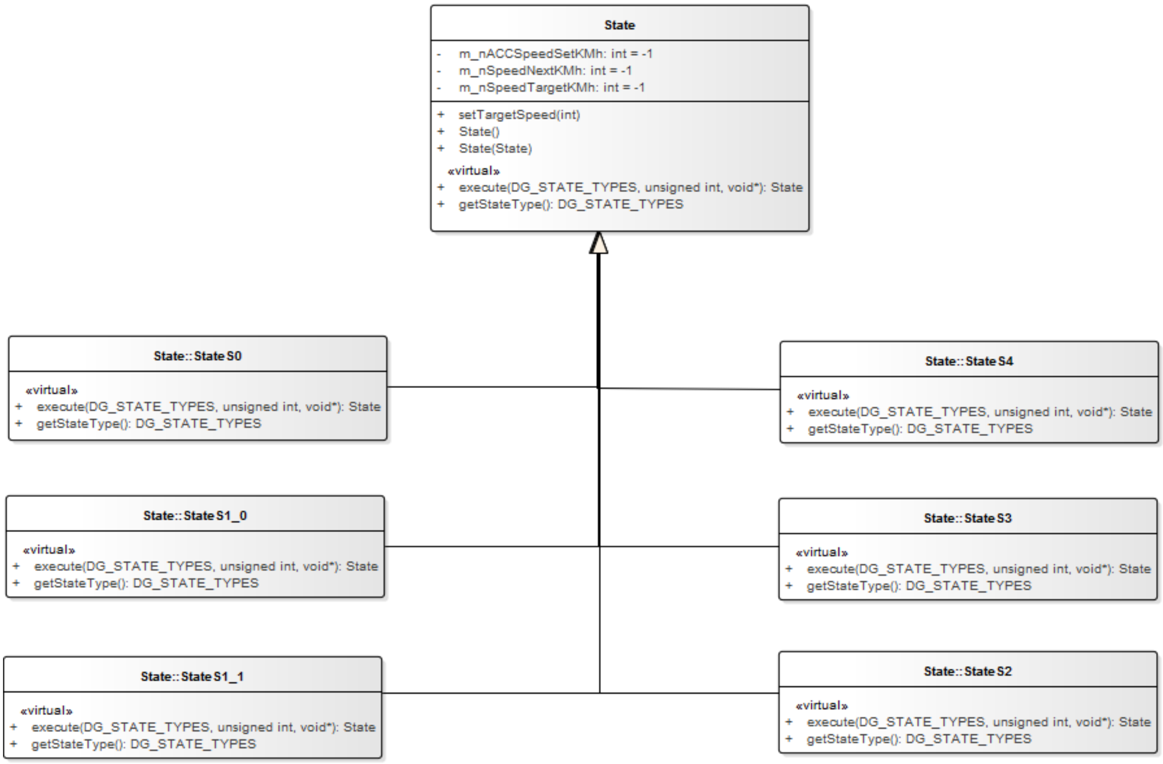

State
- m_nACCSpeedSetKMh: int = -1
- m_nSpeedNextKMh: int = -1
- m_nSpeedTargetKMh: int = -1
+ setTargetSpeed(int)
+ State()
+ State(State)
«virtual»
+ execute(DG_STATE_TYPES, unsigned int, void*): State
+ getStateType(): DG_STATE_TYPES
State:: State S0
«virtual»
+ execute(DG_STATE_TYPES, unsigned int, void*): State
+ getStateType(): DG_STATE_TYPES
State:: State S4
«virtual»
+ execute(DG_STATE_TYPES, unsigned int, void*): State
+ getStateType(): DG_STATE_TYPES
State:: State S1_0
«virtual»
+ execute(DG_STATE_TYPES, unsigned int, void*): State
+ getStateType(): DG_STATE_TYPES
State:: State S3
«virtual»
+ execute(DG_STATE_TYPES, unsigned int, void*): State
+ getStateType(): DG_STATE_TYPES
State:: State S1_1
«virtual»
+ execute(DG_STATE_TYPES, unsigned int, void*): State
+ getStateType(): DG_STATE_TYPES
State:: State S2
«virtual»
+ execute(DG_STATE_TYPES, unsigned int, void*): State
+ getStateType(): DG_STATE_TYPES

Figure 9-39. States machine for the vehicle plug-in

The state machine itself is implemented with the help of a two dimensional matrix. This matrix is called state transition matrix and holds in its columns the states and in the rows the user input causing the transition. The stored entries represent the state to which we should transit (see Table 4). The state transition matrix determines the first argument of the ***state***'s ***execute*** method.

Table 4. State Event Matrix for the State Machine Implementation

```
DG_STATE_TYPES TRANSITION_MATRIX[5][6] = {
                                  /*  S0   S1_0    S1_1   S2    S3    S4 */
       /* WA_Toggled        */    {  S1_0,  S0,     S0,   S0,   S0,   S0},
       /* ACC CHANCED       */    {   S0,  S1_1,  S1_1, S4_0,   S4,   S4},
       /* NO_LIMIT          */    {   S0,  S1_0,  S1_0, S1_0, S1_0, S1_0},
       /* LIMIT < ACC_SPEED */    {   S0,   S2,     S2,   S2,   S3,   S4},
       /* LIMIT >= ACC_SPEED*/    {   S0,   S3,     S3,   S2,   S3,   S4}

};
```

The LIMIT (Table 4) refers to the safety speed received from the server.

The class ***StateMachine*** given below shows complete implementation. It keeps information on the current state and has few helper functions. The most interesting of these are:

- The function ***conditionInd*** determines the row index by the information passed from the user (WA_TOGGLED, _CHANGED) or received from the Drive-Guard server (NO_SPEED_LIMIT, SPEED_LIMIT). Since there are different transitions for the safety speed greater or lower than the currently set ACC speed, this function can also take the safety speed value itself as an optional argument.

- As its name implies, the function ***transit*** is responsible for the actual transition of the state machine and internally changes the current state when needed.

```
class StateMachine {
public:
   StateMachine() { this->state = new StateS0(); }

   DG_STATE_TYPES getStateType() { return state->getStateType(); }

   int conditionInd(TRANSITION_TYPES cond, signed int lHint, void* pHint)
   {
      if  (lHint == BMW::HINT_DG_NONE) return (int) cond;
      else if (lHint == BMW::HINT_DG_ACC)  return (int) cond;
      else if (lHint == BMW::HINT_DG_SPEEDLIMIT)
      {
         int nSpeedLimitKMh = reinterpret_cast<int>(pHint);
         return (nSpeedLimitKMh < state->m_nSpeedTargetKMh ||
                      nSpeedLimitKMh < state->m_nACCSpeedSetKMh)
                     ? (int) cond : (int) cond + 1;
      }
      return 0;
   }
```

```
    void transit(TRANSITION_TYPES cond, signed int lHint, void* pHint)
    {
        int condInd = conditionInd(cond, lHint, pHint);
        int stateInd = (int)getStateType();

        DG_STATE_TYPES nextStateType = TRANSITION_MATRIX[condInd][stateInd];

        State* nextState = state->execute(nextStateType, lHint, pHint);
        if(nextState != state) delete state;
        state = nextState;
    }

    bool isWAActive () { return this->state->getStateType() != S0; }

private:
    State* state;
};
```

Visualisation

Each plugin receives a small UI from the MFC and can be used later on for test purposes or even to create some mock-ups. In this case (if the ACC controller and the weather assistant plugin are active), the BMWDG just visualises the current ACC speed, the target speed the ACC controller should change to, and the speed which should be set after the vehicle leaves the area with special weather condition. Figure 9-40 (from left to right, from top to bottom) describes in steps the scenario of activating the DG (Drive-Guard) weather assistant, driving into and out of a weather condition and the respective vehicle's speed adjustment process. The plugin visualises correctly by setting the concrete values of the three variables defined in Source Files\States\State.h. The value -1 for each variable indicates that the responsible element is inactive and therefore does not have to be visualised within ADASRP:

```
this->m_nACCSpeedSetKMh
this->m_nSpeedTargetKMh
this->m_nSpeedNextKMh
```

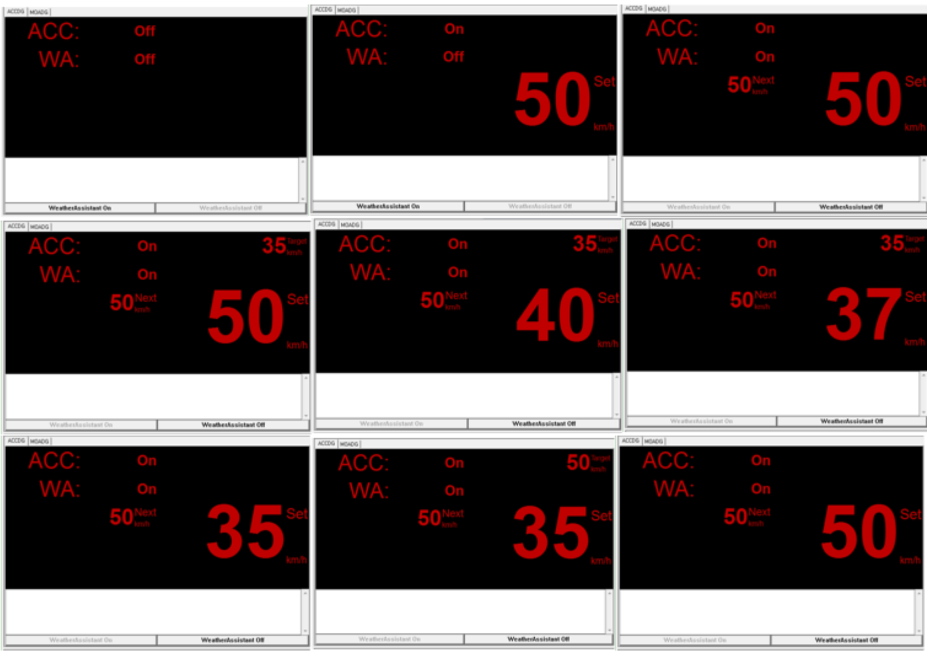


Figure 9-40. Activating Drive-Guard – visualization in steps

9.4.7 HMI on Head-Up Display on Floris Base

The Floris application is responsible for the rendering of the Head-Up Display (HuD) GUI as well as for the normal Human Machine Interaction display (HMI) in the research vehicle. Floris itself is a Flash application listening on http://localhost:8111 for different state notifications. Once such a notification arrives, the application visualises it in an appropriate manner. The protocol messages and the physical connection to Floris are realized in the ***CloudInformationLibrary*** project, which is added as a dependency to the main ADASRP plugin (BMWDG). The goal at the end is to obtain an ergonomic, attractive GUI as shown in Figure 9-41:

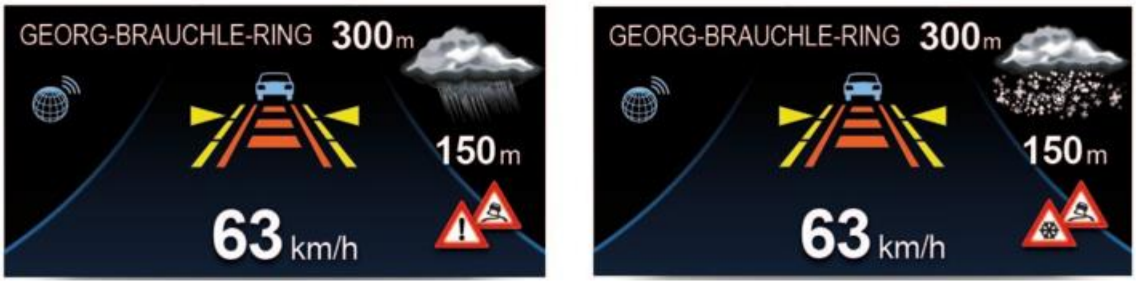


Figure 9-41. Head Up Display visualization in the experimental vehicle

9.4.8 Evaluation

To evaluate the impact of the Drive-Guard weather assistant, several aspects have to be estimated. Fuel consumption as well as velocity changes and driving comfort were evaluated using the data from several test experiments. The entire evaluation of the test and validation data obtained was carried out using MATLAB.

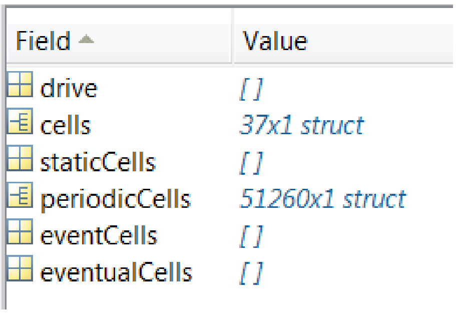
The test data was persisted in a form of a SQLite DB file. The processing is implemented in MATLAB with additional SQLite plugins.

To integrate SQL-Lite plugin:

- Download the .zip file from the link http://sourceforge.net/projects/mksqlite/?source=typ_redirect

- Extract the content of the ZIP-File to a directory inside the search path of MATLAB or use a new directory and insert it into the search path of MATLAB. To check if the plugin was integrated correctly, type "***help mksqlite***" in the command window. If the help page works, everything is correctly installed.

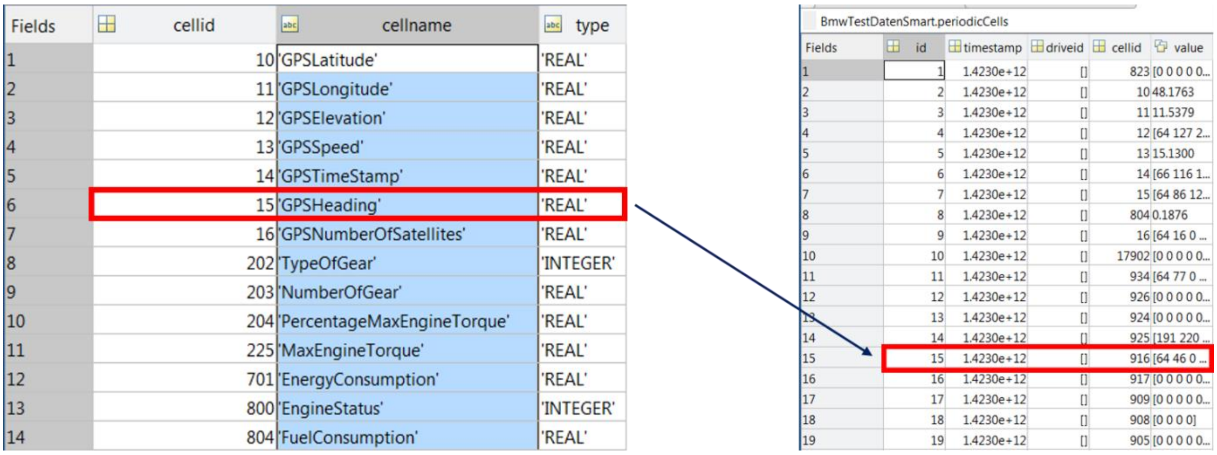
To start the program, run the main file. BmwTestDaten class performs the extraction of the necessary data using SQL commands. The data file is a 1x1 struct with 6 fields (see Figure 9-42).



Field	Value
drive	[ ]
cells	37x1 struct
staticCells	[ ]
periodicCells	51260x1 struct
eventCells	[ ]
eventualCells	[ ]

Figure 9-42. Structure of the extracted SQLite test data in MATLAB

The names of all corresponding signals are given in the field "***cells***" as well as their ID and type. In the field "***periodicCells***" the actual data is located (see Figure X). The data is encoded in bytes from right to left.



Fields	cellid	cellname	type
1	10	'GPSLatitude'	'REAL'
2	11	'GPSLongitude'	'REAL'
3	12	'GPSElevation'	'REAL'
4	13	'GPSSpeed'	'REAL'
5	14	'GPSTimeStamp'	'REAL'
6	15	'GPSHeading'	'REAL'
7	16	'GPSNumberOfSatellites'	'REAL'
8	202	'TypeOfGear'	'INTEGER'
9	203	'NumberOfGear'	'REAL'
10	204	'PercentageMaxEngineTorque'	'REAL'
11	225	'MaxEngineTorque'	'REAL'
12	701	'EnergyConsumption'	'REAL'
13	800	'EngineStatus'	'INTEGER'
14	804	'FuelConsumption'	'REAL'

BmwTestDatenSmart.periodicCells

Fields	id	timestamp	driveid	cellid	value
1	1	1.4230e+12	[]	823	[0 0 0 0 0...
2	2	1.4230e+12	[]	10	48.1763
3	3	1.4230e+12	[]	11	11.5379
4	4	1.4230e+12	[]	12	[64 127 2...
5	5	1.4230e+12	[]	13	15.1300
6	6	1.4230e+12	[]	14	[66 116 1...
7	7	1.4230e+12	[]	15	[64 86 12...
8	8	1.4230e+12	[]	804	0.1876
9	9	1.4230e+12	[]	16	[64 16 0 ...
10	10	1.4230e+12	[]	17902	[0 0 0 0 0...
11	11	1.4230e+12	[]	934	[64 77 0 ...
12	12	1.4230e+12	[]	926	[0 0 0 0 0...
13	13	1.4230e+12	[]	924	[0 0 0 0 0...
14	14	1.4230e+12	[]	925	[191 220 ...
15	15	1.4230e+12	[]	916	[64 46 0 ...
16	16	1.4230e+12	[]	917	[0 0 0 0 0...
17	17	1.4230e+12	[]	909	[0 0 0 0 0...
18	18	1.4230e+12	[]	908	[0 0 0 0]
19	19	1.4230e+12	[]	905	[0 0 0 0 0...

Figure 9-43. Association of signals and data

The same strategy for trip data analysis introduced in Chapter 9.3.5.1 was also applied to the obtained SQLite test data set.

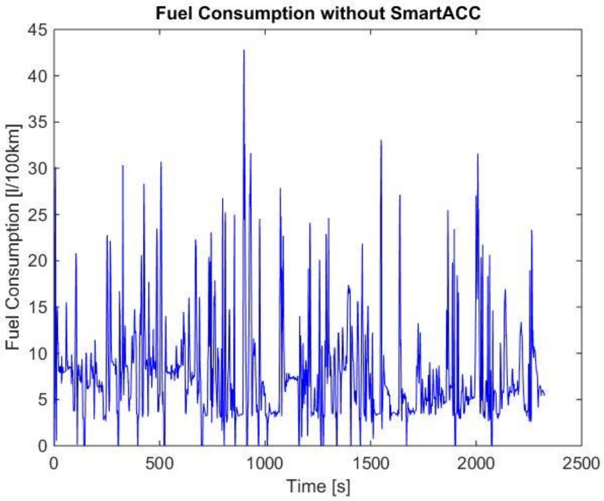


Figure 9-44. Fuel consumption with weather assistant application=OFF

About 10 trip traces were conducted without the “Weather Assistant Application” to obtain urban/suburban real life traffic data in terms of fuel consumption and CO_2 emissions as illustrated in Figure 9-44. The “Weather Assistant Application” was then activated to execute the test (again 10 trip traces on the same route and in the same traffic conditions) under research conditions, including the special control algorithms; the trace is shown in Figure 9-45.

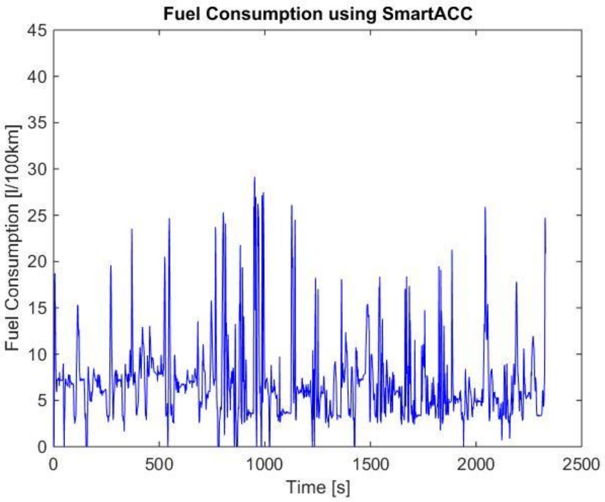


Figure 9-45. Fuel consumption with weather assistant application=ON

The final effect on fuel consumption - see Figure 9-46 - results in about 3% savings. This emphasises the potential of the reduction of fuel consumption and any related reduction of CO_2 emissions, as demonstrated by this experiment.

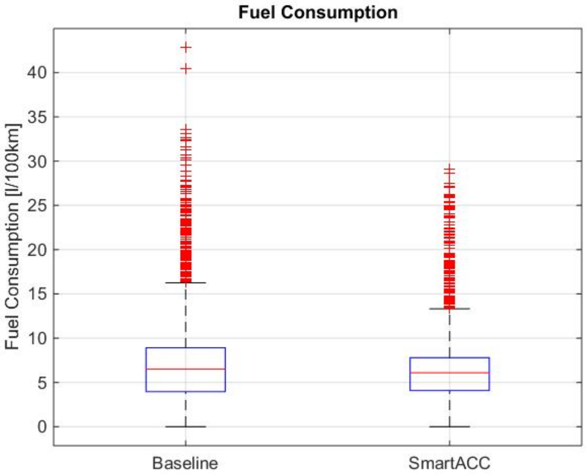


Figure 9-46. Fuel consumption with weather assistant application (BMW car)

10 Summary and outlook

10.1 Conclusion

In recent years a considerable number of scientific advances have been made in the generic domain of driver assistance and autonomous vehicles. This is recognisable in both published research results and actual vehicle industry developments. Advances are notable in the fields of vehicular complex sensing and control, wireless networks and their message addressing and routing, massive online analysis of big vehicular data, spatial-temporal data mining concepts, and comprehensive traffic control loop strategies.

At the time of writing this thesis, some first attempts have been made to gain floating car data (FCD) through readouts from mobile phones carried on board vehicles. The indirectly derived anonymous data provides qualification only of a few parameters such as near-accurate relative position, vehicle speed and directions from some initial traffic management services. The research described in this thesis illustrates that by introducing the concept of treating a ‘vehicle as a sensor within a vehicular network’, we can obtain greater granularity and data precision such that the quality of real time sensed data becomes much more reliable and meaningful when used for situational analysis and interpretation.

This thesis introduces a novel paradigm that aggregates and fuses numerous types of vehicular sensor data. This has been introduced to provide a subscriber-based business model for collaborative use and propagation of data within a vehicular traffic community. This data and interpretative activity utilizes the contemporary concepts of ‘cloud’ data storage and retrieval technologies. Within this model, the vehicle itself might subscribe vehicular messages from surrounding traffic or from the traffic cloud to become an actuator in a closed control loop thus, being a self-defined dynamical system. In combination with state of the art wireless sensors and mobile networks, experiments have shown that the communication performance will be sufficient for the expected network traffic requirements in terms of latency, data volume, message addressing & routing and the desired application types.

Sensing applications that have been demonstrated as being robust and exemplary in their capabilities and performance have been introduced and utilized. The context for this concept application is vehicle based weather detection by vision, LIDAR, and a combination of singular sensors. Degrading weather or hazardous situations, e.g. traffic congestion, or red traffic lights, were automatically recognized and broadcast by the vehicular probes in real time for processing of regional travel situation and hazard awareness. Location based traffic message addressing schemes have been put in place to demonstrate the validity of the geographical addressing of traffic participants through the back end cloud or among regional local traffic clusters.

A particular challenge addressed by this thesis has been the real time online analysis of a huge volume of location and time dependent traffic probe data. Inspired by some recent research around GIS database concepts developed for traffic analysis issues, a spatial-temporal analysis and data mining approach has been selected. Thus, the database approach assures that the real time input sensor stream is organized in terms of space and time. A subsequent data mining process provides regional clustered traffic data for broadcasting to the affected traffic areas.

Applications like ‘active driving safety’ and ‘green, eco-friendly driving’ have been demonstrated to illustrate the dynamic influence on traffic flow through a traffic management system. This approach uses the vehicular probes to gain a situational picture of a regional traffic situation. Furthermore, in case of the ‘eco-friendly driving’ application, the car becomes an actuator of the traffic management system. This is due to the exploitation of C-ACC (collaborative adaptive cruise control).

Testing the proposition of ‘Collaborative vehicles and smart traffic lights’ provides evidence that the traffic control cycle concept is sustainable. Simulations and real life experiments in urban environments have demonstrated that the control loop across vehicular sensors, traffic flow optimisation and actuator application on vehicles and traffic lights, does lead to expected or anticipated positive outcomes. Validations of experimental collected trace data have shown that the harmonised traffic flows lead to a remarkable reduction of emissions. The Drive-Guard ADAS ‘Weather Assistant Application’ is described as the fully integrated implementation of the concepts introduced within this thesis. Treating the vehicle as a sensor, it broadcasts its weather sensing to the vehicular cloud. This database is supplemented by external weather sources to provide the most comprehensive road travel situation analysis. A cloud based online analysis of this massive repository of data employs spatial-temporal methods for data mining and provides regional traffic messages. The vehicle as a subscriber receives the messages, and its control systems react/respond to the conditions prevailing by adapting the speed and running of the vehicle to the most-safe (optimal-for-conditions) cruising speed.

In conclusion, it is demonstrated that the principles and underlying concepts of the integrated methodology for building systems for safer driving and eco-clean vehicles provide plausible answers to the research questions initially posed.

10.2 Future work

10.2.1 Technology level

In the context of this thesis, vehicular sensor data has been acquired based on a huge number of uniformly distributed sensors. Within the next few years of deployment of such technology, sensor quantity and distribution will initially be non-uniform and unsynchronized. It is also predicted that a large number of sensor probes will generate clock distribution issues, and heterogeneous magnitudes will occur on sampling at different locations and time.

For such highly dynamic sensor data, a much more detailed trend analysis and short-lived event interpretation will be needed. More advanced spatial-temporal interpolation techniques combined with historical data and a probabilistic model to provide estimations at points with no current measurements (blind spots) have to be studied in more detail. For intelligent extrapolation and interpolation, sparse sensor data should be considered for reasonable calibration of a probabilistic model.

Another interesting topic will be the introduction of lifecycle management which includes confidence levels for actual values at points where currently no sensor data is available. Last but not least a cloud-based synchronization scheme needs to be designed for asynchronous sensor data.

For network load management purposes, some more detailed work should focus on sensor data throughput optimization. The transmissions of sensor value changes must be subject to data compression techniques that have to be adapted and applied to this purpose.

The cloud-vehicle control loops introduced in this thesis have been treated as real-time processes. In fact, due to technical restrictions such as network availability, bandwidth limitations or message queuing strategies, a real-time window cannot be assured at any time to any application request. A concept to grant well-defined, deterministic time slots to distributed applications has to be worked out in more detail.

While some sophisticated vision or LIDAR sensors are used in this thesis for traffic event and hazard detections, they are also essential for autonomous driving. They are capable of acquiring auxiliary information such as map data, traffic signs or road restrictions for real-time map updates in the cloud. The electronic horizon and its speed-profile described in previous chapters will benefit from always being up-to-date, supporting decisions to be made based on the most recent map data.

10.2.2 Application level

The applications for ‘improving driving safety’, ‘fuel economy’ or ‘Drive-Guard weather assistant’ researched here have been using recent historical and current data for analysis and interpretation of the traffic system within a particular region (spatial and time). The applications' impacts in the event of hazardous situations were warnings, influence on approaching speeds or simply rerouting to avoid congestion zones. New elements such as traffic flow estimations, short-term and mid-term weather forecasts or the introduction of short-term predictive traffic flow models will potentially allow proactive intervention in many types of ADAS vehicle applications.

Incorporation of other external probes (roadside units) or soft sensors on smartphones might extend situational cloud-based knowledge, which can be useful for the computation of even more detailed traffic information such as traffic jams, stop-and-go, accidents or other traffic-impairing factors.

Another topic could be multi-modal navigation which would automatically propose alternative transportation in case the ego-vehicular routing is not able to suggest reasonable routing in terms of time and distance due to traffic hazards or other disturbances.

Since in big cities, traffic management authorities are increasingly looking at air pollution aspects in relation to traffic planning, a geo-fencing concept might also be the subject of further research. This could be integrated into vehicular environmental perception and would provide a powerful mechanism to meet the goal of reducing emissions.

11 References

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, “Three Decades of Driver Assistance Systems: Review and Future Perspectives,” *Intelligent Transportation Systems Magazine, IEEE*, vol. 6, no. 4, pp. 6–22, 2014.

[2] *Unfallentwicklung auf deutschen Strassen 2012* (Deutsch). Available: https://www.destatis.de/DE/Publikationen/Thematisch/TransportVerkehr/Verkehrsunfaelle/PK_Unfallentwicklung_PDF.pdf?__blob=publicationFile.

[3] C. Roncoli, I. Papamichail, and M. Papageorgiou, Eds, *Model predictive control for multi-lane motorways in presence of VACS*, 2014.

[4] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, “Automated vehicle control developments in the PATH program,” *Vehicular Technology, IEEE Transactions on*, vol. 40, no. 1, pp. 114–130, 1991.

[5] S. E. Shladover, “Recent International Activity in Cooperative Vehicle–Highway Automation Systems,” 2012.

[6] M. E. Oonk, “Automation in Road Transport,” http://www.imobilitysupport.eu/library/imobility-forum/working-groups/active/automation/reports-3/2185-auto-wg-automation-roadmap-final-report-june-2013/file, 2013.

[7] *BMW Techniklexikon: BMW ConnectedDrive.* Available: http://www.bmw.com/com/de/insights/technology/technology_guide/articles/connecteddrive.html.

[8] Shengbo Li, Keqiang Li, R. Rajamani, and Jianqiang Wang, “Model Predictive Multi-Objective Vehicular Adaptive Cruise Control,” *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 3, pp. 556–566, 2011.

[9] C.-Y. Liang and H. Peng, “Optimal adaptive cruise control with guaranteed string stability,” *Vehicle System Dynamics*, vol. 32, no. 4-5, pp. 313–330, 1999.

[10] J. Ploeg, A. F. A. Serrarens, and G. J. Heijenk, “Connect & Drive: design and evaluation of cooperative adaptive cruise control for congestion reduction,” *Journal of Modern Transportation*, vol. 19, no. 3, pp. 207–213, 2011.

[11] H. Winner, S. Hakuli, and G. Wolf, *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort: Mit 550 Abbildungen und 45 Tabellen*: Springer-Verlag, 2009.

[12] M. Åkerholm, J. Carlson, J. Fredriksson, H. Hansson, J. Håkansson, A. Möller, P. Pettersson, and M. Tivoli, “The SAVE approach to component-based development of vehicular systems,” *Journal of Systems and Software*, vol. 80, no. 5, pp. 655–667, 2007.

[13] B. Filzek, “Abstandsverhalten auf Autobahnen - Fahrer und ACC im Vergleich,” *FORTSCHRITT-BERICHTE VDI. REIHE 12, VERKEHRSTECHNIK/FAHRZEUGTECHNIK*, no. 536, http://trid.trb.org/view.aspx?id=952908, 2003.

[14] M. Persson, F. Botling, E. Hesslow, and R. Johansson, Eds, *Stop and go controller for adaptive cruise control*: IEEE, 1999.

[15] D. Gross and C. Harris, “Fundamentals of queueing theory,” 1998.

[16] C.-F. Huang and Y.-C. Tseng, “The Coverage Problem in a Wireless Sensor Network,” *Mob. Netw. Appl*, vol. 10, no. 4, pp. 519–528, http://dl.acm.org/citation.cfm?id=1160162.1160175, 2005.

[17] “VANET: Vehicular Applications and Inter-Networking TechnologiesFront Matter,” in *VANET*, inthematics, Hannes Hartenstein diploma and BSE, Kenneth P. Laberteaux MSc, Eds.: John Wiley & Sons, Ltd, 2010, pp. i.

[18] V. Dahiya and A. Dureja, “Vehicular Ad Hoc network–Various Challenges and Applications,”

[19] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, “A comprehensive survey on vehicular Ad Hoc network,” *Journal of Network and Computer Applications*, vol. 37, pp. 380–392, http://www.sciencedirect.com/science/article/pii/S108480451300074X, 2014.

[20] M. Nekovee, Ed, *Sensor networks on the road: the promises and challenges of vehicular adhoc networks and vehicular grids*, 2005.

[21] Y. Harutoshi, D.K. Kiyasu, and H.H. Makino, Eds, *A Proving Test of the Advanced Cruise-Assist Highway System on a Highway curve section*, 2004.

[22] S. Yamada, “The strategy and deployment plan for VICS,” *Communications Magazine, IEEE*, vol. 34, no. 10, pp. 94–97, 1996.

[23] W. Franz, H. Hartenstein, and M. Mauve, *Inter-vehicle communications based on ad hoc networking principles: the FleetNet project*: Univ.-Verlag Karlsruhe, 2005.

[24] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite, and C. P. Thacker, “Autonet: A high-speed, self-configuring local area network using point-to-point links,” *Selected Areas in Communications, IEEE Journal on*, vol. 9, no. 8, pp. 1318–1335, 1991.

[25] *Sensor web*, 2014.

[26] L. Di, Ed, *Geospatial sensor web and self-adaptive Earth predictive systems (SEPS)*, 2007.

[27] Akyildiz, I. F, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, http://www.sciencedirect.com/science/article/pii/S1389128601003024, 2002.

[28] M. Bottero, B. Dalla Chiara, and Deflorio, F. P, “Wireless sensor networks for traffic monitoring in a logistic centre,” *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 99–124, http://www.sciencedirect.com/science/article/pii/S0968090X12000903, 2013.

[29] *SCIENCE HOBBYIST: Traffic Waves, physics for bored commuters.* Available: http://trafficwaves.org/ (2015, Mar. 20).

[30] Wikipedia, *Traffic wave - Wikipedia, the free encyclopedia.* Available: http://en.wikipedia.org/w/index.php?oldid=642137215 (2015, Mar. 20).

[31] *TRAFFIC WAVE EXPERIMENTS.* Available: http://trafficwaves.org/trafexp.html (2015, Mar. 20).

[32] *What Are Traffic Waves and Why Do They Happen So Much?* Available: http://blogs.kqed.org/lowdown/2013/11/12/traffic-waves (2015, Mar. 20).

[33] Bilgin, B. E. and Gungor, V. C, “Performance Comparison of IEEE 802.11p and IEEE 802.11b for Vehicle-to-Vehicle Communications in Highway, Rural, and Urban Areas,” (en), *International Journal of Vehicular Technology*, vol. 2013, 2013.

[34] W. Specks, K. Matheus, R. Morich, I. Paulus, C. Menig, A. Lübke, B. Rech, and V. Audi, Eds, *Car-to-car communication–market introduction and success factors*, 2005.

[35] C. Dannheim, C. Icking, M. Mäder, and P. Sallis, Eds, *Weather detection in Vehicles by means of Camera and LIDAR Systems*: IEEE, 2014.

[36] D. Aubert, V. Boucher, R. Brémond, P. Charbonnier, A. Cord, E. Dumont, P. Foucher, F. Fournela, F. Greffier, and D. Gruyer, Eds, *Digital imaging for assessing and improving highway visibility*, 2014.

[37] M. Pavlic, H. Belzner, G. Rigoll, and S. Ilic, Eds, *Image based fog detection in vehicles*: IEEE, 2012.

[38] M. Pavlić, *Kamerabasierte Nebeldetektion und Sichtweitenschätzung im Fahrzeug*: Verlag Dr. Hut, 2014.

[39] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *Neural Networks, IEEE Transactions on*, vol. 13, no. 3, pp. 697–710, 2002.

[40] A. N. Amritphale, "A Digital Image Processing Method for Detecting Pollution in the Atmosphere from Camera Video," *IP version*, 2013.

[41] *How LiDAR works - LiDAR-UK.* Available: http://www.lidar-uk.com/how-lidar-works/.

[42] Ibeo Automotive Systems GmbH, *Operating Manual Ibeo LUX 2010® Laserscanner*, 2010.

[43] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, "A Perception-Driven Autonomous Urban Vehicle," in *Springer Tracts in Advanced Robotics, The DARPA Urban Challenge*, B. Siciliano, O. Khatib, F. Groen, M. Buehler, K. Iagnemma, and S. Singh, Eds, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 163–230.

[44] A. B. Utkin, A. V. Lavrov, L. Costa, F. Simoes, and R. Vilar, "Detection of small forest fires by lidar," *Applied Physics B*, vol. 74, no. 1, pp. 77–83, 2002.

[45] A.B. Utkin, A. Fernandes, F. Simoes, R. Vilar, and A. Lavrov, Eds, *Forest-fire detection by means of lidar*, 2002.

[46] Vladimir Fischer, "Das Fahrzeug als mobiler Sensor," Diplom Arbeit, Kooperative Systeme, FernUniversität in Hagen, Hagen, 2014.

[47] L. Wald, *Data fusion: definitions and architectures: fusion of images of different spatial resolutions*: Presses des MINES, 2002.

[48] A. N. Steinberg, "Data fusion system engineering," *IEEE Aerospace and Electronic Systems Magazine*, vol. 16, no. 6, pp. 7–14, 2001.

[49] A. N. Steinberg and C. L. Bowman, “Rethinking the JDL data fusion levels,” *NSSDF JHAPL*, vol. 38, p. 39, 2004.

[50] M. Darms, “Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfeldsensoren fiir Fahrerassistenzsysteme,” *VDI, Diisseldorf*, 2007.

[51] M. Mäder, “Collaborative Positioning,” Dissertation, Kooperative Systeme, FernUniversität in Hagen, Hagen, 2015 in Vorbereitung.

[52] C. Dannheim, M. Mäder, and C. Icking, Eds, *Geo-Based IPv6 Multicast in LTE for Dynamic Addressing of Road Users*: IEEE, 2013.

[53] rdroms. ietf@gmail. com, Ralph Droms, *IPv6 Multicast Address Scopes.* Available: http://tools.ietf.org/html/rfc7346.

[54] B. Kloiber, T. Strang, H. Spijker, and G. Heijenk, Eds, *Improving information dissemination in sparse vehicular networks by adding satellite communication*: IEEE, 2012.

[55] D.C. Dimitrova, M. Brogle, T. Braun, G.J. Heijenk, N. Meratnia, K. Trichias, Berg, J. L. van den, Jongh, J. de, and R. Litjens, Eds, *Modeling and Evaluation of LTE in Intelligent Transportation Systems*: University of Bern, 2012.

[56] *eCall: Time saved = lives saved: Digital Agenda for Europe.* Available: ec.europa.eu//digital-agenda/ecall-time-saved-lives-saved.

[57] D. Burgstrahler, S. Zöller, M. Möbus, T. Walter, T. Rückelt, and R. Steinmetz, “Database-Aided Information Approach for Driver Assistance Systems,” in *Database-Aided Information Approach for Driver Assistance Systems*, Dortmund, 2015.

[58] Liu Jiang, Cai Bai-Gen, Wang Yun-Peng, and Wang Jian, Eds, *Location-based Cooperative Vehicle Collision Avoidance for Unsignalized Intersections: A Multi-sensor Integration Approach*. Connected Vehicles and Expo (ICCVE), 2012 International Conference on, 2012.

[59] R. Hinden and S. Deering, “RFC 4291,” *IP version*, vol. 6, pp. 13–15.

[60] *TEAM - IP project: TEAM.* Available: http://www.collaborative-team.eu/overview/objectives.

[61] Tanaev, V. S, Gordon, V. S, and Shafransky, Y. M, “NP-Hard Problems,” in *Mathematics and Its Applications, Scheduling Theory. Single-Stage Systems*: Springer Netherlands, 1994, pp. 253–311.

[62] Lindberg, P. O. and G. Razmara, “The Prize Collecting Connected Subgraph Problem - A New NP-Hard Problem arising in Snow Removal Routing,” in *Operations Research Proceedings, Operations Research Proceedings 2004*, Fleuren, Professor Dr Hein, Hertog, Professor Dr Dick den, and Kort, Professor Dr Peter, Eds.: Springer Berlin Heidelberg, 2005, pp. 360–367.

[63] S. Drobot, Mahoney III, William P, E. Schuler, G. Wiener, M. Chapman, P. A. Pisano, P. Kennedy, B. B. McKeever, and A. D. Stern, “IntelliDrivesm Road Weather Research & Development-the Vehicle Data Translator,” *National Center for Atmospheric Research. http://www. ops. fhwa. dot. gov/weather/resources/publications/itsapaper9005/itsapaper900*, vol. 5, 2009.

[64] A. Jaeger and S.A. Huss, Eds, *The weather hazard warning in simTD: A design for road weather related warnings in a large scale Car-to-X field operational test*, 2011.

[65] J. Loewenau, P.-S.D. Huang, G. Schmitz, H. Wigermo, and S. Durekovic, Eds, *eCoMove–EfficientDynamics Approach to Sustainable CO2 Reduction*, 2010.

[66] J. Löwenau and C. Dannheim*, Advanced Driver Assistance Systems (ADAS) based on Vehicular Networks and the Traffic Cloud: ADAS and the Speed-Profile*. München, 2013.

[67] *HERE – navigation, mapping and live traffic app: ADAS.* Available: https://company.here.com/automotive/automotive-services-traffic/overview/ (2015, Mar. 22).

[68] B. Thomas, J. Löwenau, S. Durekovic, and H.-U. Otto, Eds, *The ActMap–FeedMap framework a basis for in-vehicle ADAS application improvement*, 2008.

[69] C. Ress, D. Balzer, A. Bracht, S. Durekovic, and J. Löwenau, Eds, *Adasis protocol for advanced in-vehicle applications*, 2008.

[70] S. Bechtolsheim, L. Dunn, A. Hecht, M. Schmitt, J. Feigen, and M. Roser, "Method and system for providing an electronic horizon in an advanced driver assistance system architecture", May 11, 2004

[71] Fanping Bu, Han-Shue Tan, and Jihua Huang, Eds, *Design and field testing of a Cooperative Adaptive Cruise Control system*. American Control Conference (ACC), 2010, 2010.

[72] S. Andresson and J. Aronsson, “Road Shape Modelling from Digital Map Data and Implementation of a Map Supported Cruise Control,” Chalmers, 2009.

[73] A. Gühnemann, R.-P. Schäfer, K.-U. Thiessenhusen, and P. Wagner, “Monitoring traffic and emissions by floating car data,” *Institute of Transport Studies Working Paper*, no. ITS-WP-04-07, 2004.

[74] D. Caveney and W.B. Dunbar, Eds, *Cooperative driving: Beyond V2V as an ADAS sensor*, 2012.

[75] S. Geisler, C. Quix, S. Schiffer, and M. Jarke, “An evaluation framework for traffic information systems based on data streams,” *Transportation Research Part C: Emerging Technologies*, vol. 23, pp. 29–55, http://www.sciencedirect.com/science/article/pii/S0968090X11001136, 2012.

[76] C. E. White, D. Bernstein, and A. L. Kornhauser, “Some map matching algorithms for personal navigation assistants,” *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1–6, pp. 91–108, http://www.sciencedirect.com/science/article/pii/S0968090X00000267, 2000.

[77] *Publikation - Transport & Verkehr - Verkehrsunfälle - Fachserie 8 Reihe 7 - 2013 - Statistisches Bundesamt (Destatis).* Available: https://www.destatis.de/DE/Publikationen/Thematisch/TransportVerkehr/Verkehrsunfaelle/VerkehrsunfaelleJ2080700137004.pdf.

[78] P. Szczurek, B. Xu, O. Wolfson, and J. Lin, Eds, *A platform for the development and evaluation of passive safety applications*, 2012.

[79] W. Huber, M. Lädke, and R. Ogger, Eds, *Extended floating-car data for the acquisition of traffic information*, 1999.

[80] S. Breitenberger, B. Gruber, M. Neuherz, and R. Kates, “Traffic information potential and necessary penetration rates,” *Traffic Engineering and Control*, vol. 45, pp. 396–401, http://www.bmwgroup.com/e/0_0_www_bmwgroup_com/forschung_entwicklung/mobilitaet_verkehr/verkehrsforschung/ExtendedFloatingCarData_Paper.pdf, 2004.

[81] J. Wahle, “Verkehrsmanagement aus Sicht der privaten Dienstleister,” in *Verkehrsmanagementzentralen in Kommunen*, M. Sandrock and G. Riegelhuth, Eds.: Springer Fachmedien Wiesbaden, 2014, pp. 121–156.

[82] T. Otto, *Kooperative Verkehrsbeeinflussung und Verkehrssteuerung an signalisierten Knotenpunkten*: kassel university press GmbH, 2011.

[83] M. O. Cherif, S.-M. Senouci, and B. Ducourthial, “Efficient data dissemination in cooperative vehicular networks,” (en), *Wirel. Commun. Mob. Comput*, vol. 13, no. 12, pp. 1150–1160, 2013.

[84] M. Picone, M. Amoretti, and F. Zanichelli, Eds, *A decentralized smartphone based Traffic Information System*, 2012.

[85] S. Fürst, J. Mössinger, S. Bunzel, T. Weber, F. Kirschke-Biller, P. Heitkämper, G. Kinkelin, K. Nishikawa, and K. Lange, Eds, *AUTOSAR–A Worldwide Standard is on the Road*, 2009.

[86] C. Dannheim, M. Mäder, C. Icking, J. Loewenau, and K. Massow, Eds, *A Novel Approach for the Enhancement of Cooperative ACC by Deriving Real Time Weather Information*: IEEE, 2013.

[87] A. S. Vicente, “Laying the foundations for greener transport. TERM 2011: Transport indicators tracking progress towards environmental targets in Europe, EEA Report 7/2011,” *European Environment Agency, Copenhagen, Denmark*, 2011.

[88] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, Van de Molengraft, MJG, and M. Steinbuch, “String-stable CACC design and experimental validation: A frequency-domain approach,” *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 9, pp. 4268–4279, 2010.

[89] C. Weiß, “V2X communication in Europe–From research projects towards standardization and field testing of vehicle communication technology,” *Computer Networks*, vol. 55, no. 14, pp. 3103–3119, 2011.

[90] B. Arief and A. von Arnim, Eds, *TRACKSS approach to improving road safety through sensors collaboration on vehicle and in infrastructure*: IEEE, 2008.

[91] Zarr, F. Richard, “Automotive Communications Demand A Robust Infrastructure: Electronic design,” http://electronicdesign.com/communications/automotive-communications-demand-robust-infrastructure.

[92] X. Yan, Y. Luo, and X. Zheng, “Weather Recognition Based on Images Captured by Vision System in Vehicle,” in *Lecture Notes in Computer Science, Advances in Neural Networks – ISNN 2009*, W. Yu, H. He, and N. Zhang, Eds.: Springer Berlin Heidelberg, 2009, pp. 390–398.

[93] B. S. Goldsmith, “Socio-Economic Research to Build a Weather Ready Nation,” *WSW Special Issue*, http://www.sip.ucar.edu/news/pdf/WSW_January_2012.pdf, 2012.

[94] *Intelligent Transportation Systems "Weather Applications - Weather Detection"*. Available: http://fresno.ts.odu.edu/newitsd/ITS_Serv_Tech/weather_app/weather_applications_summary.html.

[95] U. Mellinghoff, T. Breitling, R. Schöneburg, and H. Metzler, Eds, *The Mercedes-benz experimental safety vehicle 2009*, 2009.

[96] S. El-Tawab, M. Abuelela, and Y. Gongjun, Eds, *Real-time weather notification system using intelligent vehicles and smart sensors*: IEEE, 2009.

[97] A. R. Foo and O. F. Bin, "LINUX Based Application Layer for Vehicle Based Road/Environment Condition Warning System using Vehicular Ad hoc Networks (VANETs)," 2012.

[98] J. M. Lutin, A. L. Kornhauser, and E. Lerner-Lam, "The revolutionary development of self-driving vehicles and implications for the transportation engineering profession," *ITE Journal*, vol. 83, no. 7, 2013.

[99] K. R. Petty, M. Iii, and W. P, "Weather Applications and Products Enabled Through Vehicle Infrastructure Integration (VII) Feasibility and Concept Development Study," http://trid.trb.org/view.aspx?id=809115, 2007.

[100] A. Festag, "Cooperative intelligent transport systems standards in Europe," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[101] A. Festag, L. Le, and M. Goleva, Eds, *Field operational tests for cooperative systems: a tussle between research, standardization and deployment*: ACM, 2011.

[102] J. Ryde and N. Hillier, "Performance of laser and radar ranging devices in adverse environmental conditions," *Journal of Field Robotics*, vol. 26, no. 9, pp. 712–727, 2009.

[103] C. Hammerschmidt, "Low-grip conditions early warning system uses sensor fusion," Analog EEtimes, 2010.

[104] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara, Eds, *Rainy weather recognition from in-vehicle camera images for driver assistance*, 2005.

[105] M. Jokela, M. Kutila, J. Laitinen, F. Ahlers, N. Hautière, and T. Schendzielorz, "Optical Road Monitoring of the Future Smart Roads – Preliminary Results," *World Academy of Science, Engineering and Technology*, no. 10, http://mediatum.ub.tum.de/node?id=1141222&change_language=en, 2007.

[106] A. Guan, S. H. Bayless, and R. Neelakantan, “Trends in Computer Vision: An overview of vision-based data acquisition and processing technology and its potential for the transportation sector,” http://trid.trb.org/view.aspx?id=1259535, 2011.

[107] N. Hautiere and A. Boubezoul, Eds, *Combination of Roadside and In-vehicle Sensors for Extensive Visibility Range Monitoring*, 2009.

[108] G. FREMONT and T. BOYER, *A co-operative vehicle / infrastructure system to improve road transport safety and provide accurate in - vehicle information: ASECAP 2007 Annual congress.* Available: http://www.asecap.com/english/documents/SAFESPOTCVIS.pdf.

[109] European Commission, *Intelligent transport systems.* Available: http://ec.europa.eu/transport/themes/its/index_en.htm.

[110] M. Bohm and A. Frotscher, Eds, *Data-Flow and Processing for Mobile In-Vehicle Weather Information Services COOPERS Service Chain for Co-operative Traffic Management*, 2009.

[111] F. Losilla, A.-J. Garcia-Sanchez, F. Garcia-Sanchez, J. Garcia-Haro, and Z. J. Haas, “A comprehensive approach to WSN-based ITS applications: A survey,” *Sensors*, vol. 11, no. 11, pp. 10220–10265, 2011.

[112] A. Spence, *Safespot SP2 – INFRASENS – SAFESPOT Infrastructure Platform.Final Report: Implementation and prototypes for infrastructure-based components.* Available: http://www.safespot-eu.org/deliverables.html.

[113] *ecoDriver - Supporting the driver in conserving energy and reducing emissions.* Available: http://www.ecodriver-project.eu/.

[114] *EcoMove - Cooperative Mobility Systems and Services for Energy Efficiency - Vision.* Available: http://www.ecomove-project.eu/about-ecomove/vision/.

[115] R. Brouwer, *Rino Brouwer explains how HMIs may improve eco-driving.* Available: http://www.ecodriver-project.eu/news/news/rino-brouwer-explains-how-hmis-may-contribute-to-eco-driving-video/.

[116] *Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system (WHITE PAPER) EUR-Lex - 52011DC0144 - EN.* Available: http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52011DC0144&from=EN.

[117] V. Miz and V. Hahanov, “Smart traffic light in terms of the cognitive road traffic management system (CTMS) based on the Internet of Things,” *Design & Test Symposium (EWDTS), 2014 East-West,* pp. 1–5, 2014.

[118] I.R. Wilmink, G.A. Klunder, and B. van Arem, Eds, *Traffic flow effects of Integrated full-Range Speed Assistance (IRSA).* Intelligent Vehicles Symposium, 2007 IEEE, 2007.

[119] X. Qian, J. Gregoire, De La Fortelle, Arnaud, and F. Moutarde, “Decentralized model predictive control for smooth coordination of automated vehicles at intersection,”

[120] L. Makarem and D. Gillet, Eds, *Fluent coordination of autonomous vehicles at intersections*: IEEE, 2012.

[121] *PHABMACS: Physics Aware Behavior Modelling Advanced Car Simulator: Fraunhofer FOKUS.* Available: https://www.fokus.fraunhofer.de/870187f17cad3c03/simulation-testing.

[122] *SQLite Home Page.* Available: http://sqlite.org/.

[123] A. Corti, V. Manzoni, S. M. Savaresi, M. D. Santucci, and O. Di Tanna, “A Centralized Real-Time Driver Assistance System for Road Safety Based on Smartphone,” in *Advanced Microsystems for Automotive Applications 2012*: Springer, 2012, pp. 221–230.

Authors Biography

Personal Data

Clemens Dannheim

Birthday: 12.08.1960, Lünen, Germany

Education

- 8/1981 – 8/1984

  Dipl.-Informatiker, Fachhochschule Dortmund, Germany

- 10/2007 – 9/2010

  M.Comp.Sc., FernUniversität Hagen, Germany

Professional Experience

- 1984 - 1986

  Siemens AG, Rohde & Schwarz München, Germany

- 1987 - 1997

  Softlab GmbH München, Germany

- 1998 - today

  Objective Software GmbH München, Germany

Academic Curriculum

- Q4/2014

  Guest research fellow, Auckland University of Technology, New Zealand

- 2012 - today

  Research fellow, EU funded automotive research projects

- 2010 - today

  Research engineer, BMW Research & Technology Munich, Germany

- 2011 - today

  Lectures at University of Applied Sciences Augsburg, Germany

Awards

- 5th German IPv6 Summit, 3rd price sponsored by “Hasso Plattner Institute“: Geo-based IPv6 routing for C2X communication within LTE,” 2012 (https://www.ipv6council.de/contest2012/vertikal_menu/winners/).

Publications

[1] J. Sieck, C. Dannheim, M. Mäder, and C. Icking, Eds, *Geobasiertes IPv6-Multicast in LTE zur dynamischen Adressierung von Verkehrsteilnehmern*: Verlag Werner Hülsbusch, 2012.

[2] J. Sieck, C. Dannheim, M. Mäder, and C. Icking, Eds, *Aktive Fahrsicherheit auf Basis leistungsfähiger Mobilfunktechnologien*: Verlag Werner Hülsbusch, 2012.

[3] P. Sallis, and C. Dannheim et al, *Instrumentation and algorithms for the identification of ill-formed objects.*

[4] J. Löwenau and C. Dannheim, *Advanced Driver Assistance Systems (ADAS) based on Vehicular Networks and the Traffic Cloud: ADAS and the Speed-Profile*. München, 2013.

[5] C. Dannheim, M. Mäder, and C. Icking, Eds, *Collaborative Traffic Technologies and their Effect on Driving Safety*: Springer, 2014.

[6] C. Dannheim, M. Mäder, C. Icking, J. Loewenau, and K. Massow, Eds, *TEAM: CO2 Reduction through Online Weather Assistant for Cooperative ACC Driving*: IEEE, 2013.

[7] C. Dannheim, M. Mäder, and C. Icking, Eds, *Geo-Based IPv6 Multicast in LTE for Dynamic Addressing of Road Users*: IEEE, 2013.

[8] C. Dannheim, M. Mäder, and C. Icking, Eds, *Cooperation and Collaboration in ITS Systems through Geo-Based Addressing of Road Users*: IEEE, 2013.

[9] C. Dannheim, M. Mäder, and C. Icking, Eds, *Active Driving Safety Based On Efficient Mobile Technologies*: IEEE, 2013.

[10] C. Dannheim, C. Icking, M. Mäder, and P. Sallis, Eds, *Weather detection in Vehicles by means of Camera and LIDAR Systems*: IEEE, 2014.

[11] C. Dannheim, C. Icking, and M. Mäder, “Geobasiertes IPv6 Routing zur C2X Kommunikation in LTE,” 2012.

[12] C. Dannheim, M. Mäder, C. Icking, J. Loewenau, and K. Massow, Eds, A Novel Approach for the Enhancement of Cooperative ACC by Deriving Real Time Weather Information: IEEE, 2013.

Acknowledgements

At this point I would like to take the opportunity to thank Prof. Dr. habil. Christian Icking and Dr. Lihong Ma of the Department of Cooperative Systems at the University in Hagen, Germany, for their encouragement and motivation to take up doctoral studies in 2010 and the excellent accompanying support they have provided. As a result, many new personal, professional and scientific topics have arisen during the past few years. I also thank the company BMW Research and Technology GmbH represented by my industry advisor Dr. Klaus Gresser and supervisor Dr. Jan Löwenau: through this organisation I was able to find an innovative research topic with a valuable public benefit. In recent years, I received particular support from Dr. Jan Löwenau in terms of brainstorming, research and validation at the research vehicle, also in relation to the EU research project TEAM. I also extend my personal thanks to Prof. Philip Sallis since he enabled me to undertake a three-month research period at his institute in 2014 and acted as a second supervisor. I would also mention those responsible for EU research projects since they enabled me to explore interesting new fields of knowledge, through which they are now able to initiate projects and thus develop pioneering new concepts of mobility in Europe. I would especially like to mention my fellow student Dipl.-Inform. Markus Mäder since we acted as a team regarding our respective research topic, providing each other with a sparring partner. The core team of my company Objective Software GmbH, consisting of Mrs. Karin von Daak and Mrs. Vyara Radeva provided countless support through motivation and schedule organisation.

Special thanks go to my family. Without the organisational skills of my wife to create the freedom for me to undertake my research and work, this thesis would not have been possible.

Höhenkirchen, April 2015

Clemens Dannheim

Danksagung

An dieser Stelle möchte mich besonders bei Herrn Prof. Dr. habil. Christian Icking und Frau Dr. Lihong Ma des Lehrgebietes Kooperative Systeme an der Fernuniversität in Hagen für den Impuls und die Motivation zur Aufnahme des Doktorandenstudiums im Jahre 2010 und der exzellenten begleitenden Betreuung bedanken. Dadurch haben sich für mich in den zurückliegenden Jahren zahlreiche neue persönliche, berufliche und wissenschaftliche Themenstellungen ergeben. Der Firma BMW Forschung & Technik GmbH vertreten durch meine Industriebetreuer Hr. Dr. Klaus Gresser und Hr. Dr. Jan Löwenau ist zu verdanken, dass ich durch sie eine innovative Aufgabenstellung mit einem wertvollen Nutzen für die Allgemeinheit finden konnte. Besonders hat mich Hr. Dr. Löwenau in den letzten Jahren, auch im Kontext zum EU-Forschungsprojekt TEAM, in der Ideenfindung, der Forschung und der Erprobung am Versuchsträger Automobil sehr unterstützt. Herrn Prof. Philip Sallis von der Auckland University of Technology (AUT) in Neuseeland gilt auch mein persönlicher Dank, da er mir im Jahr 2014 einen drei monatigen Forschungsaufenthalt an seinem ‚Geoinformatics Research Center‘ ermöglichte und sich als Zweitgutachter zur Verfügung gestellt hat. Die Verantwortlichen für EU-Forschungsvorhaben seien ebenfalls genannt, weil ich durch die von ihnen ausgelösten Projekte interessante Wissensgebiete neu erschließen konnte und damit zukunftsweisende neue Konzepte der Mobilität in Europa entstehen. Meinen Kommilitonen Herrn Dipl.-Inform. Markus Mäder möchte ich besonders erwähnen, da wir uns im Team bezüglich unserer jeweiligen Forschungsthemen zu jeder Zeit wertvolle Gesprächspartner waren. Das Kernteam meiner Firma Objective Software GmbH, bestehend aus Frau Karin von Daak und Frau Vyara Radeva hat unzählige Male durch Motivation und Freihalten von Terminen wertvolle Unterstützung geleistet.

Besonderen Dank möchte ich meiner Familie aussprechen. Ohne die organisatorische Meisterleistung meiner Frau mir die Freiräume des Forschens und Arbeitens zu schaffen, wäre die Arbeit nicht möglich gewesen.

Höhenkirchen, April 2015

Clemens Dannheim