An Automated Electric Vehicle Prototype Showing New Trends in Automotive Architectures

Martin Buechel*, Jelena Frtunikj*, Klaus Becker*, Stephan Sommer*, Christian Buckl*
Michael Armbruster†, Cornel Klein†, Andre Marek†, Andreas Zirkler† and Alois Knoll‡
*fortiss GmbH, An-Institut Technische Universität München, Munich, Germany, Email: name.surname@fortiss.org
†Siemens AG, Corporate Technology, Munich, Germany, Email: name.surname@siemens.com
‡Technische Universität München, Robotic and Embedded Systems, Munich, Germany, Email: knoll@in.tum.de

Abstract—The automotive domain is challenged by the increasing importance of Information Technology (IT) based functions. To show the possibilities of modern IT systems, a demonstrator car was developed in RACE (Robust and Reliant Automotive Computing Environment for Future eCars) based on a completely redesigned E/E architecture, which supports the integration of mixed-criticality components and offers features like Plug&Play. This paper presents the architecture and components of this vehicle prototype, which is equipped with modern systems such as Steer-by-Wire without mechanical fallback. It was designed to support future driver assistance systems, e.g. to carry out autonomous parking maneuvers onto an inductive charging station, a task, which is hard to achieve accurately enough for a human driver. Therefore, a special emphasis lies on the description of the sensor set for automated operation.

I. Introduction

In the automotive domain, two trends can be observed: First, more and more mechanical components will be substituted by mechatronics solutions which offer additional degrees of freedom but are more complex in software and embedded solutions [1]. One example is the introduction of a Steer-by-Wire system without mechanical fallback. Second, an increasing level of autonomy of driver assistance systems up to full automation will be integrated [2].

The main aim of the Project RACE[1] was to develop a new Electric / Electronic (E/E) architecture with all hardware and software components that will support the trends described above. RACE aims for reducing the amount of Electronic Control Units (ECU's) in modern vehicles as well as providing an E/E architecture with a run-time environment for fail-operational behavior which is needed to meet future homologation requirements [3]. Additionally, Plug&Play capabilities are available to be able to add software and hardware components after Start of Production without loosing the fail-operational properties.

This paper explains the architecture and the components of an electric vehicle demonstrator, which was built to illustrate the capabilities of this new E/E architecture. To address the requirements for advanced software functions such as intelligent energy management or a function that autonomously parks

Fig. 1. The RACE demonstrator vehicle

TABLE I
VEHICLE DATA

Weight	1.250 kg
Power	126 kW (240 kW peak)
Torque	1.000 Nm (2.500 Nm peak)
Battery voltage	720 V
Capacity	27 Ah
Power	19.44 kWh

the car onto an inductive charging station, not only software components have been developed, but also a compatible sensor set capable of detecting a charging station and obstacles has been selected and integrated. We also considered the trend seen in autonomous vehicle design (for example [4]) that none of the sensors should be mounted outside of the chassis parts but integrated into the vehicle.

In the following, the article will give an introduction to the RACE platform in Section II, explain the implemented E/E architecture in Section III, before listing the vehicle components in Section IV. In Section V, the sensor set to realize the autonomous parking function is explained. This information can help other researchers designing their own automated vehicles. As an example, section VI briefly explains the software architecture for the autonomous parking function before section VII gives a conclusion.

---

[1]http://www.projekt-race.de/en, a cooperation of Siemens AG, fortiss GmbH, TRW Automotive GmbH, AVL GmbH, Fraunhofer AISEC, Institute for Aerospace Systems (ILS), RWTH Aachen and Technische Universität München

IEEE computer society

II. THE RACE PLATFORM

The development of the RACE architecture was based on the design principles for future automotive E/E architectures proposed in [5]:

1) A flexible E/E architecture with a centralized topology and scalable ECU's
2) highly integrated mechatronic components
3) a standardized communication backbone
4) a Run-Time Environment (RTE) which guarantees timing, fault-tolerance and security and enables Plug&Play capabilities

A. Hardware components of the RACE Platform

1) Central Platform Computer (CPC): The CPC consists of a scalable set of Duplex Control Computers (DCC's) which are connected by an inner RACE network communication ring. Each DCC is a control unit with two execution lanes of which each one consists of a Xilinx Zynq FPGA with an integrated Dual ARM Cortex-A9™Core with up to 800 MHz and 1 GB DDR3 RAM, and Ethernet ports for network communication and testing (see IV-H2) [6]. Dependent on different Automotive Safety Integrity Levels (ASIL) from QM to ASIL-D and fault tolerance requirements (fail-silent or fail-operational), functions can be executed either on one lane of one DCC, on two lanes of one DCC or even redundantly on multiple DCC's [7]. This scalable redundancy provides cost efficient execution tailored to the requirements of the function.

2) RACE Gateway: To support the integration of non RACE compatible sensors and actuators into the RACE system, RACE supports the concept of Gateways. The Gateways connect sensors and actuators with interfaces such as CAN, Ethernet, RS232 as well as digital and analog signals.

3) RACE network: The RACE network is based on IEEE 802.1 Time-Sensitive Networks (TSN)[2] and uses standard Ethernet ports. It supports BroadR-Reach technology using cheap and lightweight single twisted pair cables [7]. Strict priority scheduling is used to separate different kinds of data, like critical and non-critical data. To minimize the influence between network data with mixed criticalities and mixed transmission rates, the preemption mechanism standardized in TSN is used [7]. Multiple Virtual Local Area Networks (VLANs) are used to define disjoint path for critical data and to protect the network rings from circulating packets [8]. The network guarantees that all sensors and actuators are accessible everywhere in the vehicle, and that critical data is transmitted in a deterministic period of time.

B. Run-Time Environment (RTE) of the RACE Platform

To achieve easy and safe integration of safety-critical automotive software functions, a RTE containing configurable safety mechanisms was developed.

The RTE of the RACE E/E architecture steers the interaction and communication between the components in the system, provides a basis for the safe execution of future automotive
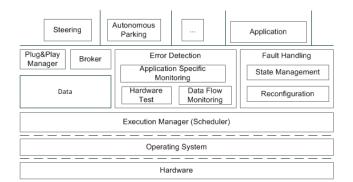

Fig. 2. Layers of the RACE Run-Time Environment (RTE)

software functions, and supports fault tolerance up to fail-operational behavior. A full version of the RACE RTE runs on the DCC's and a tailored version providing rudimentary functionality is also used on the RACE Gateways.

The RTE uses a data-centric communication approach. A publish-subscribe paradigm [9] enables data-centric design by defining topics and attributes, which are used to refer to data unambiguously. The topics are used for the communication among software components over hardware boundaries. The data-centric approach enables the decoupling of functions from the infrastructure components.

Besides mechanisms provided by common run-time systems such as real-time deterministic scheduling and data exchange services, the RTE provides a set of safety related mechanisms [10], [11]. In this way, a clear separation between functional and extra-functional properties such as safety is achieved. The safety mechanisms provide error detection via the monitoring mechanisms such as memory tests, CPU tests, data flow monitors or function specific monitors (Figure 2). Fault handling is achieved by reconfiguration and unified state management. The reconfiguration service enables adaptive graceful degradation and controlled function migration scenarios, in case of failures in the system [12]. In order to ensure separation of mixed-critical functions on one single control unit, the RTE is required to provide mechanisms for both spatial and temporal separation of the resources. In our project we used the separation mechanisms of the PikeOS[3] operating system.

III. E/E ARCHITECTURE

Figure 3 shows the E/E architecture of the demonstrator vehicle, with its core component, the Central Platform Computer, consisting of five DCC's interconnected via RACE network. Smart sensors and actors are connected to this CPC. With the exception of the power inverters, which were redesigned to have a direct interface to the RACE platform, the components shown in Figure 3 were only available with their native interface (mostly CAN). Therefore, they consist of a RACE Gateway plus the sensor / actuator with its native interface. If all components would be available with a RACE interface

---

[2]http://www.ieee802.org/1/pages/tsn.html

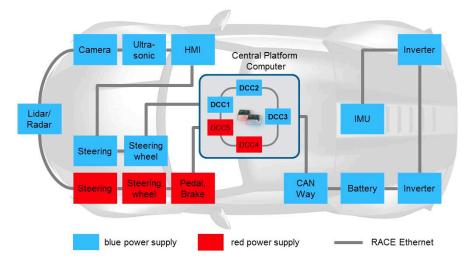[3]https://www.sysgo.com/products/pikeos-rtos-and-virtualization-concept/

Fig. 3. E/E Architecture with Central Platform Computer

in future, this architecture will be simplified dramatically. To point out the architecture which can be realized with RACE in future while describing today's vehicle at the same time, the native interfaces of the components are not drawn in Figure 3. Each block should be rather looked at as a smart sensor and actor directly connected to the RACE platform. The details of the components are described in section IV.

To support fail-operational behavior, safety critical components have redundant counterparts with a separate power supply. This is represented with a blue and red block in Figure 3. In our case, the CPC (with 3 DCC's in the blue and 2 in the red power supply) as well as the Steer-by-Wire system are powered by both power supplies.

IV. VEHICLE COMPONENTS

The RACE demonstrator (Figure 1) is based on a Roding Roadster Electric[4] [13], which was modified to integrate the RACE platform and to enable automated operation. The vehicle is an electric sports car with wheel hub motors built in a lightweight carbon chassis and has a full Drive-by-Wire system and sensors for environmental perception. In the following, the vehicle components are described in detail.

A. Power supply

1) High voltage power supply: A 720V Lithium-Ion battery[5], which is mounted in the back and the middle tunnel of the vehicle, powers the vehicle. Its maximum pulse charge current of 162 A (6 C) make the battery especially suited for recuperative charging. A total of 19.44 kWh lead to an estimated vehicle range of about 120 km.

The battery has its own Battery Management System (BMS) with a CAN Interface which is connected to a RACE Gateway. With this setup, it is possible to implement additional energy management systems on the Central Platform Computer like the one developed in [14].

The battery is charged using two water-cooled chargers [15] with 3,7 kW each, and has a Combined AC/DC Inlet "Type 2" to be charged with a IEC 62196-2 connector. A mobile charging device[6] can be connected to a standard 16A socket[7].

2) Low voltage power supply: For the sake of redundancy, two electrically isolated 12 V batteries power the low voltage components. The two 12V batteries are charged through two DC/DC 2kW converters[8].

B. Motorization

Two wheel hub motors [16] in the rear wheels can create 2 x 63 kW in permanent operation and up to 2 x 120 kW peak power. Wheel hub motors make a gearbox or a clutch redundant. They are constructed without mechanical brakes in the rear axle, but are designed to create the braking torque directly in the in-wheel motors, which enables recuperation of the kinetic energy in 70% of all braking processes [13].

The power inverters as an interface to the wheel hub motors were tailored for RACE, meaning that they support the RACE protocol and can be directly connected to the RACE infrastructure.

C. Steer-by-Wire

Core of the Steer-by-Wire system is a modified version of a Paravan Space Drive II[9] steering. It consists of a steering servomotor unit, which is connected to the steering rod and acts together with a Paravan ECU as a smart actor. All sensors and servos, as well as communication lanes, are redundant [3].

---

[4]http://www.roding-automobile.de/en/roding-technologies/references/roding-roadster-electric.html

[5]http://www.gaia-akku.com/en/batteries.html

[6]http://www.wallb-e.com/

[7]Although one of the showcases, for budgetary reasons, the vehicle is not equipped with an inductive charging station

[8]http://www.absopulse.com/Absopulse_DC_DC_Converters.php

[9]http://www.paravan.de/en/mobility-solutions/paravan-drive-by-wire-systems.html

Attached to the steering wheel, a sensor unit reads the steering wheel angle and communicates the values to another Paravan ECU. A force feed back servo motor is used to generate feedback torque to create a realistic driver experience. Both the steering wheel sensor / actuator and the steering actuator are redundantly connected via CAN to the RACE network and communicate with the CPC.

The interface to the steering system was designed in a way that the virtual wheel angle of a corresponding bicycle model can be demanded. To allow automated driving in parallel to manual operation, an interface with three operating modes had to be defined:

In manual operation mode, the steering wheel angle is directly converted into a steering angle in a feed-forward control loop, and the control deviation is transformed into feedback torque on the force feedback actuator. Being in feed-forward control mode helps creating a realistic user sensation in manual mode, but means that high control deviation is allowed, which would make accurate trajectory control in automatic driving impossible. In automatic operation mode, the steering angle is created by the trajectory controller and commanded to the actuator in a feedback control mode. The force feedback actuator is in a control mode to move the steering wheel according to the wheel position. User interaction measured as torque on the steering wheel is detected to let a decision module decide if the automatic operation should be interrupted and a transition to manual mode should be started. A third mode for manual trajectory based operation was designed to realize a scenario investigated during the project. Here, instead of feed-forwarding the steering wheel angle directly as in manual operation, the value should be used to generate trajectories, which are realized by the trajectory controller. The introduction of this mode was necessary taking into consideration that the actuator, unlike in normal manual mode, operates in a feedback control mode to be able to perform accurate trajectory control.

D. Future Brake System (FBS)

As part of the project an intelligent brake actuator was developed [17] , which integrates brake force generation and wheel slip control into one single device. It is especially designed for operation in electric vehicles, due to the fact that it operates without the need of an external vacuum.

A RACE Gateway connects the brake system via CAN to the RACE platform. Brake and accelerator pedals are also connected to this Gateway. The FBS has a mechanical fallback and therefore the RACE components are not redundant.

E. Accelerator Force Feedback Pedal

The vehicle is equipped with a prototype version of an Accelerator Force Feedback Pedal (AFFP) as presented in [18]. With this pedal it is possible to produce haptic feedback like a programmable counter force as well as vibrations on the pedal, which creates many possibilities for developing future ADAS with an additional way of driver interaction.

F. CANWay

The CANWay is the gateway to vehicle functions in the body control module like lights, indicator lights and wipers.

G. Human Machine Interface

A Nexus 7 touchscreen tablet mounted in the dashboard of the vehicle acts as Human Machine Interface (HMI).

H. Additional components for prototyping

1) Emergency stop: To enable safe operation of the prototype vehicle, an emergency stop concept is implemented for both manual and automated driving. For manual operation, an emergency button inside the vehicle cutts off the high voltage circuit and therefore switches off the engines. In case of unwanted behavior due to software of hardware errors, the driver can press the emergency button and use the mechanical brakes to stop the car. For automated including unmanned driving, the car was additionally equipped with a wireless safety system[10]. The emergency button on the remote control unit not only disables the high voltage circuit but also activates the electronic parking brake. Since automated driving is restricted to parking maneuvers with low speed and no unmanned driving is permitted at higher speeds, this is a feasible solution.

2) Testing interface: For testing purposes during the prototype phase including fault injection tests, monitoring and logging of measurement data, and for loading software updates onto all DCC's and RACE Gateways, an Advantech ARK embedded PC, which acts as server, was integrated in the vehicle. The server is connected via switches to the test port of all RACE components.

V. SMART SENSORS FOR AUTOMATED DRIVING

A sensor set had to be identified, which gives the possibility to locate a charging station as well as to enable Adaptive Cruise Control. In other words, being able to detect a leading vehicle in highway traffic scenarios.

As found in [19] a maximum lateral tolerance on the parking position of about ±10cm is needed for inductive charging. Also, during this parking scenario, the vehicle should be able to deal with static and dynamic obstacles.

Looking at other autonomous vehicles [20]–[22], they have a sensor set consisting of various radars, lidars, ultrasonic sensors and stereo or mono cameras mounted on the roof top or outside the chassis. Recent approaches for autonomous driving research platforms [23] are fulfilling our requirement to build a vehicle with sensors which are not easily visible.

For the parking scenario in this project, meaning limited speed and no public traffic situations, a limited and therefor cheaper subset of sensors is sufficient than for the full traffic scenarios other vehicles are addressing.

The initial targeted solution only consisted of using a single front camera for the target localization, ultrasonic sensors for the obstacle detection during parking maneuvers, and a radar to detect a leading vehicle.

[10]http://www.dold.com

The method for marking and localizing the target of the autonomous parking maneuver was to add Augmented Reality (AR) markers[11] to the inductive charging station and to use off-the-shelf image processing libraries to calculate the position of the marker relative to the vehicle.

Early experiments showed, that this solution was not robust and accurate enough. Especially because of the sensitivity to changing light conditions outdoors, false detections, as well as inaccurate distance readings in the purely camera based solution, lead to the conclusion that without significant software engineering effort, a lidar is necessary to be able to detect the charging station accurately enough.

In order to accurately control the vehicle along the calculated trajectory, it is essential to have a precise positioning system. An advantage of the wheel hub motors is, that they come with highly resolved encoders with 4096 ticks per revolution, acting as precise wheel speed sensors. To improve the calculation of ego motion data, an Inertial Measurement Unit (IMU) was installed.

Another requirement was to be able to show the Plug&Play capabilities of the RACE Platform. A use-case for this feature could be a customer, who wants to enhance his vehicle with further functionality by adding or exchanging a sensor.

In the following, the sensors which are mounted into the vehicle are described more in detail:

A. Ultrasonic sensors

A total of 12 BOSCH URF6 ultrasonic sensors were placed in the vehicle to provide information of close obstacles in a 360 degree view around the car. Each sensor has a Field of View (FoV) of 120 degrees horizontal and 60 degrees vertical view, and a range of about 2.5 m.

The sensors are connected to a Neobotix USboard [12] which deals with issues like (analog) low level communication and sensor cross talk. A RACE Gateway connects the USBoard's CAN interface to the RACE platform.

B. Camera

As mentioned above, the camera should be able to detect AR tags. No cameras were available on the market which have an embedded solution for this task, therefore a 'smart sensor' solution had to be developed.

This consists of an industrial mono camera and a device to perform the image processing. The camera chosen is an IDS uEye UI-1240SE-C-HQ[13] with a 1/1.8" CMOS Sensor and a resolution of 1280x1024 and a global shutter, together with a Fujinon 6mmDF6HA-1B lens. The camera is facing forward, mounted to the rear view mirror and can achieve 56 degrees FoV horizontal.

It is connected via USB2.0 to a fan-less embedded PC Advantech ARK 3440-A2 (with 8GB RAM and an i7 chip), on which the image processing and AR tag detection is implemented.
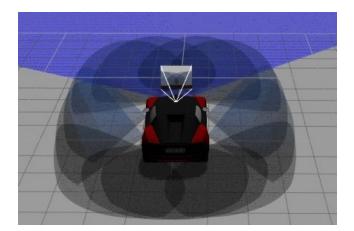
---

[11]http://www.hitl.washington.edu/artoolkit/
[12]http://www.neobotix-roboter.de/komponenten.html
[13]https://en.ids-imaging.com

Fig. 4. Illustration of the Field of View of the prototype

C. Radar

A Continental ARS-330 with a range of up to 200m and a FoV of 56 degrees [24] is able to detect a leading vehicle and provides this extracted object data on its CAN Interface. A RACE Gateway connects the sensor to the RACE platform.

D. Lidar

An IBEO LUX[14] provides object data as well as sensor raw data and was the only one found on the market which can be neatly integrated into the vehicle. The sensor is equipped with an Ethernet interface for all data including sensor raw data, as well as a CAN interface to provide ego motion data to the sensor and get extracted obstacle data. The sensor covers a vertical FoV of 3.2 degrees in 4 scanning layers, which is useful to deal with uneven surfaces and pitching movements, a FoV of up to 85-110 degrees and a range of 200m max. Again, a RACE Gateway builds the interface to the RACE platform.

E. Inertial Measurement Unit / GPS

To improve the ego motion data from the odometry, an XSens MTi-G700 GPS-INS[15] sensor was installed. This sensor is an integrated GPS and Inertial Measurement Unit (IMU) with an Attitude and Heading Reference System (AHRS) processor which offers embedded data fusion.

VI. Software architecture for autonomous parking

Figure 5 illustrates the software architecture of the autonomous parking function as one example of function development on the RACE platform. The sensor data is fused to a grid map with obstacle information and a target position for the parking maneuver. The planning component [25] creates a resulting trajectory, which is commanded to a trajectory controller that controls the actuator output for the Drive-by-Wire components. A special feature of the RTE is that different

---

[14]http://www.ibeo-as.com/
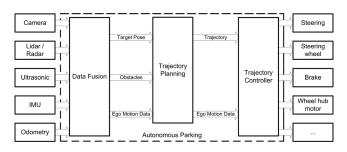[15]https://www.xsens.com/products/mti-g-700

Fig. 5. Software architecture of the autonomous parking function

actions regarding the trajectory generation can be taken based on the Quality of Services indications delivered by its safety mechanisms. For example, in case the RTE indicates sensor data as unreliable (e.g. due to communication issues), an emergency stop trajectory will be calculated in the planning module. The availability of generic safety mechanisms in the RTE contribute to dramatically reduce development time for safety critical functions.

VII. CONCLUSION

Within the project duration of 3 years, an electric vehicle was built in which a completely redesigned automotive E/E architecture was integrated. This includes the development of all electronic hardware parts, the RACE RTE, as well as all other software components from scratch. A sensor set capable of accurately detecting an inductive charging station was identified and incorporated into both the vehicle as well as to the new RACE environment.

The prototype demonstrates many capabilities of the new E/E architecture, for example, to integrate components of different safety integrity levels: safety critical functions together with non safety-critical functions. Especially mentioned should be the integration of a Steer-by-Wire system without mechanical fallback. The demonstrator vehicle is operational for manual driving and ready for the integration of the driving assistance functions like autonomous parking, which could already be developed in the RACE environment and successfully demonstrated.

ACKNOWLEDGMENT

REFERENCES

[1] O. S. Kaiser, H. Eickenbusch, V. Grimm, and A. Zweck, "Zukunft des Autos," *Zukünftige Technologien*, vol. 45, 2008.
[2] SAE, "SAE document J3016 - Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems," 2014. [Online]. Available: http://standards.sae.org/j3016_201401
[3] L. S. Lutz, T. Tang, and M. Lienkamp, "Analyse der rechtlichen Situation von teleoperierten (und autonomen) Fahrzeugen," 2012.
[4] J. Ziegler *et al.*, "Making Bertha Drive - An Autonomous Journey on a Historic Route," *Intelligent Transportation Systems Magazine, IEEE*, vol. 6, no. 2, pp. 8–20, 2014.
[5] S. Chakraborty *et al.*, "Embedded Systems and Software Challenges in Electric Vehicles," in *In Design, Automation and Test in Europe (DATE)*, 2012.
[6] C. Bachmann, "RACE D2.1 Hardwarearchitekturkonzept," 2013.
[7] S. Sommer *et al.*, "RACE: A Centralized Platform Computer Based Architecture for Automotive Applications," in *2013 IEEE International Electric Vehicle Conference, IEVC 2013*. IEEE, 2013.
[8] M. Armbruster *et al.*, "Ethernet-Based and Function-Independent Vehicle Control-Platform: Motivation, Idea and Technical Concept Fulfilling Quantitative Safety-Requirements from ISO 26262," in *Advanced Microsystems for Automotive Applications 2012: Smart Systems for Safe, Sustainable and Networked Vehicles*. Springer, 2012, pp. 91–107.
[9] G. Pardo-Castellote, "OMG Data-Distribution Service: Architectural Overview," *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, 2003.
[10] J. Frtunikj *et al.*, "A Safety Aware Run-time Environment for Adaptive Automotive Control Systems," *Embedded Real-Time Software and Systems, ERTS2*, 2014.
[11] J. Frtunikj, M. Armbruster, and A. Knoll, "Data-Centric Middleware support for ASIL decomposition in open automotive systems," in *Automotive meets Electronics*, VDE/VDI Gesellschaft Mikroelekonik Mikrosystem- und Feinwerktechnik (GMM) in Dortmund, 2014.
[12] K. Becker, B. Schätz, M. Armbruster, and C. Buckl, "A Formal Model for Constraint-Based Deployment Calculation and Analysis for Fault-Tolerant Systems," in *12th International Conference on Software Engineering and Formal Methods (SEFM)*, 2014.
[13] Siemens, "Fact Sheet — The Roding Roadster Electric as a Research Platform," 2013. [Online]. Available: http://www.siemens.com/press/pool/de/events/2013/corporate/2013-06-energiewende-dialog/factsheet-roding-e.pdf
[14] T. Isermann, S. Sester, and A. Monti, "A Multi-Agent Based Energy Management System for Electric Vehicles," in *Vehicle Power and Propulsion Conference (VPPC), 2014 IEEE*, 2014, pp. 1–6.
[15] Brusa, "NLG513 - On-Board-Charger." [Online]. Available: http://www.brusa.eu/fileadmin/Diverses/Download/Datenblaetter/BRUSA_DB_EN_NLG513.pdf
[16] Siemens, "Power where it's needed." [Online]. Available: http://www.siemens.com/innovation/apps/pof_microsite/_pof-spring-2012/_html_en/electric-vehicles.html
[17] S. Grieser-Schmitz, "Future Brake System (FBS) von TRW - Vakuumloser Bremskraftverstärker mit integrierter Radschlupfregelung," Munich, 2015.
[18] C. Leone, A. Arcati, and G. Schmitt, "Active Accelerator Pedal as Interface to Driver," *ATZ worldwide*, vol. 112, no. 4, pp. 36–39, 2010.
[19] H. Barth *et al.*, "Concept Evaluation of an Inductive Charging System for Electric Vehicles," in *3rd European Conference Smart Grids and E-Mobility, Munich, Germany*, 2011.
[20] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer, 2009.
[21] C. Urmson *et al.*, "Autonomous Driving in Urban Environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
[22] S. Kammel *et al.*, "Team AnnieWAY's Autonomous System for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.
[23] J. Wei *et al.*, "Towards a Viable Autonomous Driving Research Platform," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 763–770.
[24] Conti, "Datasheet ARS300." [Online]. Available: https://www.conti-online.com/www/download/industrial_sensors_de_en/themes/download/ars_300_datasheet_en.pdf
[25] C. Chen, M. Rickert, and A. Knoll, "Combining Space Exploration and Heuristic Search in Online Motion Planning for Nonholonomic Vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, 2013.