# Making Bertha Cooperate–Team AnnieWAY's Entry to the 2016 Grand Cooperative Driving Challenge

Ömer Şahin Taş, Niels Ole Salscheider, Fabian Poggenhans, Sascha Wirges, Claudio Bandera, Marc René Zofka, Tobias Strauss, J. Marius Zöllner, and Christoph Stiller

***Abstract*— This paper presents the concepts and methods utilized by Team AnnieWAY for the 2016 Grand Cooperative Driving Challenge. The paper introduces the automated vehicle *BerthaOne*. The vehicle, even though being based on the Bertha platform, distinguishes itself from its siblings by its software modules and algorithms. We, therefore, describe its system architecture and algorithms for perception, cooperation and motion planning. In Particular, we present a motion planner that plans different maneuvers flexibly by augmenting the cost function with situation specific cost terms. We subsequently describe the requirements of the 2016 GCDC and evaluate our performance during the competition.**

***Index Terms*— Automated driving, cooperative driving, vehicle-to-vehicle (V2V) communication, vehicle-to-infrastructure (V2I) communication, vehicle-to-x (V2X) communication.**

Fig. 1. Forming platoons is essential for cooperative maneuvering. The first cooperation and safety tests for the 2016 GCDC were performed at the IDIADA–Applus facilities in Spain, during the workshop week in April 2016.

## I. Introduction

RESEARCH on intelligent vehicles has made tremendous progress in the last decades [1]. The focus has recently shifted on realizing fully automated driving where pure sensor-based perception is only partially adequate to meet the demanding requirements, especially in urban environments. Exchanging fused sensor data and broadcasting driving intentions, and in a broader sense, negotiating motion for their mutual benefit has the potential to improve driving safety and efficiency in terms of both individual energy consumption and the overall traffic flow [2].

An overview on available cooperative driving functions is provided in [3]. The work identifies generic and novel functions for not only cooperative, but also collaborative driving and further inspects their needs on communication. Studies on cooperative driving typically fall in two categories: cooperative lane change and cooperative maneuvering at intersections. Up-to-date surveys on both aspects are presented respectively in [4] and [5].

Cooperative driving and V2X communication have been studied and applied by many researchers. The main limitation of existing applications is the imposed homogeneity of vehicles: they are developed mostly by the same group of researchers eventually resulting in the same design patterns. However, this contradicts with the fundamental requirement of cooperation, namely the interoperability: distinct frameworks and platforms need to operate in harmony reliably. In order to address this requirement and boost the development of cooperative automated vehicles the first Grand Cooperative Driving Challenge (GCDC) was organized in 2011 [6]. This competition solely focused on platooning of cooperative vehicles and evaluated performance on overall platoon driving performance and shockwaves damping. Five years later, the objective of the second GCDC was to go a step further than only maintaining platooning (cf. Figure 1). This time the challenge concentrated on both of the aforementioned categories and was composed of two scenarios for the evaluation: cooperative lane change and cooperative intersection maneuvering [7]. In order to participate in the competition, it was an obligatory requirement to be able to form an emergency corridor for approaching rescue vehicles. While this was handled as an additional task, the performance in this scenario was not considered in the final benchmarking.

This paper continues with Section II in which we introduce our team and outline our vehicle, a Mercedes-Benz E-Class in

Manuscript received November 30, 2016; revised February 28, 2017 and May 17, 2017; accepted July 2, 2017. Date of publication October 2, 2017; date of current version March 28, 2018. This work was supported in part by the contribution of the German Collaborative Research Center SPP 1835-Cooperative Interacting Vehicles granted by the German Research Foundation and in part by the framework of the Profilregion Mobilitätssysteme Karlsruhe, which is Funded by the Ministry of Science, Research and the Arts and the Ministry of Economic Affairs, Labour and Housing in Baden-Württemberg. The Associate Editor for this paper was C. Englund. *(Ömer Şahin Taş, Niels Ole Salscheider, Fabian Poggenhans, Sascha Wirges, Claudio Bandera, Marc René Zofka, Tobias Strauss, J. Marius Zöllner, and Christoph Stiller are contributed equally to this work.) (Corresponding author: Ömer Şahin Taş.)*

The authors are with the FZI Research Center for Information Technology, 76131 Karlsruhe, Germany, and the Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (e-mail: tas@fzi.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2749974

Fig. 2. *Bertha*, the flagship automated vehicle of Team AnnieWAY.

its augmented configuration, for the first time in a publication. We provide an overview of our hardware setup and software framework. In Section III, we give a brief description of our perception and localization algorithms, which are tuned for the requirements of the GCDC. The necessary communication modules are presented in Section IV. Section V reviews the scenarios and presents the planning algorithms utilized, before the human-machine interface of the vehicle is described in Section VI. The section is followed by Section VII, in which we demonstrate our simulation framework. We subsequently evaluate the performance of the overall system in Section VIII. The paper ends with highlighting conclusions and future research directives.

## II. System Overview

Team AnnieWAY is formed by researchers working on automated driving at the Karlsruhe Institute of Technology and the FZI Research Center for Information Technology, in Germany. Since its foundation in 2001, the research group has achieved many milestones in automated driving. Among others these are: being one of the finalists of the 2005 DARPA Grand Challenge in a joint team with the Ohio State University [8], being one of the finalists of the 2007 DARPA Urban Challenge in a university team of Karlsruhe and Munich [9], winning the 2011 Grand Cooperative Driving Challenge [10], and lastly in 2013, in collaboration with Daimler AG automated completion of Bertha Benz Memorial Route (BBMR) [11].

Since our participation in the 2011 GCDC, in which we competed with *AnnieWAY*, our 2006 VW Passat vehicle, both our hardware and software framework have experienced major revision. We upgraded to a modified Mercedes-Benz E-Class (W212), the *BerthaOne* (cf. Figure 2). The BBMR-mission started on this vehicle and later on, the platform was ported to two Mercedes-Benz S-Class vehicles and the BerthaOne was granted to our research group. The BerthaOne is equipped with a 195kW (261hp) diesel engine. For the sake of brevity, we simply refer to BerthaOne as *Bertha*.

Since the BBMR-mission Bertha has experienced significant modifications. Not only the majority of software modules have been rewritten, but also the software framework on which the software modules are running has been upgraded. This essentially makes Bertha a unique and up-to-date automated vehicle.

The system architecture of Bertha is displayed in Figure 3. In the following we will first present the hardware of the vehicle and then provide an overview on the software framework and main software components of the vehicle.

### A. Hardware

The vehicle hardware can be divided into two main categories: sensors and computers.

*1) Sensor Setup:* For robust and reliable perception Bertha covers its surrounding with redundant and complementary types of sensors [12]. The vehicle utilizes multiple cameras, radars and a lidar sensor. The position and coverage of the sensors are depicted in Figure 4. The frontview stereo vision is performed with two BlackFly PGE-50S5M-C cameras with Lensagon BM4018S118 lenses. While the sideview cameras use the same camera, they are equipped with Lensagon BM2920S118 "fisheye" lenses in order to cover a wider field of view. For traffic light detection we utilize a BlackFly PGE-50S5C-C color camera with the same lens as the stereo camera setup. Complemental to the camera based perception, we employ short and medium & long range radars to reliably detect the vehicles surrounding in adverse weather and lightning conditions. Bertha is equipped with the four layer lidar sensor Ibeo LUX4 to support cameras in terms of distance and radars in terms of precision. For global positioning we utilize high precision GNSS/inertial sensor, OxTS RT3000, and a low cost RTK-DGNSS, Ublox C94-M8P. V2V communication is realized with a separate 802.11p-antenna.

*2) Vehicle Computers:* The main computational work is done on a Linux based server (main machine) equipped with two 2.6 GHz 16 core Intel Xeon processors (E5-2640 v3), 64 GB memory and a Nvidia GeForce GTX 980 Ti graphics card. This machine runs the software framework hosting image processing, object fusion, situation understanding and planning algorithms.

Low level control of a planned motion is done on a separate, realtime onboard computer that is isolated from the main machine. In this way the direct operation of the vehicle is robust against system failures or process crashes that might occur during testing of immature software on the main machine. The onboard computer is also responsible for CAN bus-based communication with in-vehicle sensors and vehicle actuators. Actuator signals are passed in form of a desired acceleration to the underlying series ACC controller. Furthermore, the computer is responsible for the appropriate handling of emergency events in case of an emergency button event or driver take over.

Apart from the computers introduced above, the vehicle has a further computer for V2X communication. More details on communication hardware is provided in Section IV.

### B. Software Framework

The main computer uses the Robot Operating System (ROS) framework [13] to host processing algorithms. ROS allows individual software modules to be started as separate processes and provides a very flexible system configuration where the individual modules can be reconfigured, added or removed without the need to recompile or in most cases not even requiring a restart. Additionally it comes with a variety of tools for introspection and visualization. The main flaw of the current ROS framework is that it does not have any real time guarantees. When data loads reach the limits of

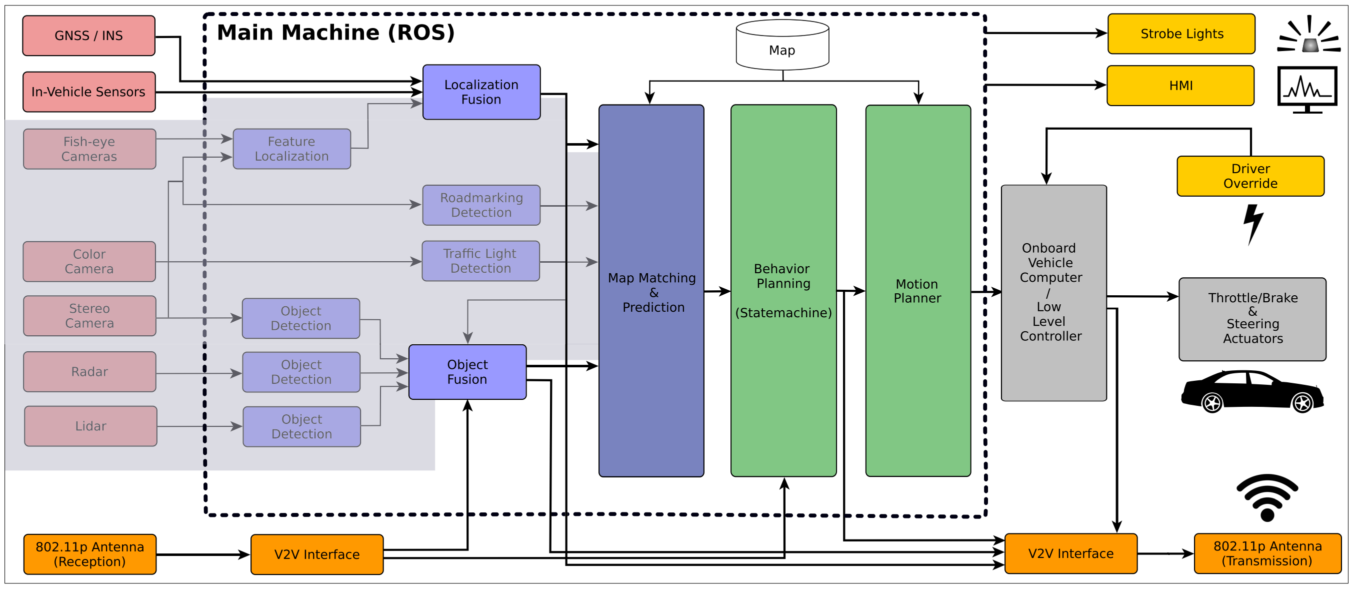


Fig. 3. System architecture of Bertha. The areas marked with gray highlight the redundancy provided through V2V and V2I communication. Notice that although the communication interface is drawn twice for receiving and transmitting, they physically correspond to the same unit and communication is bidirectional.

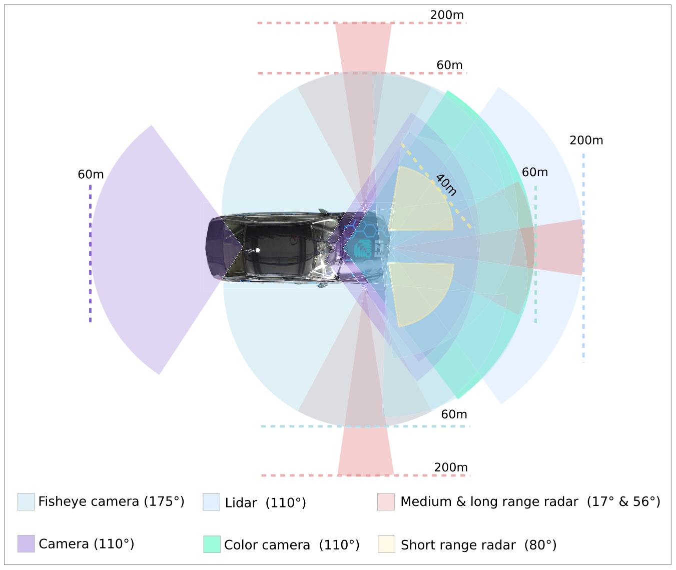


Fig. 4. Sensor coverage of Bertha.

bandwith, silent package loss can occur. However, studies show that ROS is in practice quite useful as an online system for automatic driving and has a significantly higher performance than current hard real time systems [14]. Also, the dynamically configurable structure makes it very simple to construct test environments. We used this feature to replace all sensor/actuator interfaces by a simulation environment for realistic offline testing, as described in Section VII.

In the following we will briefly explain the general software structure of the vehicle. We delve into the algorithmic details in the later chapters.

*1) Localization:* Bertha can rely on three distinct localization sources: GNSS/INS measurements, in-vehicle odometry and vision-only feature localization. We can choose between these sources and also fuse them to get a precise global position. The global position information is afterwards sent to the object fusion, map matching and V2V interfaces.

*2) Communication Interface:* This interface processes the received messages and sends them to the main computer via UDP connection. Motion and state information of other vehicles are passed to the Object Fusion module, whereas the environmental state information and the bidirectional communication are passed to the statemachine. The interface also broadcasts ego vehicle motion information together with the current vehicle state received from the statemachine, which includes the current lane, merging intentions, the most important object etc.

*3) Object Fusion:* The vehicle can perceive its environment from distinct sensors, where they may yield inconsistent information with respect to each other. The Object Fusion module resolves such ambiguities in the perception system, and fuses the data together with the information gained from intervehicular communication. It thereby monitors and filters unreliable global position measurements transmitted by other participants. The object list for a given timestamp is later passed to the Map Matching and Prediction module.

*4) Map Matching & Prediction:* The positions of the ego vehicle as well as objects received from the fusion module are either in map frame or in global frame. The motion data in this case does not behold any information about the street topology and the geodesic distances. The module assigns vehicle positions to their lanes and predicts their motion assuming constant acceleration, which are eventually passed to the Behavior Planner.

*5) Behavior Planner:* The behavior planner is implemented as a *statemachine* and is the core module of the vehicle's software system. It defines the operation modi of the vehicle and reflects the different states defined in the rule book of the GCDC. According to the environmental conditions, it creates a *decision base* from which state transitions are triggered. The environmental conditions can comprise motion of other objects and infrastructure-to-vehicle information.

*6) Motion Planner:* The module takes the state information from the statemachine and calculates the optimal motion. Depending on the current state, different goals and constraints are applied on the motion planner. The motion planner utilizes a local, continuous optimization method to find the best motion. The result of every planning run is a list of desired positions over time which is then converted to a reference acceleration and steering angle [11]. These are consequently passed to the onboard computer via a UDP/IP connection.

## III. Perception and Localization

Bertha is equipped with an array of sensors that use distinct measurement principles. This allows to compensate an inadequacy of one measurement principle with another one that yields more reliable results for a given condition. The performance judgement of the GCDC, on the other hand, is solely based on the broadcasted GPS positions of the participants, and hence, ameliorating inconsistent data over the V2V communication data would not yield any advantage in the evaluation. Considering the evaluation and the fact, that there comes an increased risk of bugs and failures with a more complex software, we decided to rely on intervehicular communication only. But as we are aware, that this was a solution tailored for a specific competition and that future automated vehicles will not come into the market without an enhanced and redundant perception system, we will introduce the algorithmic details of our perception system in the following.

### A. Vision Data Processing

Visual data is gathered in our vehicle by six cameras in total. The setup is intentionally flexible, in order to support a variety of image processing algorithms. The cameras are calibrated extrinsically and intrinsically in a bundle adjustment algorithm [15]. The very high resolution of 5 MP per camera, allows the selection of suitable undistortion procedures (pinhole, rectification, spherical camera models) for each algorithm, while at the same time providing a satisfactory resolution.

Spherical camera models are mainly used for accurate localization due to the large viewing angle. We can employ visual odometry [16] as well as visual global location based on a map of visual landmarks [17].

Bertha also has a stereo camera that is used for classical perception tasks, such as object detection with stixels [18] or the detection of road markings [19].

To detect traffic lights, a color camera is installed as well. In order to save computing time, only the regions that are useful for the detection of traffic lights are selected from the camera image. The large opening angle of the cameras allows detection even if the vehicle is directly under a traffic light. The detection and state estimation of the traffic lights is carried out using a Convolutional Neural Network [20].

### B. Radar and Lidar Processing

The vehicle contains five radar and one lidar sensor from which we generate tracks of static and dynamic objects in the vehicle surroundings. Whereas each radar interface already provides tracked objects, we perform a custom object segmentation for the lidar sensor. The designed lidar interface provides surface information as point clouds of the Point Cloud Library [21]. Based on that cloud, we perform Euclidean cluster extraction that yields segmented objects. The algorithm iterates through each point of the point cloud and associates it to the same class as the closest neighbor point in a certain radius. If there is no neighbor point within that radius, a new segment is created. To improve segmentation speed, the algorithm uses a kd-tree [22] created from the point cloud. Lidar object velocities, in contrast to radar objects, can however not be estimated in this stage.

### C. Dynamic Object Tracking

To perform a unified object tracking for different types of sensors and sensing principles, we use a generic object representation. This object representation consists of the objects' surface points, the geometric center position with its covariance estimate and the rotational and translational velocity with their covariance estimates.

The object fusion takes as input a set of objects at arbitrary times, performs sensor-interdependent association and tracking. The result is an object list that consists of improved information on the object state. As a first step within the fusion pipeline, objects are transformed in a common reference frame. Here we use the vehicle odometry frame, as we need continuous object trajectories. In the sensor-interdepent association step, we merge objects from different sensors using a nearest neighbor metric based on their geometric centers. In order to associate different measurements in time, we also use a nearest neighbor metric but based on a hull similarity measure. If a motion estimate already exists, we predict objects for the next time step and compare it to the measurement we received. If the object hull is similar to the predicted one, we associate the new measurement to the already existing track, otherwise we create a new one.

### D. Localization

The localization module can take IMU or position measurements as an input. These measurements can be taken from high precison GNSS/IMU, or from low cost GNSS and visual odometry [23], or from tightly coupled GNSS and visual odometry [24], or from vision only localization [25], or from lane marking based localization [26], or even from multiple of them. The filter internally uses a dynamic one track model and fuses the measurements with an unscented Kalman filter to a reference coordinate system. Whichever sources we rely on, the module typically uses two instances of localization-fusion filters. The first instance fuses odometry data only and publishes the resulting position in a *pseudo-global odometry frame* – or *map frame*, which has no global reference but has the advantage of being continuous. The second, cascaded one receives the fused odometry as well as all global position updates and subsequently fuses them to a globally referenced coordinate frame, which might in exchange be subject to jumps.

The filter is able to reject measurements that are too far from its current state, in order to reduce outliers. This is done using a Mahalanobis distance weighted with the filter's covariances. The filter can further keep a history of recent filter states and mesurements and eventually sort them into a measurement history based on measurement time. This helps with asynchronous or delayed measurement updates. Because delayed measurements can change the filter history, a measurement that was rejected initially might later on be revised as valid. In this case, the filter is reset to the state before the measurement and re-runs the history back to the current filter time. This way, old measurements can be filtered correctly. However, this correction can lead to discontinuities in the position. For this reason, the filter broadcasts its own diagnostic information, which originates from continuous comparison of the current speed, acceleration, steering angle and yaw rate with previous states. In case of exceeded thresholds, the filter will reset itself and override measurements in an attempt to become stable again.

### E. Map Matching & Prediction

This module processes the incoming position information from the object fusion module and transforms the motion information from Cartesian coordinates into Frenet coordinates [27]. As Frenet coordinates require a reference frame for the transformation, positions of other vehicles are matched to the lane they are driving on. Every lane at the competition site is stored in a map database in the form of polylines. These polylines were created in advance with Bertha during a manual drive along the route of the competition heat.

In order to associate vehicles to polylines, our initial approach was the utilization of nearest neighbors. However, as we decided to solely rely on intervehicular communication and turn off our own perception system, we had to discard our original approach. Because, when we only rely on the positions sent by participants, some of the information were too imprecise and partially subjected to a constant bias resulting in matching to wrong lanes. Instead, association of vehicles to a track was done by considering the Lane ID sent by the participating vehicles. In highway scenarios, all relevant vehicles were projected onto our lane to reliably estimate the time gap to the other vehicles. In the intersection scenario, we used the longitudinal component of each vehicle in Frenet frame to calculate the distance to the intersection point. This way, we could virtually project participating vehicles onto our lane and treat the problem of keeping a safe distance in the same way in all of the GCDC scenarios.

## IV. Communication

The vehicles that participated in the *Grand Cooperative Driving Challenge 2016* communicated with each other over the IEEE 802.11p wireless standard as the physical and data link layer in the OSI model. GeoNetworking is used at the network layer, while the Basic Transport Protocol (BTP) is used at the transport layer. Both protocols are developed by ETSI ITS (European Telecommunications Standards Institute – Intelligent Transport Systems) for V2X communication.

On top of this, multiple messages are broadcast to all other vehicles. These are CAM (Cooperative Awareness Messages), DENM (Decentralized Environmental Notification Messages) and iCLCM (i-GAME Cooperative Lane Change Messages). CAM and DENM are also standardized by ETSI ITS, while iCLCM is a message that was specifically introduced to meet the needs of the GCDC 2016. During GCDC however, all messages were sent with a non-standard frequency of 25 Hz. This increase of frequency was necessary because many teams relied only on the position information from the V2V communication and had no additional sensors.

Our communication computer consists of a small embedded board with an *Intel J1900* CPU. It is equipped with a *Compex WLE200NX* wireless card and runs *Ubuntu 14.04*. The kernel is a patched 4.8 mainline kernel to add the 5.9 GHz frequency bands used by IEEE 802.11p to the *ath9k* driver for the wireless card. We also updated *iw* and use patched versions of *crda* and *wireless-regdb* that contain the additional frequency bands. The transmitter power was set to 100 mW.

Everything from the network layer on upwards runs in user space. We use *udp2eth* for this which treats the UDP payloads it receives as Ethernet frames and the other way around.

We use the open source GeoNetworking stack[1] that was partly developed within i-GAME project for the upper layers. Its *vehicle adapter* sends the contents of the GeoNetworking messages to a ROS node on our main computer via UDP. It also receives message contents from the same node and generates corresponding GeoNetworking messages. This node in turn communicates with the object fusion and localization fusion modules and also with the state machine for behavior generation.

We evaluated the range of the V2V communication between two of our vehicles when there are no occluding object in the line-of-sight. Up to a distance of 100 m, the packet loss always stays below 20 % in time windows of 1 s. From 100 m to 300 m the packet loss starts to increase and blocks of packages get lost so that a reliable communication is not always possible. From 300 m onwards, the communication fails completely and all packages are lost.

## V. Planning

### A. Behavior Planning and State machine

The behavior of our vehicle is determined by a state machine. During GCDC 2016, this state machine implemented the interaction protocols for the different scenarios. State changes can be triggered by communication with other vehicles (e. g. when we receive the safe to merge flag) or by our perception (e. g. when we cross the edge of the lane).

This is in accordance with the interaction protocols of GCDC 2016 that require the vehicles to be triggered by

[1]https://github.com/alexvoronov/geonetworking

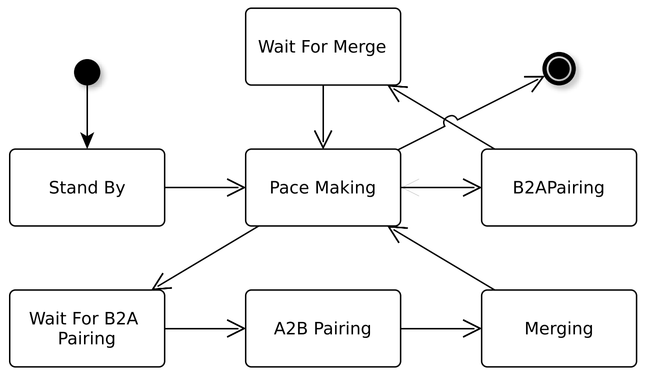


Fig. 5. State machine implemented for Scenario 1.

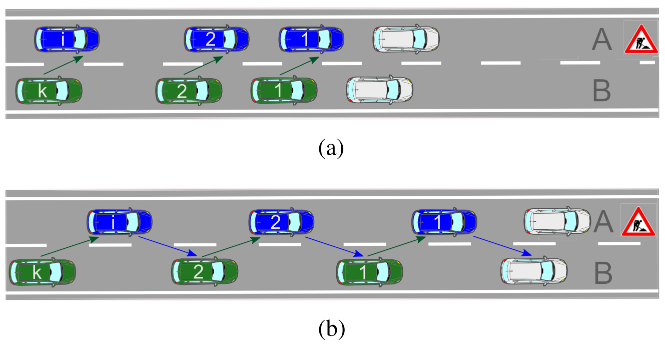


Fig. 6. The pairing states in the highway scenario. The vehicles in competing platoons are shown in blue or green, whereas the OPCs are shown in white color. (a) In the *B2A Pairing* state the vehicles on the right platoon choose a forward partner on the left lane. They start to keep a safety distance to them so that a gap can be opened. (b) In the *A2B Pairing* state the vehicles on the left platoon choose the leader of their backward partners as their respective forward partner. The first vehicle on the left lane starts to keep a safety distance to its forward partner so that the gap opens.

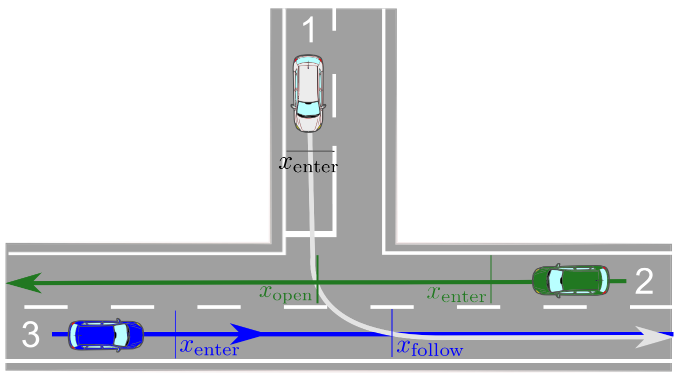


Fig. 7. Intersection scenario of the GCDC. The OPC is again shown in white color.

wireless communication. It also means that we might not be able to proceed with the current scenario if we do not receive correct interaction protocols from our partners.

*1) Cooperation on Highway:* The first scenario contains a highway scenario where the left lane is blocked by a construction site. Two platoons of autonomous vehicles therefore have to merge on the right lane so that they can pass the construction site.

In this scenario, our state machine starts in the *Stand By* state as shown in Figure 5. Upon receipt of the *Start Platoon A/ Start Platoon B* signal (depending on the platoon our vehicle is in) the state changes to *Pace Making*. Starting with the organizer pace cars (OPC), which are the vehicles of the organization to maintain fair group evaluation and are placed first in a platoon, the vehicles will form platoons, start driving and keep a fixed distance to their leader vehicle.

At some point, our vehicle receives the *Roadwork Ahead* message and a *Merge Request* from the platoon leader on the left lane. This indicates that the platoons should merge. If our vehicle is on the right lane, the state machine will go into the *B2A Pairing* state. In this state, the forward partner is determined and our vehicle makes a gap for it (cf. Figure 6a). Once the gap is open and all pairing IDs are set up correctly, the state machine will go into the *Wait For Merge* state. Our vehicle sends the *Safe To Merge* flag as long as it is in this state. As soon as the merging vehicle indicates that it has finished merging the state will change back to *Pace Making*. Our vehicle platoons with the new leader and is ready to start again.

If our vehicle starts on the left lane, the state machine will go into the *Wait for B2A Pairing* state when our vehicle receives the *Roadwork Ahead* and *Merge Request* messages. Our vehicle will continue to platoon with its leader. If our backward partner has indicated that it chose our vehicle as forward partner and if our vehicle is the leader, the state changes to *A2B Pairing*. In this state, our vehicle starts to keep the desired distance to the forward partner on the right lane (cf. Figure 6b). Once it receives the *Safe To Merge* flag from the backward partner, the state changes to *Merging*. The vehicle starts with the merging process immediately. When the merging is finished the state changes back to *Pace Making* and we continue to platoon with the new leader.

*2) Cooperative Intersection:* Scenario two takes place on a T-junction. Three vehicles approach the center of the crossing so that they would collide in the center if they continued to drive with constant velocity. The goal of this scenario is that all vehicles leave the intersection as fast as possible without violating minimum safety distances or maximum velocity constraints. The vehicle on Lane 1 (cf. Figure 7), namely the OPC, has right of way.

Our state machine contains only two states for this scenario. It starts in the *Stand By* state and waits for a *Start Platoon* signal. As soon as this signal is received the state changes to *Intersection* and stays there until the end of the scenario. In this state the behavior of the vehicle is completely determined by the planner.

The planner has several constraints to make sure that our vehicle leaves the competition zone as fast as possible. The first constraint is that the point $x_{\text{enter}}$ has to be passed with a velocity of 30 km/h exactly 20 s after receiving the start signal. This is mandated by the rules of Scenario 2.

If our vehicle starts on Lane 2 we add the constraint that forces the vehicle to pass $x_{\text{open}}$ with 30 km/h right after the OPC has crossed the intersection with safety distance. This makes sure that we can leave the intersection as fast as possible.

If our vehicle starts on Lane 3 the additional constraint serves to follow the OPC with the safety distance after $x_{\text{follow}}$.

The planner is described in more detail in Section V-B.3.

### *B. Motion Planning and Control*

The evaluation criteria of the competition are recapitulated in the previous subsection. The demands on motion planning are intertwined and require simultaneous consideration, eventually resulting in a nonlinear, multicriteria problem. If the individual summands posing the problem are convex, the global optimum can be found with a local optimization scheme in real time. As the workspace is not discretized, the solution further does not behold any inherent inaccuracy [28].

*1) Problem Definition and Formalization:* The goals of the distinct scenarios of the competition pose scenario-specific requirements on the motion planner. However, these specific requirements share a common base that is valid for all of the scenarios. We will first formalize our motion problem for the common base and later extend its terms for the scenario-specific requirements.

We define the optimal trajectory $\mathbf{x}(t) = (x(t), y(t))^{\mathsf{T}}$ for the rear axle center of the vehicle which we get through the minimization of an objective functional of the form

$$J\left(\mathbf{x}\left(t\right)\right) = \int_{t_0}^{t_0+T} L(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \dddot{\mathbf{x}})\, \mathrm{d}t, \tag{1}$$

where the integrand $L$ in common base consists of the summands

$$L_{\text{base}} = j_{\text{v}_{\text{acc}}} + j_{\text{v}_{\text{jrk}}} + j_{\text{r}_{\text{vel}}} + j_{\text{r}_{\text{acc}}} + j_{\text{r}_{\text{jrk}}}. \tag{2}$$

The summands consists either of value residual terms $j_{\text{v}}$ or of range residual terms $j_{\text{r}}$. For a variable $p(t)$ of the variable parameter vector $\mathbf{p}(t)$ the value residual $j_{\text{v}_p}$ is given as

$$j_{\text{v}_p} = w_{\text{v}_p} \left| p(t)_{\text{desired}} - p(t)_{\text{current}} \right|^2,$$

and the range residual $j_{\text{r}_p}$ as

$$j_{\text{r}_p} = \begin{cases} w_{\text{r}_p} |p(t)_{\text{max}} - p(t)_{\text{current}}|^2 & \text{if } p(t) > p_{\text{max}}, \\ w_{\text{r}_p} |p(t)_{\text{min}} - p(t)_{\text{current}}|^2 & \text{if } p(t) < p_{\text{min}}, \\ 0 & \text{otherwise.} \end{cases}$$

The parameters $w_{\text{v}_p}$ and $w_{\text{r}_p}$ represent the weighting factor of the corresponding cost term in the multicriteria problem. It should be underlined that, during a stable and comfortable ride acceleration and jerk values are generally small and their range is limited. Hence, the summands presented in Equation (2) have desired acceleration and jerk values of 0, and their range is bounded to the physical limits of the vehicle.

In order to solve the objective functional defined in Equation (1) the trajectory must be discretized. The trajectory $\mathbf{x}(t)$ can be approximated by a number of sampling points that are equidistant in time

$$t_i = t_0 + ih, 0 \leq i < N,$$

where $h$ is the sampling interval and $N$ is the number of sampling points. The derivatives, which yield velocity, acceleration and jerk values can then be calculated with forward finite differences

$$\mathbf{x}_i^{\text{d}} := \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h} \approx \dot{\mathbf{x}}(t_i) \tag{3}$$

$$\mathbf{x}_i^{\text{dd}} := \frac{\mathbf{x}_{i+2} - 2\mathbf{x}_{i+1} + \mathbf{x}_i}{h^2} \approx \ddot{\mathbf{x}}(t_i) \tag{4}$$

$$\mathbf{x}_i^{\text{ddd}} := \frac{-\mathbf{x}_{i+3} + 3\mathbf{x}_{i+2} - 3\mathbf{x}_{i+1} + \mathbf{x}_i}{h^3} \approx \dddot{\mathbf{x}}(t_i) \tag{5}$$

If the forward differences presented in Equation (3)–Equation (5) are applied, the integral presented in Equation (1) can be approximated by the sum

$$J^{\text{d}}(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}) = \sum_{i=0}^{N-4} L^{\text{d}}(\mathbf{x}_i, \mathbf{x}_i^{\text{d}}, \mathbf{x}_i^{\text{dd}}, \mathbf{x}_i^{\text{ddd}}) \tag{6}$$

which can be solved with numerical methods. In accordance with the work of Ziegler *et al.* [28], which serves as a foundation to the formulations above, we refer the sampling points $\mathbf{x}_i$ as *trajectory support points*. These points are used for feed-forward and feed-back control of Bertha.

In order to minimize the nonlinear cost function defined in Equation (6), the optimization library Ceres is utilized [29], [30]. In Ceres, we apply vehicle motion limits in form of *soft constraints*. We use the trust region algorithm Levenberg-Marquardt to solve the minimization problem. For a planning horizon of $T = 10\,\text{s}$ and discretization of $100\,\text{ms}$ we get 200 parameters for optimization. Such a medium scale problem can highly benefit from sparsity and hence we use sparse Cholesky factorization to solve the linearized least squares problem.

The presented optimization routine is defined only for the trajectory planning of a single time step. The vehicle continuously re-runs these computations within a numerical computation time $t_{\text{comp}}$, which is typically 15 ms. For initializing the optimization problem of a later timestamp, we cut our last solution until the time being planned for and shift it to the estimated vehicle position. To maintain consistency between the last solution and the computed solution, we insert the last four trajectory points of the executed motion to the initialization. To fill the missing data at the end of the planning horizon, we use linear extrapolation. We then interpolate the trajectory to match the new timestamps. To ensure temporal planning consistency and stability we hold the first $3 + N_{\text{pin}}$ trajectory support points of the initialization constant. The pinned points $N_{\text{pin}}$ are being driven while the current trajectory is computed, and hence cover a slightly longer time than $t_{\text{comp}}$

$$(N_{\text{pin}} - 1)\, h < t_{\text{comp}} \leq N_{\text{pin}}\, h\,.$$

The first 3 points are due to the utilization of forward finite differences.

The formulation presented above, with minor modifications and several additional cost terms, yields very good results for automated driving. As presented in Section V-A, the GCDC has a synthetic nature in sense that the organization defines several rules to make the competition bearable and evaluable. The heats take place either on straight roads or on roads with very small curvature. The scoring is merely based on longitudinal performance and does not regard the automation level and the quality of lane changes. Therefore, we considered only the longitudinal component and tuned the planner to achieve the highest scoring in the benchmarking. In this way, we could reduce the number of optimization parameters by half and furthermore reduced the computational burden of vehicle motion constraints due to steering maneuvers in the online optimization. Hence, by basically making a path-velocity decomposition [31], harder tolerance values were applied as a termination criterion and the quality of the resulting speed profile was enhanced.

*2) Highway Planner:* For the Scenario 1, we differentiate between two planners that build on the base planner. If the ego vehicle is the platoon leader, we engage the *planner lead* and if the ego vehicle is somewhere else in the platoon and hence is following another vehicles, we engage the *planner follow multiple*.

*a) Planner lead:* In case the ego vehicle is the platoon leader, a further objective will be driving with the reference speed. Hence, we add an additional summand for the set speed value residual

$$L_{\text{lead}} = L_{\text{base}} + j_{v_{\text{vel}}}. \tag{7}$$

*b) Planner follow multiple:* This planner encapsulates all formations other than the one in which ego vehicle is the platoon leader. The main objective of this planner instance is to maintain a desired distance $x_{\text{desired}}$ to the leader vehicle while observing the predefined base criteria. Hence, we add the distance value summand to penalize deviations from it

$$L_{\text{multi}} = L_{\text{base}} + j_{v_{\text{dist}}}. \tag{8}$$

The desired distance $x_{\text{desired}}$ can be calculated as

$$x_{\text{desired}} = x_{\text{stand\_still}} + v\,\text{H}_{\text{desired}}$$

where $v$ is the vehicle speed, $x_{\text{stand\_still}}$ is the standstill distance, and $\text{H}_{\text{desired}}$ is the desired headway time, and are both predefined by the organization. For safety reasons, the organization set a further scoring function that is based on the inter-vehicle distance, namely the *penalty scoring function*. If a follower vehicle comes closer to its leader than 0.7 $x_{\text{desired}}$ its score points are subtracted. Hence, we applied a further range cost term that restricts undershoots greater than the threshold defined above. Once the lane change is initiated, the planner gradually switches its reference vehicle to the one on the other lane and completes the switch within the planning horizon $T$. Such a switch results in smooth transitions.

*3) Intersection Planner:* The intersection scenario has been presented in Section V-A.2. The scoring objectives of the intersection scenario fundamentally consider position and speed values and reward observation of their desired values. In order to establish the same foundation with the highway planner, described in Section III-E, we project the position of the OPC (cf. Figure 7) on our lane. In this way, we build the planner again on the base summand and define additional constraints to meet scenario specific requirements.

In order to plan the optimal trajectory we select a planning horizon that covers the time until maneuvering is completed.

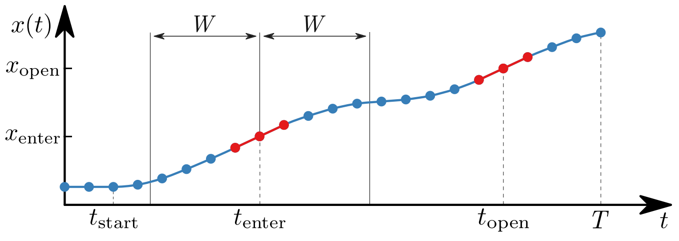


Fig. 8. Optimized trajectory illustrated on path-time diagram. Constraints are visualized with red bars.

We select a planning horizon $T = 50\,s$ and a sampling interval of $h = 500\,\text{ms}$.

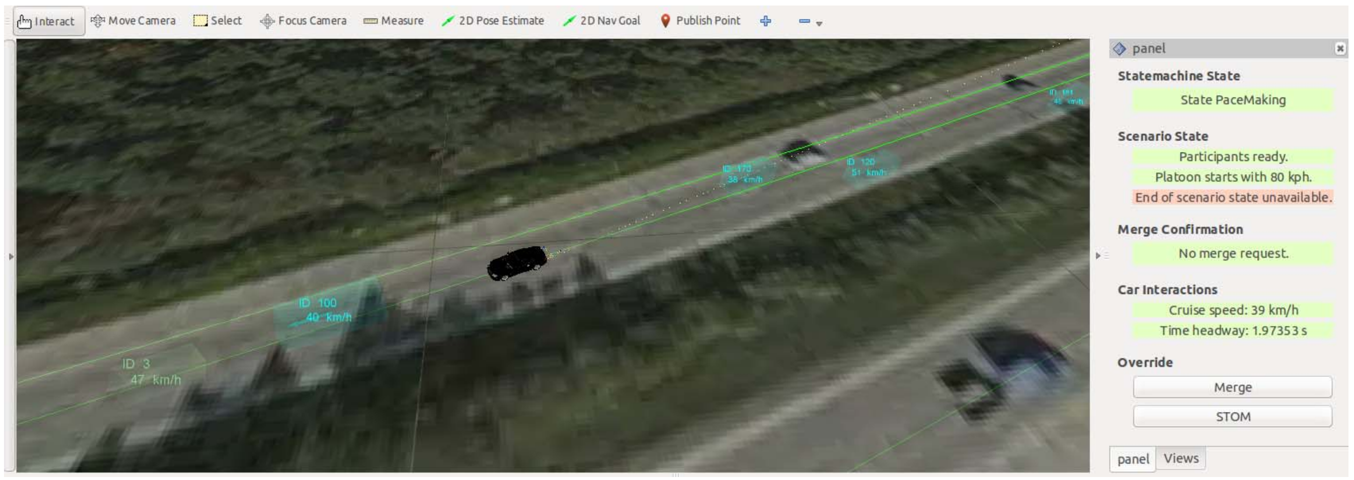
We apply two constraints to the planner. The first one ensures to enter the competition zone (CZ), $x_{\text{enter}}$, at a predfined time $t_{\text{enter}}$ with a speed of 30 km/h. When the vehicles are inside the CZ, they start transmitting maneuver intentions. Once we receive the maneuver intention of the OPC, we calculate the time $t_{\text{open}}$ at which the driving corridor $x_{\text{open}}$ is not blocked anymore. We then apply the constraint to drive on $x_{\text{open}}$ at $t_{\text{open}}$ with a speed of 30 km/h.

On path-time diagram (cf. Figure 8) the desired values of position correspond to points and the speed to the slope of those successive points. We first pin trajectory support points according to desired position and speed, then let the remaining support points be optimized by the planner.

Entering and leaving the CZ is only one of the criteria of judging. If these are hard constrained, other criteria that contribute to the final score, such as holding the desired distance, or the speed limit, can deviate more from their desired values. Therefore, in order to allow slight deviations and guide the solver to the optimal solution, we consider these in form of soft constraints and develop an activation and deactivation scheme as presented in Algorithm 1. For the sake of brevity, we will derive constraint activation for a single constraint event at $t_{\text{event}}$, the time at which a specific constraint must be active. The activation can either be at $t_{\text{enter}}$ or at $t_{\text{open}}$.

**Algorithm 1** Constraint activation and deactivation

```
Input:
    $W,\ t_{\text{event}},\ t_{\text{current}},\ x_{\text{event}},\ x_{\text{current}},\ v_{\text{desired}},\ v_{\text{current}}$
    $w_x,\ w_v$
Output:
    $j_{\text{error}_x},\ j_{\text{error}_v}$
 1: $v_{\text{error}} \leftarrow 0$
 2: $x_{\text{error}} \leftarrow 0$
 3: $\Delta t \leftarrow t_{\text{current}} - t_{\text{event}}$
 4: if $(|\Delta t| < W)$ then
 5:     $w_{\text{error}} \leftarrow 1.0 - \frac{|\Delta t|}{W}$
 6:     $x_{\text{desired}} \leftarrow x_{\text{event}} + v_{\text{desired}}\,\Delta t$
 7:     $x_{\text{error}} \leftarrow x_{\text{desired}} - x_{\text{current}}$
 8:     $v_{\text{error}} \leftarrow v_{\text{desired}} - v_{\text{current}}$
 9: else
10:     $w_{\text{error}} \leftarrow 0.0$
11: end if
12: $j_{\text{error}_x} \leftarrow w_x\,|w_{\text{error}}\,x_{\text{error}}|^2$
13: $j_{\text{error}_v} \leftarrow w_v\,|w_{\text{error}}\,v_{\text{error}}|^2$
```



panel

Statemachine State
State PaceMaking

Scenario State
Participants ready.
Platoon starts with 80 kph.
End of scenario state unavailable.

Merge Confirmation
No merge request.

Car Interactions
Cruise speed: 39 km/h
Time headway: 1.97353 s

Override
Merge
STOM

Fig. 9. Visualization of the vehicle positions, velocities and the scenario state of other participants. The panel to the right displays spatial information of participants, the active state of the ego vehicle's state machine, the state of the scenario and also buttons to override automated actions.

During planning, we continuously check the current time $t_{\text{current}}$ and compare it with event time $t_{\text{event}}$. We define a time window of duration $W$ for the activation or deactivation of the constraint. Once the current time $t_{\text{current}}$ is in the $W$ vicinity of $t_{\text{event}}$ we start the activation and once it passes the event time $t_{\text{event}}$ we deactivate the constraint (cf. Figure 8). We perform this computation for every single trajectory support point, i.e. for $N$ points.

Depending on the lane in which the ego vehicle starts, we identify two distinct cases.

*a) Planner crossing blocked:* In this case, the driving corridor will be blocked by the OPC until it completes its turn maneuver. Once the crossing is open, the planner is essentially switched to Planner Lead, presented in Equation (7).

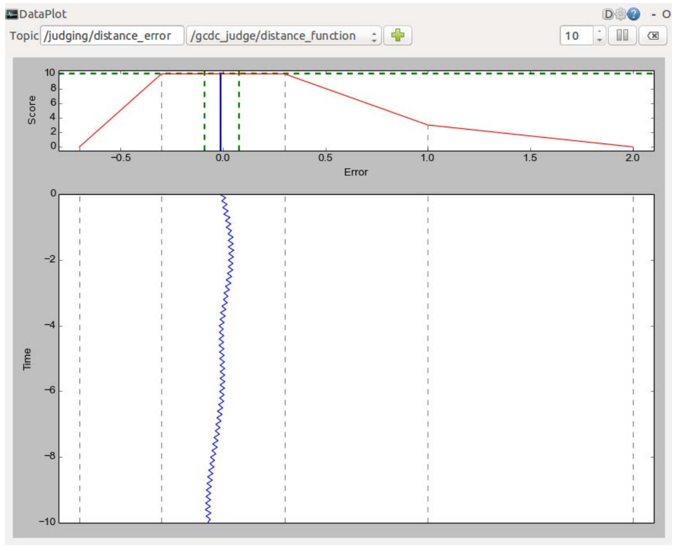
*b) Planner crossing follow:* In case we approach the intersection from Lane 3 in Figure 7, we use the same approach as the Planner Crossing Blocked, introduced previously. The main difference is that, once the OPC is physically on our lane, we use the follow behavior and utilize the formulations for Planner Follow Multiple, presented in Equation (8).

## VI. Human-Machine Interface

As part of the benchmarking, our goal in Human Machine Interface (HMI) design is to give users detailed information about the current vehicle state, the processing pipeline configuration and to let them trigger actions in case of unexpected situations.

To show relevant environmental information, we use the 3D visualization environment RViz that comes with ROS. Figure 9 on the left-hand side depicts our visualization during one of the highway scenarios. We display the position and extent of non-moving obstacles, motion state of moving vehicles, desired vehicle interactions based on the communication protocol and the desired ego vehicle trajectory.[2] Furthermore, we add satellite maps to help the user monitor the current geo-referenced position. In a separate panel, we display the state machine state, merge confirmations, car interactions and the scenario state which indicates whether a heat has started or is

[2]http://url.fzi.de/gcdc_hmi



Topic /judging/distance_error /gcdc_judge/distance_function 10

Fig. 10. Scoring plots developed for visualizing individual components of the total score obtained during a heat. The lower part of the figure shows the time history of the error, and the upper part the score associated to the error. The red line in the upper part corresponds to the scoring profile for the desired distance defined in the competition rules. The vertical dashed green lines signalize the upper and lower bound of obtained the score and error, both since the start of the heat.

already finished. This panel, as depicted on the right-hand side in Figure 9, is implemented in a separate process and keeps running if any system component fails. The panel is directly connected to our diagnostic system and capable to visualize different diagnostic levels `ok`, `warn`, `error` with different colors. For testing purposes we also implemented override functions for scenario flags such as merge confirmations and lane change initiations.

Within ROS, we employ a parameter server that allows us to tune all the parameters during runtime using a graphical user interface (GUI). We also implemented a GUI in which we visualize individual components of the total score (cf. Figure 10). By means of the parameter server GUI and the score plot we can continuously monitor the obtained score and tune the optimization parameters when the heat is over.

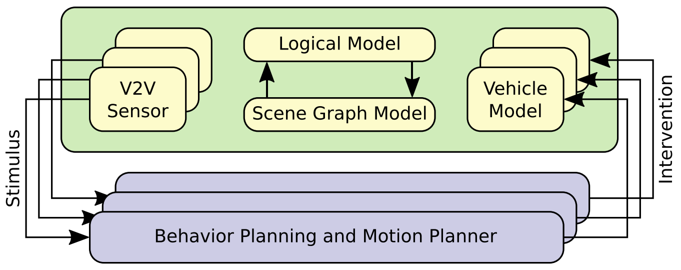


Fig. 11. Concept of the virtual validation framework: By n-fold instantiation, the internal consistency of the cooperative capabilities can be tested and validated.

## VII. Simulation Framework

### A. Concept

The development and testing of cooperative driving maneuvers offer new challenges to a virtual development and testing framework compared to present, self-centered advanced driver assistance systems. In the case of the GCDC, we test the cooperative behavior and the motion planning capabilities of Bertha (cf. Figure 3). We call these components as *system under test*. In the following, we describe the simulation framework, which enabled us to reproduce different traffic scenarios with the system under test in a safe and cost-efficient manner.

For simulation we adapt the framework proposed in [32]. The virtual world is administrated within a *scene graph structure*, which holds all attributes and properties of distinct entities in the virtual world. Dynamic situation aspects, such as the behavior of other traffic participants, are mapped by a *logical model*. The *vehicle mechanics* and the *vehicle sensor model* provide interfaces to integrate different system under tests. These models have been adapted in order to embed the cooperative capabilities as a system under test (cf. Figure 11).

The agents, which represent cooperative participants, are instantiable multiple times. They are equipped with a behavior model and a vehicle mechanics model. We use the *Intelligent Driver Model* (IDM) as an initial behavior model [33]. The IDM provides an acceleration controller strategy for collision-avoidance on a reference lane. For imitating vehicle mechanics, we apply the acceleration with a delay. Every agent also contains its own instantiation of the system under test, which then can be activated during simulation in order to oversteer the IDM-based behavior. In this way, a set of agents can be brought into a defined configuration before testing the system under test.

An abstract sensor is instantiated for every agent, imitating an ideal V2V communication. In simulation we focus on the testing of the high level planning components and neglect effects of delayed messages or packet losses. The transmitted CAM messages of every agent are first aggregated within a message pool and then distributed to every agent's sensor model.

### B. Integration Into the ROS Framework

As described in Sec. II-B, different processing algorithms are integrated in separate ROS nodes. Therefore, every agent in our framework is implemented as a Gazebo [34] plugin. The plugin simulates the vehicle interfaces virtually by mimicking all low level messages from sensors and actuators. This way, ROS messages from the virtual V2V sensors are fed into Bertha's planning algorithm and its output is redirected back to the simulation. With this approach, the same binary nodes can be tested seamlessly in simulation as on the real vehicle.

By using the concept of namespaces, multiple instances of agents' nodes as well as of the system under test are connectable without interfering each other. Thereby, the behavior and subsequent motion planning is instantiated for every agent which facilitates the testing against itself and thereby expose internal consistency failures of the state machine, for example.

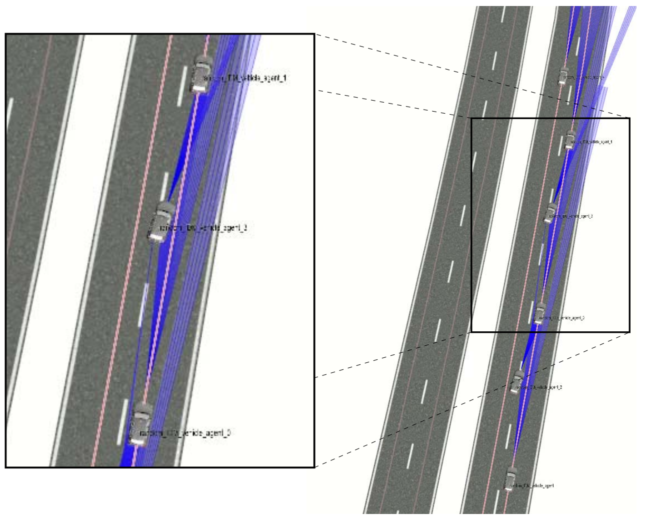


Fig. 12. Highway scenario in simulation: The vehicles are controlled by separate instances of the system under test.

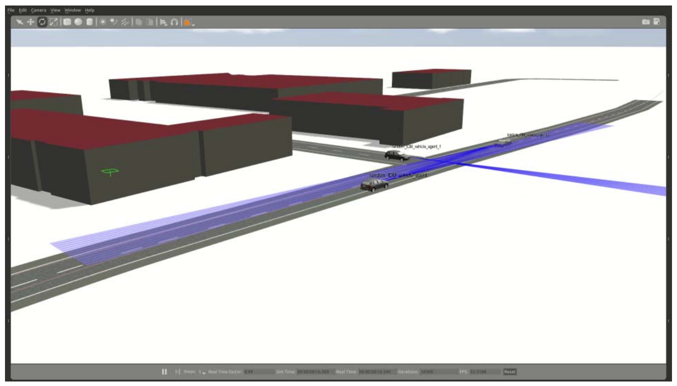


Fig. 13. Intersection scenario in simulation: Three agents approaching the intersection.

### C. Testing in the Loop

Testing in a non-intrusive and seamless way using ROS nodes allows us to use all debugging and introspection tools both on the real and the virtual vehicle. For the reproduction of scenarios, we modeled simulation scenarios of Highway Merging (cf. Figure 12), and the Cooperative Intersection (cf. Figure 13 or the uploaded video).[3]

[3]http://url.fzi.de/gcdc_validation

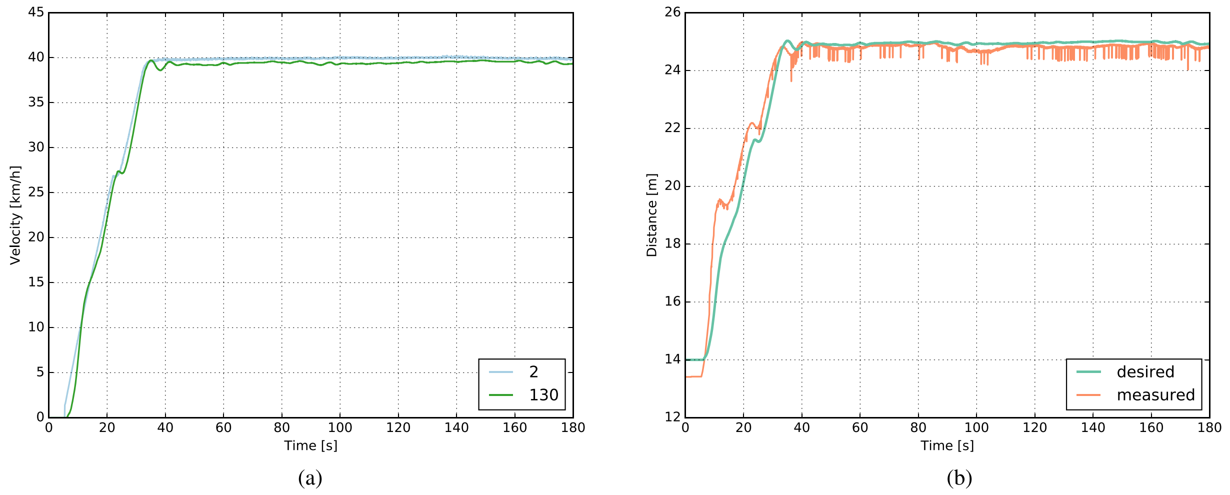


Fig. 14. The first 180 seconds of the Scenario 1 heat started at 09:06. Bertha follows the vehicle with the station ID 2. (a) The speed profile of the leader vehicle and Bertha. (b) Distance between the leader vehicle and Bertha.

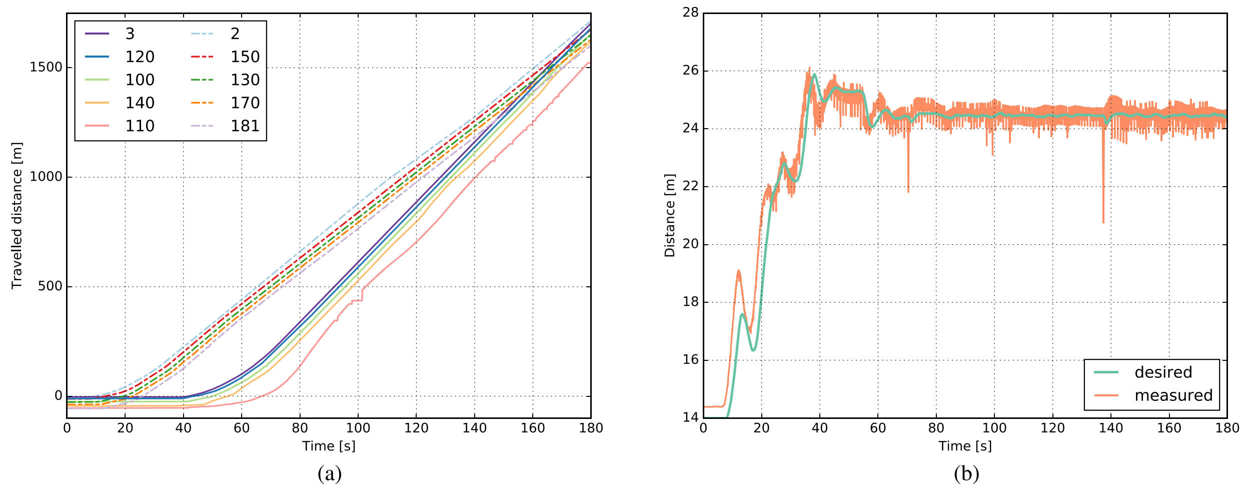


Fig. 15. The first 180 seconds of the Scenario 1 heat that was started at 10:41. (a) The travelled distance of all participants. Vehicles on Lane A are depicted with continuous lines whereas the ones on Lane B are depicted with dashed lines. Positions of some participants are subject to jumps. (b) The desired and the measured distances to the leader vehicle of Bertha. Notice the jitter in the measured values. This occurs due to positioning imperfections of the transmitting vehicle.

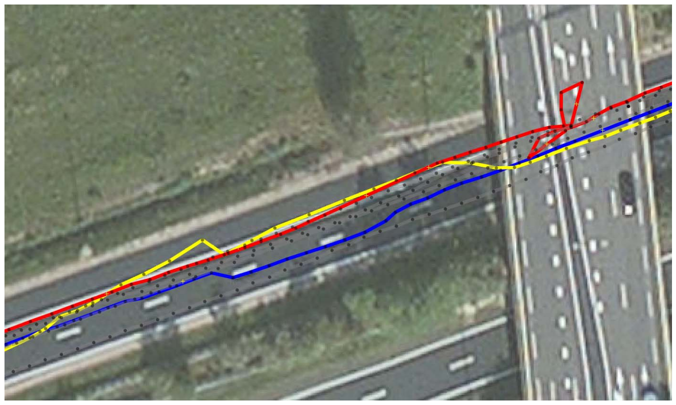


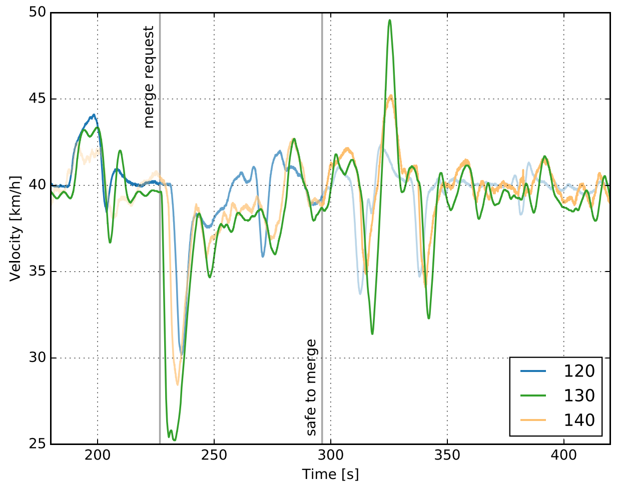
Fig. 16. Vehicles subject to GNSS drift after passing under a bridge. The aerial imagery is taken from Bing.

Such a simulation scenario consists of the static environment, such as the road layout and buildings, the configuration of the participating agents' starting positions and their behavior settings, such as the IDM's parameters, as well as their namespace configurations. Once we instantiate the virtual world, we start the ROS nodes and attach them to multiple agents in the simulation. Afterwards, we run the simulation either for a given time period, or until a failure condition is detected.

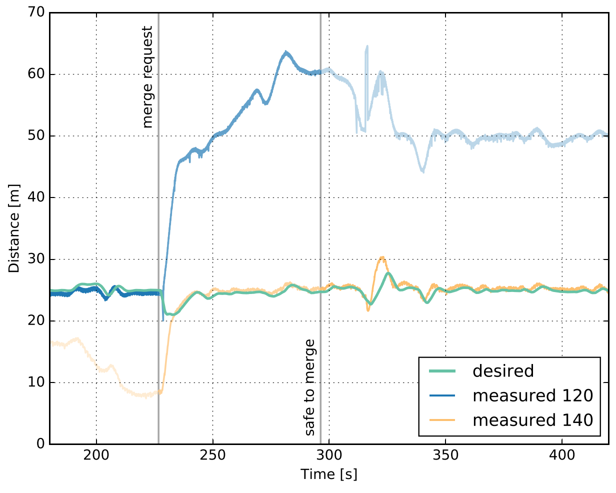
The simulation framework brought many benefits. Focusing on the testing of the high level planning modules allowed a very fast development cycle, where new features in Bertha's code base were tested and validated rapidly. The framework saved the team from a large category of failures before encountering them in real world. The simulation's capability to create, modify and reload traffic scenarios also helped to investigate failure states. The disadvantage of the proposed approach is that the system under test is only exposed to a subset of failures of the software. In particular, it is limited to the detection of internal consistency failures of the implementation.

## VIII. Benchmarking and Evaluation

The GCDC is based on cooperation and the performance of individual teams depends highly on the performance of

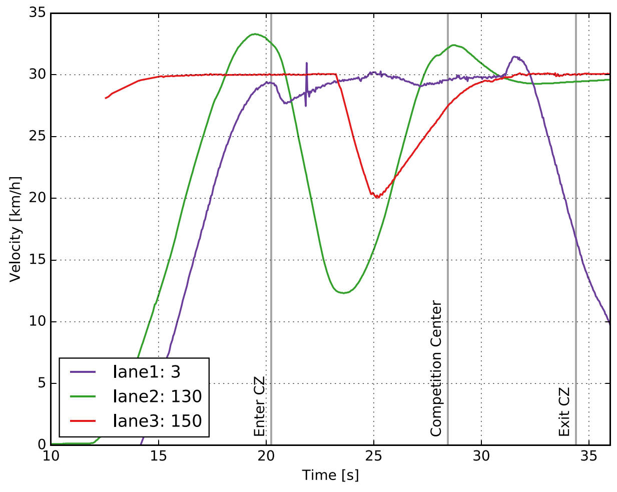


(a)

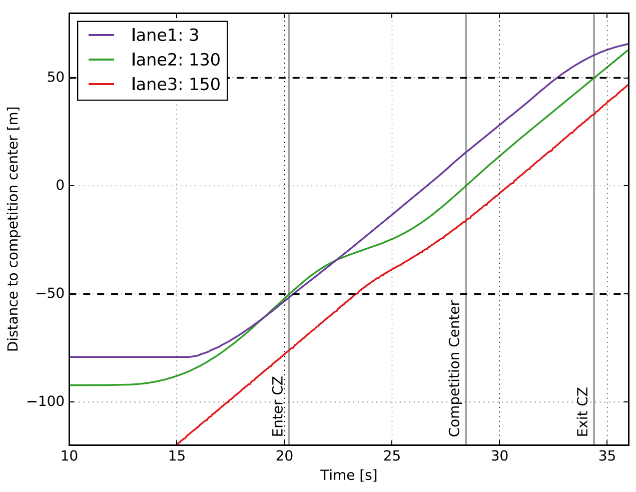


(b)

Fig. 17. The Scenario 1 heat that was started at 14:29. The figure focuses on the lane change maneuver of the heat. The vertical lines indicate the time when the lane change is initiated and is performed. Before the lane change is initiated the reference vehicle of Bertha is the one with ID 120 and afterwards the one with the ID 140. (a) Speed profiles of the vehicles. (b) Desired distances relative to Bertha.



(a)



(b)

Fig. 18. The Scenario 2 heat that was started at 13:51. The vertical lines indicate the times at which Bertha entered the competition zone, was on the competition center, and left it. (a) Speed profile of Bertha together with the received speed profile of other participants. (b) Relative distance of the participants to the competition center.

other platoon members. In order to reduce the effect of group performance and to have a fair benchmarking, a number of heats were performed on the competition day, on 29.05.2016. In the following we will evaluate the performance of Bertha, the station ID 130, based on the intervehicular communication logs. We provide velocity profiles, path-time diagrams, and the measured distance together with its desired values.

Figure 14 shows the first 180 seconds of a Scenario 1 heat. During the selected time frame Bertha adjusts its speed while keeping its relative speed and distance to the leader vehicle. Figure 14a shows a very good speed profile with almost no overshoot and a negligible delay. The distance in Figure 14b is, however, subject to larger deviations from its reference. This is because the GCDC rules have tolerance in position values (cf. Figure 10). Nevertheless, position errors remain less than 2 meters.

Figure 15 shows the first 180 seconds of another Scenario 1 heat. On Figure 15a the path-time diagram of all participants is shown. Most of the vehicles in both platoons exhibit very good follow behavior. Some of the vehicles are subject to jumps in their transmitted positions. Figure 15b displays the desired and the measured distance of Bertha to its leader. The jitters in the figure highlight positioning imperfections of the transmitting vehicle.

The transmitted positions of vehicles play an important role in cooperative driving. Most of the teams rely on GNSS-based localization solutions which are subject to inaccuracies due to multipath-propagation and GNSS-signal clutters. The first scenario of GCDC was performed on a highway which is on an open area and hence most of the time, the participants were not subject to such inaccuracies. However, as depicted in Figure 16, while participants were driving under bridges GNSS drifts have occurred.

An analysis of an entire Scenario 1 heat is shown in Figure 17. Bertha follows the vehicle with the station ID 120 before the lane change is initiated. Once the merge request is received, because of the summands of the mutlicriteria problem, it gently starts following the vehicle with the station ID 140 even though the reference vehicle switch occurs instantenously. The speed profile and the desired distance relative to Bertha are given in Figure 17a and Figure 17b, respectively. It should be underlined that the transmitted positions of the vehicles are the positions of the vehicle rear axle center. In all figures, if the bumper-to-bumper distances would be plotted, a vehicle length must be subtracted from these values.

Figure 18 shows the speed profiles and the travelled distance of an intersection scenario.[4] While Bertha was able to receive the intervehicular communication messages of the vehicle with ID 3 during the entire heat, it could only receive messages from station ID 150 for a small time interval because of the distance between the vehicles. The speed profile of the station ID 3 is subject to small jitters (cf. Figure 18a). However, this does not influence the quality of planned motion and Bertha exhibits a very smooth motion. As indicated on Figure 18b, the Bertha enters the CZ slightly after 20 seconds. Once it receives the maneuver intention, it gently brakes to allow a safe passage and then accelerates back to 30 km/h. The speed profile exhibits a small overshoot while entering and exiting the CZ. This is because the slightly delayed entering into CZ, and the unmodeled dynamics of the vehicle.

## IX. Conclusion and Future Work

The GCDC gave again an important impetus to the real-life implementation of cooperative automated driving. Researchers from different institutions could cooperate their systems by means of intervehicular communication. Compared to the first GCDC, the key development was the interaction protocols and its additional message sets defined for cooperative maneuvering. By means of this, safe and efficient merging and intersection crossing were made possible.

We, Team AnnieWAY participated in the challenge with the ultimate goal of defending our title which we received in the 2011 GCDC. Although, we took the second place this time, we have gained substantial experience in the research field and used the opportunity to robustify and advance our algorithms. The 2016 GCDC was the first competition we took part in with Bertha. Within the past two years we have replaced our existing software framework with ROS. In spite of the fact that we had to rewrite many of our software modules, the modular structure of ROS allowed us to perform short term modifications and provided perfect tools that facilitated the development of our algorithms.

Turning off our perception system greatly simplified our system and eliminated failures that might have arisen due to software bugs and ghost objects. This sacrifice of redundancy against complexity, however, made us vulnerable to localization imperfections of other participants. Even though the GCDC heats were performed on open areas, that are not subject to multipath propagation and satellite occlusion, and the organizers provided GPS correction signals directly on the track as well, localization accuracy has been a very significant problem.

Our motion planner utilized the same fundamental methodology as the BBMR-mission. Problem formulation and solution approach were however significantly different. Multi-criteria optimization based motion planner played a dominant role in getting the excellent final score. Our enhanced virtual validation framework rescued us facing failures of our system, especially of the statemachine. We tested the heats in the framework and tuned the optimization parameters.

Our main deficiency, where we missed to get a high score, was the HMI. We were able to visualize perception, intervehicular communication information, and their fusion results and override the actions of the automated vehicle, if required. However, it failed to be user friendly for non-developers resulting in low score points.

The heats of the GCDC have shown that the main problem in cooperative driving is robustness. Because of the nature of cooperation, the individual vehicles are too interdependent and a single point of failure inside a platoon usually ends up with the failure of the cooperation among the platoon. In order to bring cooperative driving into real life, future research must focus on fall-back methodologies. We believe that sensor-based perception will play a major role in the detection and execution of those fall-back methodologies.

## Acknowledgement

The presented automated vehicle has been developed by a group of researchers at our institute. The authors like to express their gratitude to other members in the research group who did not actively work on the GCDC 2016 but made significant contributions to the development of the vehicle, in particular to Johannes Beck, Philipp Bender, André-Marcel Hellmund, and Jens Doll.

The authors thank the partners of the i-GAME consortium for excellent organization of the GCDC 2016.

## References

[1] K. Bengler, K. Dietmayer, B. Färber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, Oct. 2014.

[2] A. De La Fortelle *et al.*, "Network of automated vehicles: The AutoNet 2030 vision," in *Proc. ITS World Congr.*, 2014, pp. 1–10.

[3] O. Sawade and I. Radusch, "Survey and classification of cooperative automated driver assistance systems," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2015, pp. 1–5.

[4] D. Bevly *et al.*, "Lane change and merge maneuvers for connected and automated vehicles: A survey," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 105–120, Mar. 2016.

[5] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[6] E. V. Nunen, M. R. J. A. E. Kwakkernaat, J. Ploeg, and B. D. Netten, "Cooperative competition for future mobility," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1018–1025, Sep. 2012.

[7] C. Englund *et al.*, "The grand cooperative driving challenge 2016: Boosting the introduction of cooperative automated vehicles," *IEEE Wireless. Commun.*, vol. 23, no. 4, pp. 146–152, Aug. 2016.

[4]http://url.fzi.de/gcdc_intersect

[8] U. Ozguner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA grand challenge experience," *Proc. IEEE*, vol. 95, no. 2, pp. 397–412, Feb. 2007.

[9] S. Kammel *et al.*, "Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 615–639, 2008.

[10] A. Geiger *et al.*, "Team AnnieWAY's entry to the 2011 grand cooperative driving challenge," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1008–1017, Sep. 2012.

[11] J. Ziegler *et al.*, "Making Bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Oct. 2014.

[12] Ö. Ş. Taş, F. Kuhnt, J. M. Zöllner, and C. Stiller, "Functional system architectures towards fully automated driving," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2016, pp. 304–309.

[13] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009, p. 5.

[14] A.-M. Hellmund, S. Wirges, Ö. Ş. Taş, C. Bandera, and N. O. Salscheider, "Robot operating system: A modular software framework for automated driving," in *Proc. IEEE Intell. Trans. Syst. Conf.*, Nov. 2016, pp. 1564–1570.

[15] T. Strauß, J. Ziegler, and J. Beck, "Calibrating multiple cameras with non-overlapping views using coded checkerboard targets," in *Proc. IEEE Intell. Trans. Syst. Conf.*, Oct. 2014, pp. 2623–2628.

[16] J. Gräter, T. Schwarze, and M. Lauer, "Robust scale estimation for monocular visual odometry using structure from motion and vanishing points," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2015, pp. 475–480.

[17] M. Sons, H. Lategahn, C. G. Keller, and C. Stiller, "Multi trajectory pose adjustment for life-long mapping," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2015, pp. 901–906.

[18] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2010, pp. 217–224.

[19] F. Poggenhans, M. Schreiber, and C. Stiller, "A universal approach to detect and classify road surface markings," in *Proc. IEEE Intell. Trans. Syst. Conf.*, Sep. 2015, pp. 1915–1921.

[20] M. Weber, P. Wolf, and J. M. Zöllner, "DeepTLR: A single deep convolutional network for detection and classification of traffic lights," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2016, pp. 342–348.

[21] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

[22] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

[23] M. Schreiber, A.-M. Hellmund, and C. Stiller, "Multi-drive feature association for automated map generation using low-cost sensor data," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2015, pp. 1140–1147.

[24] M. Schreiber, H. Königshof, A.-M. Hellmund, and C. Stiller, "Vehicle localization with tightly coupled GNSS and visual odometry," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2016, pp. 858–863.

[25] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1246–1257, Jun. 2014.

[26] M. Schreiber, C. Knöppel, and U. Franke, "LaneLoc: Lane marking based localization using highly accurate maps," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2013, pp. 449–454.

[27] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 987–993.

[28] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha—A local, continuous method," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2014, pp. 450–457.

[29] P. Bender, Ö. Ş. Taş, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2015, pp. 1386–1392.

[30] S. Agarwal and K. Mierle. *Ceres Solver*. Accessed Feb. 1, 2016. [Online]. Available: http://ceres-solver.org

[31] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986.

[32] M. R. Zofka, S. Klemm, F. Kuhnt, T. Schamm, and J. M. Zöllner, "Testing and validating high level components for automated driving: Simulation framework for traffic scenarios," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2016, pp. 144–150.

[33] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, pp. 1805–1824, Aug. 2000.

[34] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep./Oct. 2004, pp. 2149–2154.

**Ömer Şahin Taş** received the B.Sc. degree in mechanical engineering from Istanbul Technical University in 2011, and the M.Sc. degree in mechanical engineering from the Karlsruhe Institute of Technology (KIT) in 2014. He is currently pursuing the Ph.D. degree with the Institute of Measurement and Control Systems, KIT. He founded the Turkey's first Formula SAE/Student Team and also participated in the Grand Cooperative Driving Challenge 2011 as a member of the Team Maker. He is currently a Research Scientist with the Mobile Perception Systems Department, FZI Research Center for Information Technology.

His research interests include automated driving in general and motion planning under uncertainty in particular.

**Niels Ole Salscheider** received the M.Sc. degree in electrical engineering from RWTH Aachen University in 2014. He is currently pursuing the Ph.D. degree with the Institute of Measurement and Control Systems, Karlsruhe Institute of Technology. He is currently a Research Scientist with the Mobile Perception Systems Department, FZI Research Center for Information Technology.

His research interests include machine learning and vision-based environment perception for automated driving.

**Fabian Poggenhans** received the M.Sc. degree in mechanical engineering from the Karlsruhe Institute of Technology in 2014, where he is currently pursuing the Ph.D. degree with the Institute of Measurement and Control Systems. He is currently a Research Scientist with the Mobile Perception Systems Department, FZI Research Center for Information Technology.

His research interests include road surface marking detection and classification, and vision-based localization.

**Sascha Wirges** received the M.Sc. degree in electrical engineering from the Karlsruhe Institute of Technology in 2015, where he is currently pursuing the Ph.D. degree with the Institute of Measurement and Control Systems. He is currently a Research Scientist with the Mobile Perception Systems Department, FZI Research Center for Information Technology.

His research interests include laser-based environment perception and online localization and mapping.

**Claudio Bandera** received the M.Sc. degree in mechanical engineering from KIT in 2015. He led a students' group with the Karlsruhe Institute of Technology, developing autonomous model vehicles for an international competition. From 2015 to 2016, he was a Research Scientist with the Mobile Perception Systems Department, FZI Research Center for Information Technology.

**Marc René Zofka** received the B.Sc. degree in information engineering from the University of Konstanz in 2010, and the M.Sc. degree in computer science from the Karlsruhe Institute of Technology in 2012. He is currently a Research Scientist with the Technical Cognitive Systems Department, FZI Research Center for Information Technology.

His research interests include the validation and testing of highly automated driving functions, especially in virtual environments.

**Tobias Strauss** received the Dipl.-Ing. degree in mechanical engineering and the Dipl.-Ing. degree in electrical engineering from the Karlsruhe Institute of Technology (KIT) in 2009, and the Ph.D. degree, with a focus on multi camera calibration, in 2015. From 2015 to 2016, he was a Group Leader with the Institute of Measurement and Control Systems, KIT, and the captain of the Team AnnieWAY. In 2016, he joined Bosch Corporate Research.

His research interests are camera calibration and automated vehicle perception systems.

**J. Marius Zöllner** received the degree in computer science, with special focus on artificial intelligence and robotics, from the Karlsruhe Institute of Technology (KIT), and the Ph.D. degree from KIT in 2005. Since 1999, he has been with the FZI Research Center for Information Technology, where he became the Division Manager in 2006. Since 2008, he has been a Professor of applied technical cognitive systems with KIT and the Director with FZI. Since 2012, he has been a member of the Executive Board of FZI. His main research interests include the perception and interpretation of the driving environment, probabilistic situation understanding, behavior decision, cognitive cars, service robotics, and machine learning.

**Christoph Stiller** received Diploma degree in electrical engineering, in Aachen, Germany, and Trondheim, Norway, and the Ph.D. from RWTH Aachen University in 1994. He held a postdoctoral position with INRS, Montreal, Canada. In 1995, he joined the Corporate Research and Advanced Development, Robert Bosch GmbH, Hildesheim, Germany. In 2001, he became the Chaired Professor with Karlsruhe Institute of Technology, Germany. In 2010, he spent three months by invitation at CSIRO in Brisbane, Australia. In 2015, he spent a four month sabbatical with Bosch RTC and Stanford University in California. He served as the President of the IEEE Intelligent Transportation Systems Society from 2012 to 2013 and the Vice-President before since 2006. He served as the Editor-in-Chief of the *IEEE Intelligent Transportation Systems Magazine* from 2009 to 2011, and as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1999 to 2003. He has been an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS since 2004 and the *IEEE Intelligent Transportation Systems Magazine* since 2012.

He is a Co-Founder of Atlatec UG, a startup company offering visual localization products for automotive applications. His Autonomous Vehicle AnnieWAY has been Finalist in the Urban Challenge 2007, USA. He received the Grand Cooperative Driving Challenge 2011, The Netherlands. Recently, he collaborated with Daimler on the automated Bertha–Benz Memorial Tour in 2013.