An integrated simulation framework for cognitive automobiles

Stefan Vacek, Robert Nagel, Thomas Batz, Frank Moosmann and Rüdiger Dillmann

– intended for Special Session ”Cooperative Cognitive Automobiles” –

Abstract— **A cognitive automobile is a complex system. It is an indisputable fact that simulations are valuable tools for the development and testing of such complex systems. This paper presents an integrated closed-loop simulation framework which supports the development of a cognitive automobile. The framework aims at simulating complex traffic scenes in inner-city environments. The key features of the simulation are the generation of synthetic data for high level inference mechanisms, providing data for the analysis of car-to-car communication strategies and evaluation of cooperative vehicle behavior.**

I. INTRODUCTION

The development of software, especially of mission critical software as in the case of cognitive automobiles, requires a simulation environment. This is the only possibility for testing the software before real use.

Having a simulation environment, it is possible to test dangerous situations and to start with a precisely described environment, to change the environmental parameters such as weather, quality of measurement methods and – after changing the algorithms or their parameters – to restart the simulation process with exactly the same conditions. Disturbances or failures of sensors and of the communication can be evaluated in a controlled manner. Furthermore, a simulation saves a considerable effort for testing interaction and cooperation strategies for multiple vehicles.

In the field of simulation of vehicles and traffic there are many diverse publications. Most simulations concentrate on certain aspects such as dynamics [1], [2], microscopic [3] or macroscopic traffic models [4]. Possible applications vary from generation of realistic images of traffic scenes [5] to testing driver assistance systems [6]. In the latter case, a driving simulation is often incorporated into the simulation environment in order to study the implications of future driver assistance systems for the driver [1].

In this work the main emphasis lies on the generation of higher level synthetic data, evaluation of car-to-car communication strategies, and analysis of cooperative vehicle behavior. An important aspect is the seamless integration of the simulation into the real target system which allows the replacement of components of the vehicle with the simulation.

In order to understand how the simulation can support the development of a cognitive vehicle, a short overview of the project ”Cognitive Automobiles” is given in the next section. After that, the integrated simulation environment is presented, followed by the traffic simulation. Thereafter, existing applications are described and some results are shown.

II. PROJECT DESCRIPTION

The research centre ”Cognitive Automobiles” aims at developing an intelligent vehicle which is able to act safely and robustly in complex traffic scenes such as inner-city environments. The core components of the vehicle are the perception of its environment using complementary sensors, an interpretation of the perceived information in order to deduce an understanding of the current situation, a decision making process providing constraints for path planning and a component for path planning and control guiding the vehicle. Furthermore, the vehicle is equipped with communication capabilities to enable the interaction of multiple vehicles, both in terms of cooperative perception as well as cooperative driving. As can be seen in figure 1, the core components of the vehicle are linked with each other through a real-time database. This database serves as a system-wide data storage which holds all information from all components.

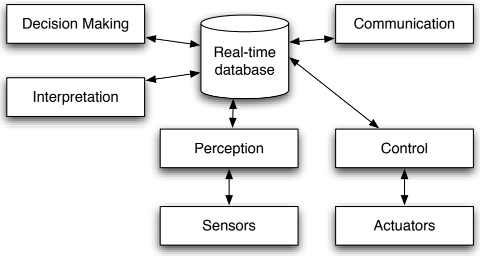

Fig. 1. Structure of the cognitive vehicle.

The perception has to detect lanes, intersections, vehicles and traffic blocks. Different sensors will be used such as cameras, laser, radar and time-of-flight cameras. Data fusion methods are required for handling spatially and temporally varying data. Additional fusion will take place at a symbolic level such as combining lane and vehicle detection in a common approach. Another aspect is the fusion of perceived data from different vehicles. As a result, the perception

S. Vacek and R. Dillmann are with Insitute for Computer Science and Engineering, University of Karlsruhe, D-76128 Karlsruhe, Germany {vacek|dillmann}@ira.uka.de

R. Nagel is with the Institute of Communication Networks, Munich University of Technology, D-80333 München, Germany robert.nagel@tum.de

T. Batz is with the Fraunhofer IITB Institute for Information and Data Processing, D-76131 Karlsruhe, Germany Thomas.Batz@iitb.fraunhofer.de

F. Moosmann is with the Institute of Measurement and Control, University of Karlsruhe, D-76128 Karlsruhe, Germany moosmann@mrt.uka.de

delivers object descriptions which will then be used in the higher level components.

The first high-level component is the interpretation of the perceived environment in order to get an understanding of the situation and to derive relevant information for decision making. The interpretation takes into account prior knowledge such as traffic rules and previously learned situations. The interpreted situation consists of a classification of the scene, critical objects within this scene and conflicting areas. It is used in the decision making process which determines the correct behavior for the current situation. The decision making uses a behavior network to calculate the resulting driving corridor which is used for control. This resulting corridor consists of an area allowed for driving as well as speed hints for control. Path planning is carried out based on this corridor and the vehicle is driven along this path.

The communication connects the real-time databases of several, equally equipped vehicles and thus allows a single cooperative vehicle to exchange data with other cooperative vehicles. This enables vehicles to expand their perceptional abilities by augmenting information about their local environment with data from other vehicles. Thus situation interpretation and decision making can be based on information much richer than that of a single vehicle.

Therefore a great potential in cognitive automobiles lies in the interaction of the different vehicles involved within the traffic scene. Special situations such as unexpected oncoming traffic or unexpected traffic blocks can only be solved adequately by changing the behavior of all participated road users. Using the mentioned communication channel, cognitive vehicles can affiliate as a Cooperative Group (CG). This group builds a unit for common decision making and also takes over the responsibility for the decision making of each vehicle in the CG.

III. INTEGRATED SIMULATION ENVIRONMENT

A. *Purpose of the simulation environment*

The main purpose of the simulation is to facilitate quick development of system components, to allow testing of components under controlled conditions and to compare different approaches without interfering with real vehicles. For some applications, testing in real life can only be realized with a huge effort, for the case of cooperative vehicles – especially dealing with cooperative driving in dangerous situations – it might even be impossible if there is a great risk of hurting other people.

Prototyping and testing components relies heavily on the ability to construct and replay scenarios that can not be easily realized in the real-world. One objective is the ability to exactly reproduce different scenarios. This helps in understanding why the system behaved in a certain manner and including the analysis of the scenario data the components involved. The main objective of our simulation environment is to be able to substitute the real world by a simulated environment. This substitution can be divided into two areas.

The first area is to serve as a basis for other simulations such as dynamic simulations for testing control strategies or communication simulations for investigating routing strategies. For instance, it is vital for the simulation of communication systems to precisely model both the mobility of nodes as well as the channel characteristics. For the movement of nodes, a suitable mobility model that fits the requirement of a certain application has to be selected. Therefore, the simulation shall on the one hand provide the communication module with realistic vehicle behavior to provide a good mobility model and on the other hand supply a sufficient description of the surrounding environment that can be used for computing and modeling the channel characteristics.

The second objective in substituting the real world is to replace main components of a cognitive vehicle by preliminary stubs. So far, we plan to provide stubs for perception, interpretation, communication, and distributed cooperation. The replacement by stubs is on the one hand necessary since some components rely on data which is not available in the lab (e.g. video cameras, GPS). On the other hand this is also desired as it leads to a speed-up of component development. Furthermore, having created the stubs it is possible to completely test the defined interfaces with the simulation at this early stage. Later, this facilitates the integration of components into the vehicle.

Replacing main components by stubs makes artificial data generation an integral part of the simulation. Especially the controlled generation of noisy data is a very valuable ability. Together with the ground truth information implicitly known inside the simulation, it allows benchmarking and verification of all algorithms. The range of data should comprise raw sensor data as well as high-level data such as detected objects or classification results. The range of disturbances should include noisy raw sensor data, wrong or late perception of objects, incomplete data, wrong data assignment, misclassification and misinterpretation of objects, actions and situations.

This artificial data generation together with the simulation of communication behavior makes the necessary data input not only available for a single vehicle but for all cognitive vehicles in that scene, and thus allows to develop realistically cooperative perception and cooperative driving. Vehicles can be affiliated into a Cooperative Group (CG), carrying out central decision making.

The simulation of whole traffic scenes as a closed loop allows to test the results of the decision of the CG and includes also non cooperative vehicles and traffic blocks.

B. *Structure of the simulation environment*

The requirements, described in the previous section, yielded in an integrated simulation environment which consists of six main components shown in figure 2.

The component *data storage* holds all information about the scene. We have mentioned above that the simulation is designed to be used as a replacement for the real world. Therefore, the data storage acts as a central coordination instance to allow a holistic simulation based on the artificial world driven by the simulation. The scene consists of all static and dynamic objects that model a certain scene. Static

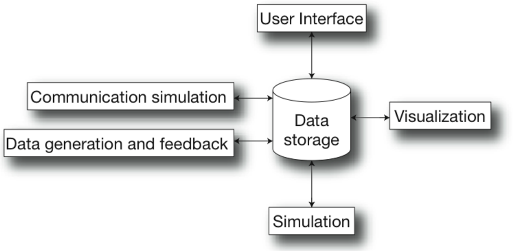

Fig. 2. Structure of the integrated simulation environment.

elements represent the environment in which dynamic objects can move, possibly obeying rules of movement imposed by the static elements. The *visualization* uses OpenGL for displaying the scene and the *simulation* takes control of all dynamic objects to be simulated. It calculates new states and positions for each time-step. The *user interface* is used to inspect data of the scene and to control the simulation behavior. A *data generation* module exists for each cognitive vehicle and simulates the vehicle's view. From the overall scene description provided by the data storage, it extracts the scene view as seen from the specific vehicle and converts the data into a format that is used throughout the project. Furthermore, it gathers data from that vehicle such as current position and internal state.

Figure 3 shows how the simulation is integrated into the overall project. As can be seen, the main interface is the data generation module. It exists for each cooperative vehicle and is connected with the perception component and the real-time database in order to transfer the generated artificial data. In the first case, raw sensor data is delivered for processing and in the second case, object hypotheses are delivered directly ignoring the perception. Additional links are provided for the project's communication component to allow the analysis of reachability and calculation of possible data bandwidth. This link is also used for the distributed decision making which facilitates quicker development and comparison of decision strategies.

IV. TRAFFIC SIMULATION

The simulation is built on top of a defined scene modeling. The scene and the objects in the scene are kept in a data storage as described in the next subsection. Then, the simulation of the behavior of a single vehicle is described, which leads to an updated position of the vehicle in the scene. Finally, the overall course of the simulation and its interaction with the project's real vehicle is shown.

A. Scene modeling and Data storage

The basic concept for modeling a scene is a *lane*. A lane is described by its geometry which can be a straight line, a circle arc or a clothoide. Furthermore, a lane is specified with multiple attributes such as left and right border (solid or broken line), maximum allowed speed and type of right of way at the end of the lane. A lane can be used to represent a real, physical lane as well as a virtual lane. Virtual lanes are used within intersections to model all possible ways that can be taken by a vehicle. The topology of a scene is defined using connections of lanes. Lanes can have left and right neighboring lanes as well as preceding and succeeding lanes.

Streets and *intersections* are defined as abstract concepts, serving as containers. Each lane is assigned exclusively to a street or an intersection. Therefore streets and intersections consist of a set of lanes and represent the topology of the scene. Connections between streets and intersections are derived utilizing the connections of the underlying lanes. In order to represent real world scenes, intersections contain GPS-coordinates and positions between intersections are derived by interpolation.

Buildings as integral parts of urban scenarios are modeled as prisms with a polygonal base and an individual height. They can also have a texture for the visualization component and additional parameters such as reflectivity for wave propagation analysis.

Static scenes can be set up using an editor and constructing the scene lane by lane. Another possibility is to use digital maps, for example from navigation systems or GIS-data, and to convert that data into the description of a scene. Additionally, an XML representation was defined which allows scene descriptions to be stored and loaded.

Having defined static scenes, the remaining objects that need to be added are *vehicles*, *obstacles* and *unknown objects*. The only difference between obstacles and unknown objects is that obstacles are static and cannot move whereas unknown objects represent un- or misclassified vehicles, other traffic participants or moveable objects such as a ball rolling over the street.

Vehicles are described by their position, their current speed and acceleration, maximum speed, minimum and maximum acceleration, and state of indicator signals and lights. Each vehicle either belongs to the class of cognitive vehicles or to the class of simulated vehicles. If it is a cognitive vehicle, it is controlled from the real vehicle system connected to the simulation. Otherwise it is in control of the simulation. The route which is to be driven by a vehicle is represented by a list of consecutive lanes.

A complete simulation scenario is thus composed of a static scene together with a set of obstacles, unknown objects and (simulated and cognitive) vehicles.

All the information of a simulation scenario is maintained in a central data storage which is also used for exchanging scene data between all connected components (simulation, visualization, communication, and data generation). For the communication between these components, a simple event mechanism was implemented. Events are used to signal addition, deletion and change of objects of the scene as well as the creation of a new scene and the completion of a simulated time step.

B. Simulation of vehicle behavior

The role of the simulation of vehicle behavior is to select the appropriate behavior for each simulated (non-cognitive) vehicle which results in an updated position. In order to

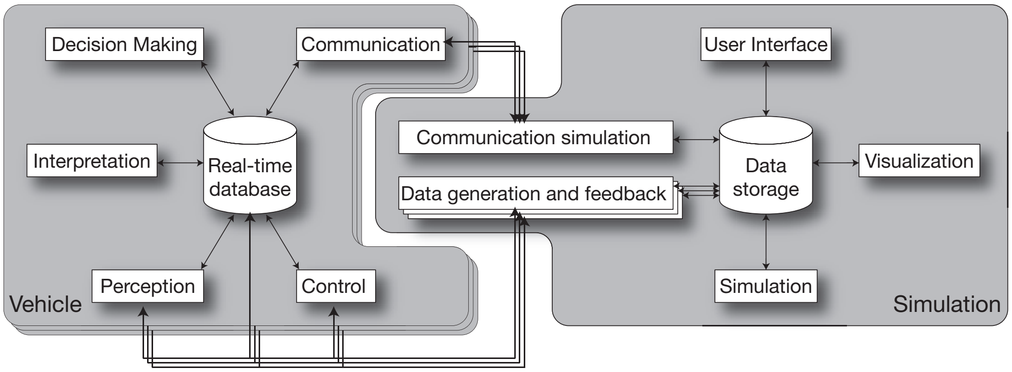

Fig. 3. Connection of the simulation framework with the vehicle's components.

select the correct behavior, the approach of Pellkofer [7] was adopted.

The idea relies on two basic concepts. Several so-called *situation aspects* are combined in a *rule-base*. The evaluation of the rule-base results in the selection of *actions* and the execution of the actions give the new position of the vehicle.

A situation aspect describes a dedicated partial state of the environment. It can be seen as a higher-level description of some part of the environment. Examples are:

- the lane on which the vehicle is currently driving
- the permitted maximum speed on a lane
- the distance to the next vehicle ahead (if any).

The rule-base consists of several rules which guarantee the compliance with traffic rules and the avoidance of collisions with other vehicles. Each rule of the rule-base checks if it is applicable by evaluating several situation aspects. The conclusion of a rule is a set of actions which are then executed. Currently, the following rules are used:

- follow vehicle
- keep speed
- respect right of way
- drive curve.

Actions which can be applied are acceleration, deceleration, steering, and blinking. Except for the blinking and steering actions, the execution of the actions results in a new acceleration value. Together with the steering action, the next position of the vehicle is then calculated using the updated acceleration value. To date, the simulation does not take into account any sophisticated dynamic model, e.g. lateral accelerations are not calculated.

C. Simulation control flow

At the beginning of the simulation, the static part of a scene is loaded which describes all streets, intersections and traffic signs. Next, a script for this scene is set up. This script contains all vehicles which participate in the simulation run. There are two types of vehicles: vehicles, which are in control of the simulation and vehicles which are cognitive vehicles and controlled outside the simulation. Now, cognitive vehicles and other components can connect to the simulation and the simulation is ready to run.

Since the simulation works with discrete time, simulation steps are only performed at defined time steps. For each time step, the order of tasks is shown in algorithm 1.

Algorithm 1 Order of one simulation step

```
Calculate new simulation time
Analyze connectivity and bandwidth of attached cognitive vehicles
for all cognitive vehicle $v_i$ do
    Generate view for vehicle $v_i$
    Add artificial noise to data
    Transform data
    Send data to vehicle $v_i$
end for
Receive updated position and state of all cognitive vehicles
Calculate position and state of all simulated (non-cognitive) vehicles
Store positions and states of all vehicles (cognitive and non-cognitive) in data storage
Update visualization
```

V. APPLICATIONS AND RESULTS

A. Data Generation

The task of data generation is to simulate a cognitive vehicle's perception. Within our framework any kind of sensor can be simulated since we have groundtruth information available. Each sensor is characterized by the following values: field of vision, object detection probability, false detection rate, classification performance, and precision. As the main sensor of our vehicle is a stereo camera, the data generation so far comprises only this sensor type. However, the following description applies also to other sensor types.

The field of vision is specified by its direction and beam width. The two parameters can be modified by other processes to realize *e.g.* attention based vision by actively rotating the cameras.

The object detection probability $p_d(obj)$ is a function depending on the state of the object. It defines the probability of the object being detected within the current frame /

simulation step. For our stereo vision sensor, we simply defined it as $p_d(obj) = e^{-c \cdot d(obj)}$, d being the distance of the object and c a scaling constant. Further sophistication could include making it dependent on the object's velocity, class[1] or size.

Yet another characteristic is the classification performance, given by the probability distribution $p(class = C|class(obj))$. It could also be made distance dependent, but for sake of simplicity we assume it to be a constant $n \times (n+1)$ matrix. n here is the number of classes, $n+1$ additionally includes the class "Unknown".

The precision of the measurements plays a central part in data generation. For each measurement value we assume a gaussian distribution which is characterized by the standard deviation and mean value *E*. Since we describe objects by their surrounding cuboid, the respective parameters are $(\sigma_x, \sigma_y, \sigma_z)$ for the corner position, for the rotation $(\sigma_\psi, \sigma_\theta, \sigma_\phi)$, and for the dimensions $(\sigma_{length}, \sigma_{width}, \sigma_{height})$. All of these parameters are assumed to be independent.

The false detection rate is object independent and defines the probability p_f of detecting an object that is non existent within the current frame. This characteristic is not yet implemented.

These characteristics can be acquired experimentally for an existing sensor. Otherwise, the values can be estimated and their impact tested by executing the simulation with various values. Values giving good results can then act as quality requirements for the development of real sensors.

When executing a simulation step for a cognitive vehicle, the following steps for each object within the simulation are processed: First it is checked whether the object lies within the field of vision and it is assured that it is not occluded. Then, the object detection flag det is calculated by

$$det(t+1) = \left[det(t) \vee (rand_0^1 > p_d(obj))\right] \wedge isVisible$$

This is responsible for randomly simulating late detection or even missing detection. Once an object has been detected, we assume that the sensor tracks it correctly until it moves out of the field of view. If an object has been detected by the sensor, we add artificial noise to the correct values by sampling from a gaussian distribution. As we reference a corner of the cuboid, this makes corners close to the own vehicle more precise than others. An example can be seen in figure 4, where a cognitive vehicle detectes another vehicle when approaching the crossing from the bottom.

Note that we include random numbers to make decisions and to add noise to the data. Yet, the use of pseudo random numbers guarantees a deterministic procedure which is repeatable. This assures that error tracing is easily possible.

The simulation framework alternatively allows data to be generated in a second way: images can be rendered from the vehicle's perspective and the vision algorithms can be directly applied on these images. Although the images are not as "real" and thus the algorithms do not lead to the same results as if executed on natural images, this allows the simulation loop to be closed completely, including all components of cognitive vehicles.

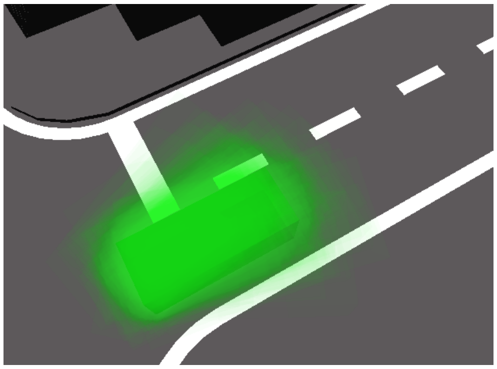

Fig. 4. Example of a detected vehicle. The amount of blurring around the cuboid denotes the probability distribution of the object position and its size.

B. Situation interpretation

One part of the interpretation of traffic situations is the recognition of vehicle behavior. The simulation environment was used for the generation of features for the classifier using the data with added noise of the data generation module, the generation of training data, and the automated evaluation of classification results. Since the simulation framework has ground truth of the behavior of the simulated vehicles, the training data for the classifier can be automatically generated. Furthermore, ground truth is also used for the evaluation of the classifier.

The behavior recognition was done using Bayesian Networks. Ten different behaviors were selected for recognition. These are standing, driving with constant speed, accelerate, brake, wait at intersection, turn left, turn right, go straight, follow a vehicle, and drive freely. The behaviors were grouped into three categories and a Bayesian Network for each category was set up. For training the networks, 4473 frames were recorded and the classifier delivered recognition rates from 83 % for acceleration up to 99 % for vehicle following.

C. Inter-vehicle communication

To enable communication between cognitive vehicles, radio equipment will be installed that allows these vehicles to exchange information and to cooperate in terms of perception and generation of appropriate behavior. Due to the lack of a fixed radio infrastructure, the communication concept imposes certain challenges on the designer: networks between vehicles have to be formed instantaneously and autonomously (so-called ad-hoc networks). Additionally, data exchange has to satisfy strict constraints concerning minimum data rate and maximum delay of transmissions.

Although various protocols for ad-hoc networks already exist (see [8], [9]), they tackle only some aspects of the particular requirements concerning communication between cognitive vehicles. The distinct mobility of vehicles in their

[1] *E.g.* radar sensors hardly detect wooden objects.

surroundings makes communication a special case of ad-hoc networks where a solution has to be carefully designed.

The simulator inherently allows a realistic mobility model for communicating vehicles to be derived: their movement is based on real-world behavior, with rules to be respected and paths to be followed. Furthermore, the description of their environment can be used to calculate the characteristics of the communication channel. Channel models based on ray-tracing have proven to yield extremely good results (see [10]) and will be used to precisely model the properties of wave propagation, allowing a very good prediction of link status and connectivity among vehicles.

In our simulation environment, the vehicles' communication modules connect to the communication simulation which, in turn, connects to the main simulation. The exchange of data between vehicles can now be simulated and the communication protocols performance be evaluated based on a specific traffic flow and environment. This allows the designer to simulate a solution in realistic environments with actual data flows and vehicle behavior.

D. Distributed, shared cooperation

Basis for the decision making of the Cooperative Group (CG) is a *common situation description* (CSD) which consists of the position of all relevant objects, in case of moving objects their speed, direction, intention, preferences and mission. It also includes the traffic infrastructure such as roads, crossings, lanes, intersections, traffic lights and road signs.

The CSD is a fusion of the individual situation description of each cooperative vehicle in the situation and is frequently refreshed. The refresh cycle depends on the features (bandwidth and reliability) of the communication system. Possible restrictions of the bandwidth, which are taken into consideration in the simulation, result in a classification of the importance of the data to be transmitted. Thus, the CSD includes all relevant cooperative vehicles as well as non-cooperative objects.

The situation is evaluated based on the CSD. The whole scene is divided into non-critical and critical sections which are investigated in more detail. This could result in instructions for the vehicles involved.

A major design decision, which is investigated in the simulation, is the partitioning of the involved vehicles into CGs. One possibility is a fragmentation, another the building of overlapping CGs. Design aspects are the simplicity of the negotiation protocol in case of dangerous situations (which is best in the case of fragmentation), the decision quality for all members of the CG (which is worst for the members on the border of the CG in case of fragmentation) and the complexity of the split and union operation for CGs.

Our first results are based on a single-lane country road and a simple algorithm for lifecycle management in cooperative groups. They show that the best result can be obtained by building a group consisting of all vehicles, whereas in the worst case decision making is based on local information only. The results of building cooperative groups dynamically lie in-between.

VI. SUMMARY AND CONCLUSION

In this work an integrated simulation framework for cognitive vehicles is presented. Its main purposes are the reproduction of traffic scenes, the generation of artificial data for all components of the cognitive vehicle and the investigation of communication strategies between multiple vehicles. Using this functionality, we presented results for situation interpretation and the analysis of strategies for common decision making of cooperative vehicles. Obviously, a simulation is only a tool which cannot substitute tests on the real vehicle, nevertheless it is a valuable tool which helps the quick development and testing of components.

One of our next steps comprises the creation of a simple scenario which we can not only test by simulation, but also executein the real-world. This will allow us to compare simulation results and detect simulation errors. Hence, the simulation can be corrected which will increase the probability of simulation results being correct for more complex scenarios such as inner-city environments.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge support of this work by Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Centre 28 ”Cognitive Automobiles”.

REFERENCES

[1] D. Vollmar, “Fahrt- und Verkehrssimulationen zur Evaluierung von verkehrsadaptiven Fahrerassistenzsystemen,” *Automotive – safety and security*, 2004.

[2] Forschungsgesellschaft Kraftfahrwesen mbH, Aachen, “PELOPS, White Paper,” http://www.pelops.de.

[3] K. Li and P. Ioannou, “Modeling of traffic flow of automated vehicles,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, 2004, pp. 2:99–113.

[4] M. J. Lighthill and G. B. Whitham, “On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads,” in *Proceedings Series A*, R. S. of London, Ed., vol. 229, May 1955, pp. 317–345.

[5] S. Ono, K. Ogawara, M. Kagesawa, H. Kawasaki, M. Onuki, J. Abeki, T. Yano, M. Nerio, K. Honda, and K. Ikeuchi, “A photo-realistic driving simulation system for mixed-reality traffic experiment space,” in *Proc. of the IEEE Intelligent Vehicles Symposium (IV’05)*, 2005, pp. 747–752.

[6] D. Neunzig, K. Breuer, and D. Ehmanns, “Traffic and Vehicle Technologies Assessment with the Simulator PELOPS,” in *ISSE Conference*, 2000.

[7] M. Pellkofer, “Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung,” Ph.D. dissertation, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, 2003.

[8] H. Xiaoyan, X. Kaixin, and M. Gerla, “Scalable Routing Protocols for Mobile Ad Hoc Networks,” *IEEE Network*, vol. 16, no. 4, pp. 11–21, July/August 2002.

[9] P. G. Argyroudis and D. O’Mahony, “Secure routing for mobile ad hoc networks,” *IEEE Communications Surveys & Tutorials*, vol. 7, no. 3, pp. 2–21, 2005.

[10] J. Maurer, “Strahlenoptisches Kanalmodell für die Fahrzeug-Fahrzeug Kommunikation,” Ph.D. dissertation, Universität Karlsruhe (TH), May 2005.