A Modular Architecture of an Interactive Simulation and Training Environment for Advanced Driver Assistance Systems

Kareem Abdelgawad, Bassem Hassan, Jan Berssenbrügge, Jörg Stöcklein, and Michael Grafe
Heinz Nixdorf Institute
University of Paderborn, Germany
{Kareem.Abdelgawad, Bassem.Hassan, Jan.Berssenbruegge, Joerg.Stoecklein, Michael.Grafe}@hni.upb.de

***Abstract* - Advanced Driver Assistance Systems (ADAS) are mechatronic vehicle systems that collaborate with the driver to improve road safety and increase driving comfort. Apart from all technical challenges regarding control algorithms and sensor quality, customer acceptance of ADAS is an important concern to automobile manufacturers. Simulating ADAS and demonstrating their benefits to customers in real traffic environments are impractical and leads to significant efforts and costs. This paper presents a modular architecture of a driving simulation environment for ADAS demonstration using driving simulators. The structure of the driving simulation environment is discussed. Special focus is given to the embedded framework for ADAS virtual prototyping and demonstration. This framework is built in a flexible form that ensures system scalability. That is, new ADAS prototypes can be designed and added almost without significant input-output interface adjustments. Furthermore, different ADAS can be integrated together to implement more advanced capabilities, such as autonomous driving. The framework is composed of modular functional units, which enclose real-time capable simulation models developed with MATLAB/Simulink. The design of the functional units and the input-output relationships are presented. Prototypes for Emergency Brake Assist and Emergency steer Assist are presented as examples of innovative ADAS that can be demonstrated using the developed simulation environment.**

Keywords - Advanced Driver Assistance Systems (ADAS); Driving simulators; Virtual prototyping; MATLAB/Simulink.

I. INTRODUCTION

Driving is one of the most popular daily activities that people perform. Nevertheless, it is a complex and relatively dangerous activity. Drivers have to concentrate on many tasks at the same time. Improving road safety standards is one of the main concerns in the automotive industry. Therefore, the automotive manufacturers develop Advanced Driver Assistance Systems (ADAS) with the aim of helping drivers in the complex driving task. ADAS are innovative mechatronic vehicle systems that monitor vehicle surroundings, as well as driving behavior [1]. They provide drivers with essential information and take over difficult or repetitive tasks. In critical driving situations, these systems warn and may intervene actively to support the drivers, and hence, lead to increased road safety. ADAS belong to the active safety systems, which help to prevent accidents or at least minimize possible consequences [2].

Using ADAS in cars and trucks has great benefits regarding accident prevention. Reference [2] presented an analysis for thousands of accidents insurance claims in Germany in order to investigate the safety benefits of ADAS. It was found that using one ADAS can prevent up to 45% of a specific type of accident. Therefore, modern vehicles are equipped with various types of sensors, which recognize and analyze the environment. Moreover, the sensory data, which is detected by each sensor can be integrated together to assure its accuracy, and hence, to take appropriate decisions.

Diverse sensor technologies (camera, radar, ultrasonic, etc.) and decision algorithms can provide different levels of assistance [3]. On the one hand, some ADAS, like, e.g., Lane Departure Warning [4], only alert the driver to critical situations by means of optical, acoustic and/or haptic feedback. On the other hand, other ADAS do not only recognize driving situations and warn the driver, but also intervene actively in order to prevent possible collisions. A common example of the latter type is Emergency Brake Assist [5], which applies full braking if driver fails to respond to obstacles in front of the vehicle.

In general, ADAS can be classified according to their functionality in two main categories [3]. Firstly, systems that support the diver and make the driving task easier, like, e.g., navigation devices, night vision systems, and auto-parking systems. Secondly, systems that support the vehicle and make the driving task more safe [3], like, e.g., Adaptive Cruise Control (ACC), Lane Keeping Assistance (LKA), and Lane Change Assistance (LCA).

Automobile manufacturers and suppliers are confronted with considerable technical challenges while developing ADAS. However, there are additional challenging aspects related to ADAS deployment and public acceptance. A flexible test environment is required in order to validate ADAS concepts and assess their decision logic. Clear concepts for driver-vehicle interface have to be addressed in early development phases; this ensures that drivers can handle the systems appropriately. On the other hand, demonstrating safety and comfort benefits of ADAS to consumers is a key factor for smooth market penetration and development.

However, validating and demonstrating ADAS in real traffic environments are impractical and lead to significant efforts and costs. Moreover, real traffic environments are principally random and do not allow for standardized driving tests or reproducible research results. Driving simulators offer a potent virtual prototyping platform to test and verify

ADAS in different development phases [6]. They allow the design, testing, and validation of ADAS in a closed loop together with vehicle components, environment, and driver. ADAS control units and vehicle components could be real, virtual, or a combination of real and virtual components. For demonstration and training purposes, driving simulators can be utilized to make drivers familiar with new ADAS, and hence, accelerate the learning phase.

Driving simulators vary in their cost, structural complexity, and validity from low-level to high-level driving simulators. They extend from driving simulation games for computers or smart phones to highly-sophisticated driving simulators incorporating complex motion platforms and high fidelity visualization systems. ADAS development requires test environments with different levels of details and complexity [7], e.g., Software-in-the-Loop (SiL), Model-in-the-Loop (MiL), Hardware-in-the-Loop (HiL), Driver-in-the-Loop (DiL), etc. For instance, while a SiL environment can be used to test basic ADAS concepts and control algorithms, a DiL environment can be utilized to address the interaction between the driver and vehicle and its systems.

The project TRAFFIS (German acronym for Test and Training Environment for ADAS) is carried out at the University of Paderborn with the target of supporting industrial development, testing and training of modern ADAS using a reconfigurable driving simulator [7]. Despite the fact that the development of driving simulators is costly and complex, available driving simulators in the market nowadays are usually special purpose facilities. They are individually developed by suppliers for a specific task. These driving simulators cannot be reconfigured, or in the best case, they have only some exchangeable components. Only a driving simulator expert can modify the system architecture or exchange one or more components. The existing driving simulators do not allow the system operator to change the system architecture or to exchange simulation models without in-depth know-how of the driving simulator system and its architecture. Therefore, the project TRAFFIS aims to develop a comprehensive environment of reconfigurable driving simulators to support ADAS development. The project is funded by the European Union “ERDF: European Regional Development Fund” and the Ministry of Economy, Energy, Industry, Trade and Craft of North Rhine Westphalia in Germany.

Three driving simulator variants with different complexity levels and simulation fidelity have been built: TRAFFIS-Light, TRAFFIS-Portable, and TRAFFIS-Full. The TRAFFIS-Full variant was first developed for the German Army in 1997 with the aim of performing safety training for the military truck drivers. The Heinz Nixdorf Institute of the University of Paderborn adopted this driving simulator in 2009 in cooperation with Rheinmetall Defence Electronics GmbH. This driving simulator incorporates a complex motion platform, which consists of two dynamical parts with 5 Degrees Of Freedom (DOF) to fully simulate vehicle lateral and longitudinal accelerations. These two parts are independent of each other and the system is fully electrically actuated. The first dynamical part is the moving base. It has 2 DOF and is used to simulate the lateral and longitudinal acceleration of the simulated vehicle. It can move in the lateral plane and at the same time, it has the ability to tilt around the lateral axis with a maximum angle of 13.5 degrees and around the longitudinal axis with a maximum angle of 10 degrees. Four linear actuators are used to control the movements in both directions. The second dynamical part is the shaker system, which has 3 DOF to simulate the roll and pitch angular movements and the heave translation of the simulated vehicle. The shaker is driven by a three drive crank mechanism and by three electrical motors. The driving simulator has an eight-channel cylindrical projection system (powered by 8 LCD-projectors), which covers a 240 degrees horizontal field of view and three displays in order to visualize the simulated rear mirror views. Moreover, the motion platform is equipped with an innovative fixation system, which allows the utilization of different driving cabins, e.g., truck cabin or passenger vehicle cabin, so that drivers experience realistic control cues. The driving simulator is operated by software developed by dSPACE and the University of Paderborn. The software consists of the simulation core, an operator council GUI, a training scenario editing tool, vehicle model, traffic model, and visualization and audio generation components. Figure 1 shows the TRAFFIS-Full driving simulator operated by the University of Paderborn.

Figure 1. TRAFFIS-Full driving simulator operated by the University of Paderborn.

The TRAFFIS-Portable variant has a pneumatic motion platform, which is composed of an actuated inverted hexapod system. A simple motion controller regulates the movements of the motion platform; it is based on virtual vehicle position and orientation. This driving simulator has a four-wall projection system. Figure 2 shows TRAFFIS-Portable driving simulator at the University of Paderborn.

Figure 2. TRAFFIS-Portable driving simulator at the University of Paderborn.

The TRAFFIS-Light variant is simple a PC-based driving simulator with no motion platform. It has a commercial wheel-transmission-pedals set and a racing seat to provide low-cost, but reasonable, physical feedback and control cues. Figure 3 shows TRAFFIS-Light driving simulator at the University of Paderborn.

Figure 3. TRAFFIS-Light driving simulator at the University of Paderborn.

These driving simulator variants, i.e., TRAFFIS-Full, TRAFFIS-Portable, and TRAFFIS-Light, together with an innovative configurability concept offer a flexible test and training environment for various in-vehicle systems [7]. However, the focus is given mainly to the development of ADAS. One particular objective of the project TRAFFIS is the development of a modular simulation environment for ADAS demonstration and training purposes. That is, a simulation framework with flexible prototyping concepts is required for easy and convenient ADAS demonstration and training. As an extension to the work presented in [1], this paper describes the structure of the whole simulation environment utilized in a driving simulator within the project TRAFFIS. Particular focus is given to the module responsible for ADAS simulation and interactive demonstration. Moreover, the main concepts of the visualisation software are presented. That is, the topic of ADAS simulation with driving simulators is addressed thoroughly in this paper.

The architecture of the ADAS virtual prototyping framework is discussed in more details. This framework consists of several functional units enclosing simulation models that were implemented with MATLAB/Simulink. The models are arranged in a modular architecture and developed, so that they communicate in a loosely coupled fashion. The design of the architecture conforms to the configurability concept discussed previously in this section. Adaptation of models interfaces can be performed with minimum effort. The design approach ensures maximum flexibility and scalability for implementing any ADAS virtual prototypes. The design of the functional units is discussed along with input-output relationships of the underlying models. All models are real-time capable, i.e., the simulation runs in real time using the Real-Time Windows Target library from Mathworks.

The developed ADAS simulation framework was integrated with the simulation environment of the TRAFFIS-Light driving simulator, which represents the simplest driving simulator variant within the project TRAFFIS. Furthermore, virtual prototypes of two innovative ADAS are presented to show and validate the capability of the simulation environment and the ADAS prototyping framework for demonstration and training.

This paper is structured as follows: Section II presents related work in the field of ADAS simulation. Section III discusses the modular driving simulation environment, with which the developed ADAS framework was integrated. Section IV presents the design approach of the developed ADAS virtual prototyping framework along with the concepts of its functional units and models. Section V demonstrates two ADAS prototypes realised with the developed framework and demonstrated using the TRAFFIS-Light driving simulator. Section VI derives the conclusion and summarizes the benefits of the presented approach. Finally, Section VII presents the future work with respect to interactive vehicle systems simulations.

II. Related Work

According to literature review, most research work in the field of ADAS simulation considers only the development of specific components, like, e.g., sensor models [8] [9], decision units [10] [11], signal or image processing algorithms [12], etc. On the other hand, there are several commercial solutions for specific ADAS simulation and development. However, a common problem among commercial solutions is the lake of sufficient modularity. In other words, they provide solutions for individual ADAS functionalities. Even if they can be parameterised flexibly to some good extent, adding new ADAS logic and integrating different ADAS functions are typical challenging issues among these commercial solutions.

ASM software from dSPACE provides flexible models for traffic and environment simulations to support the development and testing of ADAS [13]. Developers can simulate a test vehicle, complex networks, a large number of fellow vehicles, and environmental objects, like, e.g., pedestrians and traffic signs. Moreover, ASM has a graphical user interface to facilitate defining the simulation scenarios and the necessary components.

DYNA4 software from TESIS is a flexible test framework with software and hardware implementations of environment sensors and some defined ADAS functions [14]. It provides overview and automatic comparison of simulation results for further analysis.

CarMaker software from IPG presents an open test platform, which enables a wide spectrum of automotive applications beside ADAS [15]. It offers sophisticated driver model that performs complex driving manoeuvres.

SCANeR Studio software from OKTAL provides a simulation environment to prototype, test and validate some ADAS systems [16]. It includes several sensor models with different levels of complexity. Its high quality real-time visual rendering makes it suitable for camera-based ADAS simulation.

Despite promising work in the research and commercial fields, there is still no comprehensive ADAS simulation platform that can be easily and fast extended to add or integrate new ADAS functions. On contrary, the flexibility and scalability of the developed architecture in this work provide an extensive solution for ADAS simulation and development. Due to its unique modular structure, it presents no limits on the type and complexity of the simulated ADAS functions. The following section describes the developed driving simulation environment; the ADAS simulation framework was integrated with this environment.

III. Driving Simulation Environment

The simulation environment of the TRAFFIS-Light driving simulator consists of two main functional units: a vehicle dynamics model and a traffic model. Figure 4 illustrates its structure and the direction of information flow.

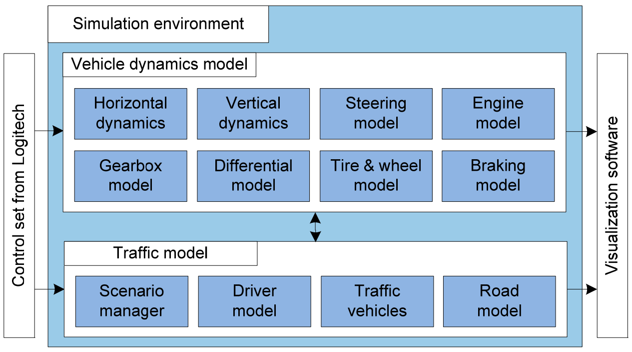

Figure 4. Simulation environment of the PC-based simulator.

Each functional unit consists of real-time capable sub-models implemented with MATLAB/Simulink. The visualization software represents the main feedback cue of the driving simulator. 3D models for the main vehicle, road, and traffic participants are controlled through the corresponding sub-models of the driving simulation environment.

The visualization software was implemented with Unity [17]; a development engine that provides rich and easy functionalities for creating interactive 3D tools. Figure 5 presents sample screen shots for the 3D environment developed with Unity.

Figure 5. Sample screen shots for the 3D environment developed with Unity software.

Night and daylight drives can be performed and the driver can be subjected to different weather conditions, like, e.g., rain, snow, fog, etc. Moreover real test tracks, city streets, and highways can be generated, this is necessary for realistic and engaging driver training. However, modeling real world roads is a cumbersome and time-consuming task. It involves a lot of manual modeling of details along the test track, such as road signs, buildings, vegetation, or other scenery details. Therefore, a method to automate the process of generating models of real roads is utilized [18]. The process uses data from a navigation database to define road sections, from which geometries are generated. This is based on official road construction regulations and guidelines. These geometries are then integrated into models of the surrounding landscape, which are generated from Digital Elevation Models (DEM), aerial images, and Digital Landscape Models (DLM) [18]. Moreover, a procedural rule system for enriching digital terrain with authentic vegetation is used [19]. This procedural approach defines planting rules, which control the placement and distribution of plants in the scenery based on data from DLM and aerial images [18]. Figure 6 shows a screen shot of a test track with vegetation generation based on the described procedural rule system, it is developed with Unity software [17].

Figure 6. Impression of a test track enriched with vegetation.

Furthermore, realistic sound effects that accompany the 3D models are also used to provide good acoustic feedback cues to the driver. Hence, visual and acoustic information from the ADAS functions are delivered to driver in accordance with traffic situations.

Regarding the hardware and mechanical components, the TRAFFIS-Light driving simulator incorporates a racing wheel-transmission-pedals set from Logitech and a racing seat from Speedmaster. That is, it is still fully interactive with respect to steering, gears, acceleration, and brake controls. This simulator and its simulation environment are considered in this work. The next sub-sections discuss each main functional unit of the simulation environment.

A. *Vehicle dynamics model*

Modeling realistic vehicle dynamics is essential for the development of different in-vehicle systems. In particular, the design of ADAS controllers relies primarily on the underlying vehicle dynamics. The utilized vehicle dynamics model produces the actual physical characteristics of the main vehicle and allows for a total of 16 Degrees Of Freedom (DOF) [20]. The so-called nonlinear double-track model is used for modeling horizontal vehicle dynamics. This model is responsible for 3 DOF: longitudinal and lateral translational motions and a rotational motion around the vertical direction of the road. Figure 7 shows the double-track model, some of the parameters used in the differential equations of this model are also depicted [20].

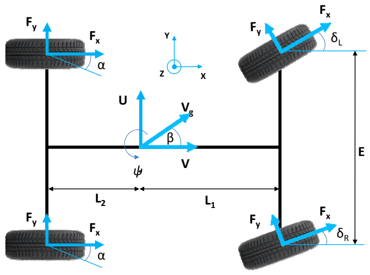

Figure 7. Double-track model for horizontal vehicle dynamics.

In the double-track model, the longitudinal and lateral velocities, as well as the yaw rate of the vehicle are described by a set of differential equations using Newton's law of motion and some basic geometrical relationships [20].

The vertical dynamics of the vehicle depends principally on suspension units at each wheel of the vehicle. The chassis of the vehicle is connected to four wheels through these suspension units. Each unit consists of a simple mass-spring-damper model [21]. Springs and dampers represent the four shock absorbers of the vehicle. The units are constitutively connected through basic mathematical and geometrical relationships [21]. Figure 8 illustrates a sketch for the vertical dynamics of the vehicle.

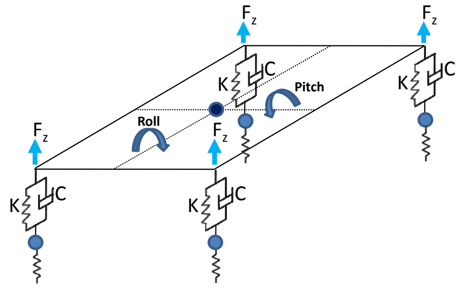

Figure 8. Vertical vehicle model with simple suspension units.

Each wheel has a relative vertical translational motion and a rotational motion around the wheel axis. In addition, each of the front wheels has a relative rotational motion around the vertical direction of the road. The vehicle dynamics model receives control signals from the hardware control set and calculates the resultant motions; these are exported mainly to the visualization software to update vehicle position and orientation on the screen.

The traffic model provides information about the road, i.e., height and friction under each of the vehicle tires; these in turn are used by the vehicle dynamics model to update the calculations of the vehicle position, orientation and speed. The vehicle dynamics model is composed of various sub-models [21]. It implements the blocks shown in Figure 4 as modular Simulink subsystems.

B. *Traffic model*

The traffic model is used to simulate the surrounding vehicles and the road [22]. It simulates realistic behavior of the traffic vehicles and their interactions, which is necessary to give realistic feedback cue to the driver on the one hand, and to efficiently test ADAS functions on the other hand. The traffic model consists mainly of four sub-models: road model, traffic vehicles models, driver model and a scenario manager model. Figure 9 shows these sub-models and their interconnections. The traffic model receives current position, orientation and speed of the main vehicle from the vehicle dynamics model; these are used mainly by the driver model to arrange for appropriate traffic flow without collisions with the main vehicle.

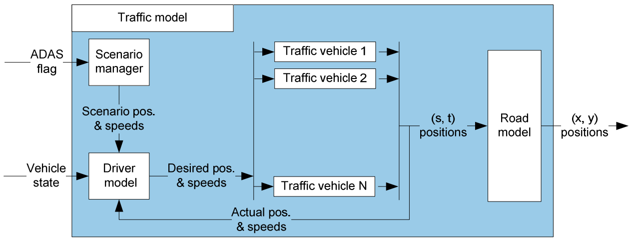

Figure 9. Traffic model and its sub-models.

The road mathematical model is a Matlab function implemented in Simulink, it is responsible for two tasks. The first task is to perform the necessary transformations from local coordinate system (s, t) to global coordinate system (x, y) used by the visualization software. The position of each object within the simulation environment is defined relative to road local coordinate system. However, the visualization software defines each object in 3D world relative to a global coordinate system, which is fixed to the ground. Both are right-hand coordinate systems.

The road consists simply of four straight segments and four round corners. Each road segment has a mathematical description that correlates the (s, t) and (x, y) coordinate systems. Figure 10 shows the geometrical design of the road, the origins of the local and global coordinate systems are depicted together with a numerical example of a sample input (s, t) point and the corresponding output (x, y) point from the road model.

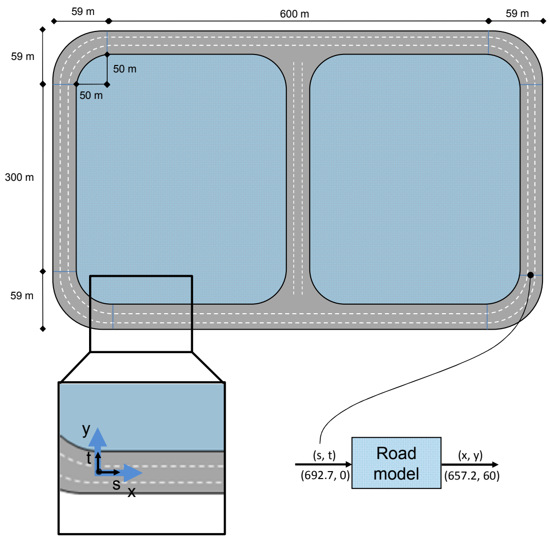

Figure 10. Road geometrical design.

A simple geometrical structure was designed to facilitate the mathematics of coordinate transformation. This also simplifies the implementation of the road 3D model. The second task of the road model is to define the friction 'f' and height 'z' of each point (s, t) of the road. The 'z' position is required by the visualization software for appropriate objects positioning within the 3D world. Both 'f' and 'z' values are required by the main vehicle model; they contribute to the calculations of horizontal and vertical vehicle dynamics, respectively.

Each traffic vehicle model consists of two sub-models: longitudinal direction vehicle sub-model and lateral direction vehicle sub-model. The longitudinal direction sub-model receives the desired s-speed from the driver model, discussed shortly. It calculates the actual s-speed with a smooth transition, which results from a combination of a simple second-order system and a P-controller. The actual s-position of the traffic vehicle is then calculated by integrating the actual s-speed. Similarly, the lateral direction sub-model receives the desired t-position from the driver model. It calculates the actual t-position with a smooth transition, which results from a combination of a simple second-order system and a P-controller. The idea of the traffic vehicle model is to produce smooth and realistic, i.e., not abrupt, movements for the traffic vehicles [22]. This is achieved through the transitional response of the second-order system to unit step inputs of the driver model. The model can be replicated arbitrarily according to the desired number of traffic vehicles. Figure 11 shows a traffic vehicle model and its main connections with the driver model. The calculated (s, t) position of each traffic vehicle is exported mainly to the visualization software to update the position of the corresponding 3D models.

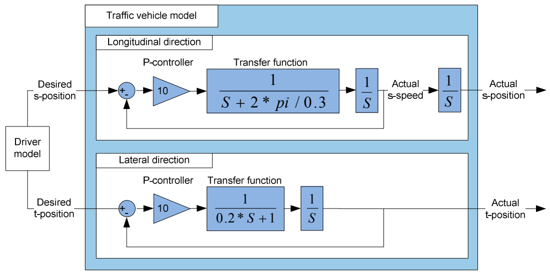

Figure 11. Traffic vehicle model.

The driver model is a Matlab function implemented in Simulink. It arranges for smooth traffic flow by controlling the speeds, and hence the positions, of all traffic vehicles. The driver model calculates and adjusts the speeds according to the current traffic situation. It receives the current (s, t) position of each traffic vehicle as well as the position and orientation of the main vehicle. Accordingly, it monitors the distances between all the vehicles on the road and overrides the default speed values of the traffic vehicles in case of any possible collision. The traffic vehicles have to follow the predetermined longitudinal speed and lateral position given by the driver model.

The scenario manager model is used for moving the traffic vehicles to compose a specific traffic situation, like, e.g., a sudden vehicle incursion from right. It is a Matlab function implemented in Simulink. The scenario manager observes the position and speed of the main vehicle. It moves the traffic vehicles according to a desired predefined scenario. According to the vehicle systems or functions under test, arbitrarily further traffic scenarios can be added to this model. The driver model receives the vehicle positions and speeds determined by the scenario manager model. According to the current traffic situation, the driver model decides whether to execute the orders of the scenario manager or to override them. The main target is it to achieve the desired traffic scenario with smooth flow and without vehicle collisions. Switching between the different scenarios can be performed during simulation runtime.

IV. ADAS Simulation Framework

The vehicle dynamics model and traffic model constitute the central functional units of a simulation environment for a simple driving simulator. However, a comprehensive simulation framework is still required to conveniently simulate different ADAS functionalities. Active safety in general and ADAS in particular exhibit continuous development. New ADAS functions are developed to achieve safer traffic flow and more comfortable driving. Moreover, the availability of a wide range of sensors and the possibility to integrate different sources of information allow the development of more new reliable ADAS. Hence, one principal requirement for building a flexible ADAS test and training environment is to maintain maximum modularity and scalability. The developed ADAS virtual prototyping framework is structured in a modular form that ensures its scalability. That is, new ADAS prototypes can be added almost without significant input-output interface adjustments. Furthermore, different ADAS can be integrated together to implement more advanced capabilities such as autonomous driving.

Driving is a multitasking activity, where drivers have to manage their attention between various actions and reactions within a dynamic traffic environment [23]. The design approach of the developed ADAS simulation framework is based on an analogy between human driving behavior and the functionality of ADAS. Figure 12 shows the structure of the ADAS simulation framework. It consists of four functional units or stages: user interface stage, recognition stage, guidance stage and control stage. The latter three functional units resemble the activity model the human driver mainly follows while driving a vehicle. The recognition stage represents the senses of human drivers for recognizing road path and other traffic participants, i.e., current traffic situation. The guidance stage corresponds to the reasoning capabilities of the human driver and compromises made according to the recognized traffic situation, i.e., decisions to accelerate, brake, steer or to make a certain maneuver. The control stage simulates the actual physical actions the human driver performs to carry out appropriate decisions.

Related approaches for human driving models are presented in [24] and [25]. The analogical comparison with human drivers is valid under the assumption that any ADAS can be represented as an assisting automatic driver that warns the driver and/or takes over the driving tasks in critical traffic situations.

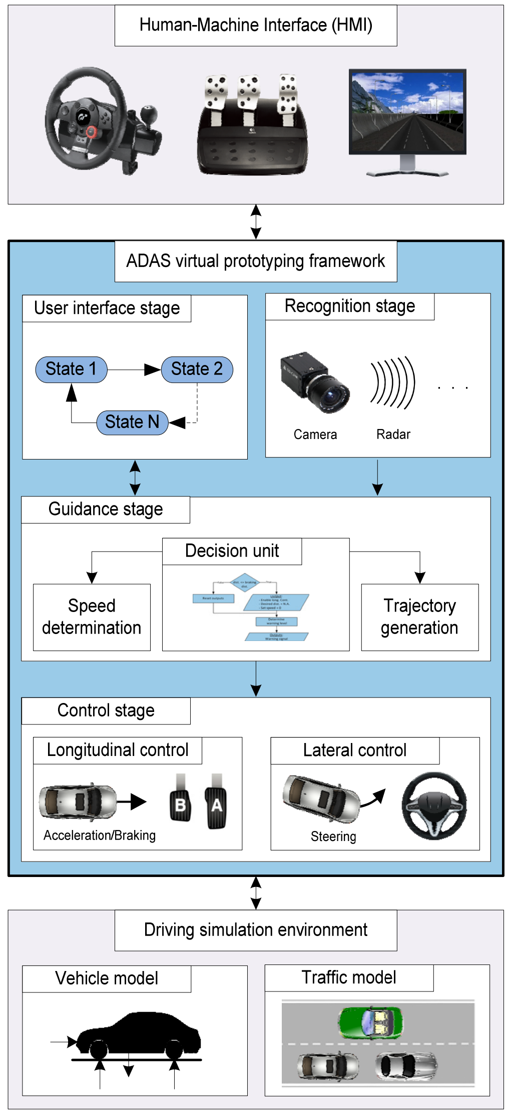

Figure 12. ADAS simulation framework and its relation with the driving simulation environment and HMI.

As shown in Figure 12, the ADAS virtual prototyping framework is connected to the other functional units of the driving simulation environment and the hardware controller set along with the visualization software (HMI) of the TRAFFIS-Light driving simulator. The ADAS simulation framework receives inputs from the HMI to set the ADAS states, i.e., activate, deactivate, or alter some parameters. It eventually applies force feedback on the steering wheel according to the driving situation and the type of the activated ADAS. The ADAS simulation framework gets the states of the main vehicle, i.e., position, orientation and speed, which are calculated by the vehicle dynamics model. In case of ADAS with active intervention, it overrides the requests of the human driver and controls the states of the vehicle. The ADAS simulation framework notifies the traffic model regarding the activated ADAS, the traffic model invokes in turn predefined traffic scenarios and provides information about the traffic participants. The following sub-sections discuss the design of each functional unit of the ADAS simulation framework and the fundamental input-output signals.

A. User interface stage

The user interface stage accounts for the interaction between user, i.e., simulator driver, and the ADAS simulation framework. It implements the logic required for transitioning between different ADAS functional states, like, e.g., on, off, standby, etc. Each ADAS user interface is modeled separately as a Stateflow sub-model (a control logic tool used to model event-driven systems within Simulink). Figure 13 shows the structure of the user interface stage and the main input-output signals.

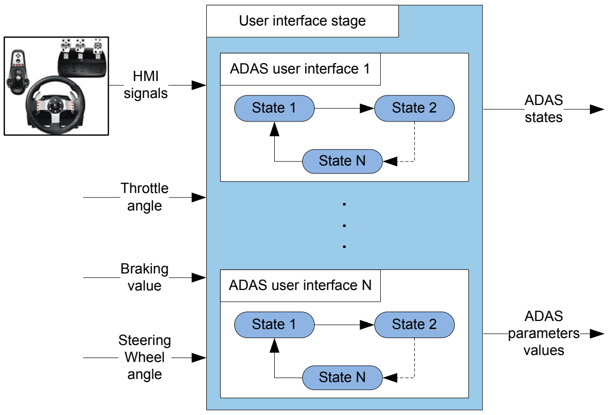

Figure 13. ADAS user interface stage.

Each sub-model receives an enable/disable signal from the buttons set, as well as the values of the acceleration and brake pedals, gear selector and steering wheel of the Logitech controller. Furthermore, it gets feedback signals indicating the desired maneuvers of ADAS controllers, namely, throttle angle, braking value and steering wheel angle. These are compared with corresponding signals indicating the intention of the driver, which is provided through the HMI. If there is a difference, and taking ADAS type into account, the corresponding sub-model decides if ADAS should make a transition from one functional state to another. For instance, while an autonomous driving function will be deactivated if the driver moves the steering wheel slightly; an emergency braking function will not be deactivated for such an action.

As outputs, indications for ADAS functional states along with the desired ADAS parameter values are exported to the corresponding ADAS sub-routines within the guidance stage, discussed in a later section.

This arrangement for the user interface stage conforms to the modularity and scalability requirement of the ADAS simulation framework. For modeling new ADAS, corresponding Stateflow sub-models have to be implemented separately within the user interface stage using the same set of input-output interfaces.

B. Recognition stage

Driver assistance systems require surrounding recognition capabilities to be able to perceive the traffic environment. Any ADAS must incorporate one or more sensors, like, e.g., GPS, cameras, radar, ultrasonic, laser, lidar. Many variants already exist in market; moreover, a lot of new sensor technologies and concepts are being developed, like, e.g., sensor fusion [26]. Hence, there are a lot of sensor models to be integrated in order to achieve a comprehensive ADAS virtual prototyping framework. The recognition stage is composed mainly of two units: a detection unit containing different sensor models and a relevance filter unit. Figure 14 shows the structure of the recognition stage and the essential input-output signals.

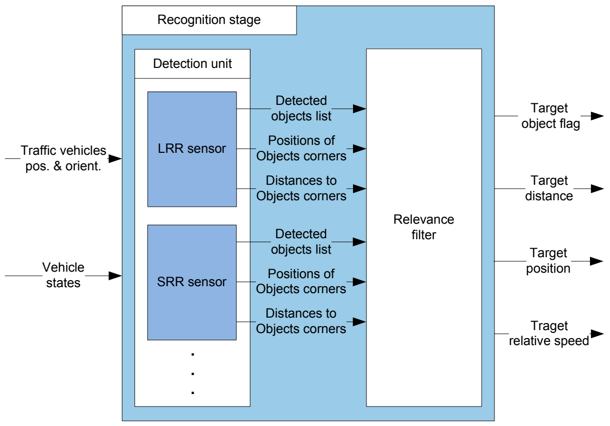

Figure 14. ADAS recognition stage.

Information about road and traffic participants is provided through the traffic model. Vehicle position, orientation and speed, i.e., vehicle states, are provided by the vehicle dynamics model. The detection unit is designed in the form of a bowl that contains different sensor models, like, e.g., radar sensor model, ultrasound sensor model, etc. The main output from a sensor model is a list of objects characterized with detection flags, i.e., detected objects list. In addition, each sensor model provides the positions and distances of detected object corners. Short-Range Radar (SRR) and Long-Range Radar (LRR) sensor models have been implemented within the detection unit. Both models are based on the mathematical description or geometry of detection area [27]. The long-range radar model is ideally suited for detection distance longer than 30 meters; it can typically detect objects 250 meters away. On the other hand, the short-range radar model provides wider view and detection distance below 30 meters. All parameter values can be modified to alter the geometrical description of detection area if necessary, i.e., the geometrical coverage and detection range are adjustable, so that sensor characteristics can be changed arbitrarily.

Within the relevance filter unit, detected objects are further filtered according to the position and orientation of the main vehicle relative to the road. That is, the outputs of all sensor models are forwarded to a relevance filter, which generates a flag indicating the most relevant object to the main vehicle, i.e., target object. Moreover, relative speed of the target object and distance and position of its nearest corner are calculated. Figure 15 illustrates the selection functionality of the relevance filter unit.

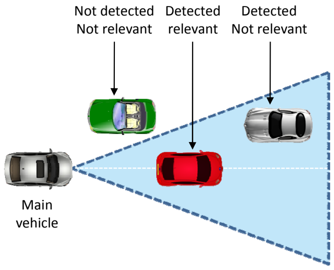

Figure 15. Target object selection of the relevance filter unit.

The detection unit is extensible for additional sensor models to be developed, whereas the functionality of the relevance filter unit has not to be altered. However, the relevance filter unit considers only sensors of the same direction of detection and determines only one target object. If other sensor models for other directions of detection are to be implemented, like, e.g., right and left sides of the vehicle, corresponding relevance filter units have to be designed conforming to the structure of the recognition stage and the same set of input-output signals.

C. Guidance stage

As mentioned previously while making analogy between the developed ADAS simulation framework and the human driving model, the guidance stage represents the understanding of recognized traffic situations and the decisions required for safe or comfortable driving. Figure 16 shows the structure of the guidance stage and the main input-output signals.

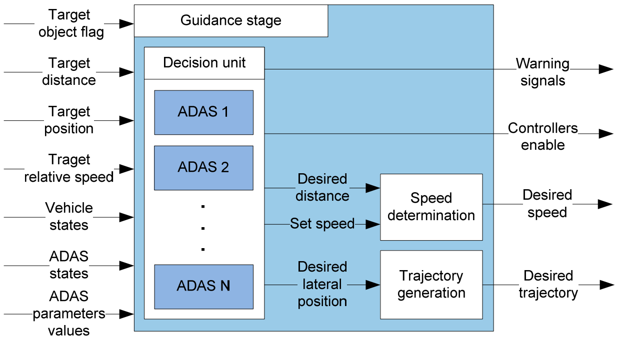

Figure 16. ADAS guidance stage.

The guidance stage derives its central role from being in the middle of a detection phase, i.e., recognition stage, and an action phase, i.e., control stage. On the one hand, it interprets the information provided by the recognition stage, i.e., it evaluates the perceived traffic situations. On the other hand, it determines the actions required to avert undesirable traffic situations.

The guidance stage is consisted of three sub-functions: Decision unit, speed determination and trajectory generation. These sub-functions are discussed next.

- Decision unit

The logic of each ADAS is implemented within the decision unit as a separate sub-routine. The decision unit receives indication for the presence of a target object along with its relative speed, distance to and position of its nearest corner from the recognition stage. Figure 17 shows a flow chart for the main function of the decision unit.

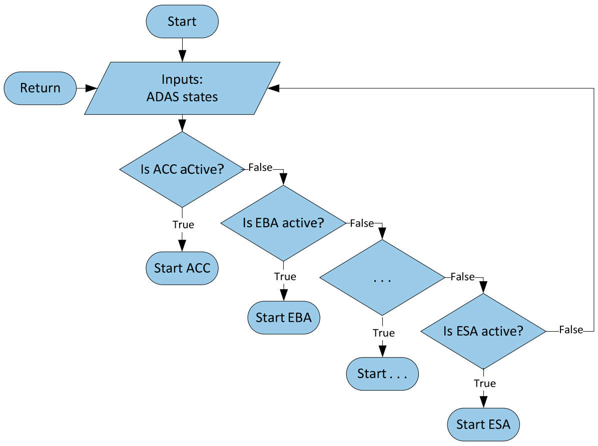

Figure 17. Transition logic between ADAS sub-routines.

The user interface stage implies which ADAS is to be activated with which parameter values. The main function of the decision unit loops through all the implemented ADAS sub-routines. Only that of the chosen ADAS is executed while other ADAS sub-routines are ignored. It considers the traffic situation detected by the recognition stage, ADAS states and parameter values exported by user interface stage and vehicle states provided by vehicle dynamics model. Accordingly, it determines desired distance to a target object, set speed or desired lateral position required to alter the path of the main vehicle. In addition, it sends enable signals to corresponding vehicle controllers, i.e., longitudinal and/or lateral controller, discussed in a later section. The activated ADAS generates warning signals required to trigger some display elements within the visualization software.

Similar to the user interface stage and recognition stage, the decision unit is extensible, so that any logic for new ADAS prototypes can be simply added as new separate sub-routines. The set of input-output signals is comprehensive and suitable for almost all active and passive ADAS.

- Speed determination

This function maintains constant time headway space to a target object that eventually drives with lower speed than that of the main vehicle [28]. Principally, the headway distance varies with main vehicle speed; this allows for a fixed margin in time for the ADAS to react to changes in the speed of the target object. The speed determination function is basically a distance controller that determines the speed required to maintain the desired headway space, taking the speed of the target object into account. It is based on the so-called slide mode control [29]. It is a simple control method that proves good stability especially, where the control actions are discontinuous functions of system states and inputs.

The speed determination function handles the orders of the decision unit with respect to the longitudinal direction. While the desired headway space is provided by the decision unit, i.e., the sub-routine of an activated ADAS, a speed command is generated to obtain this distance accordingly. Figure 18 shows the difference between the desired and actual headway distances.

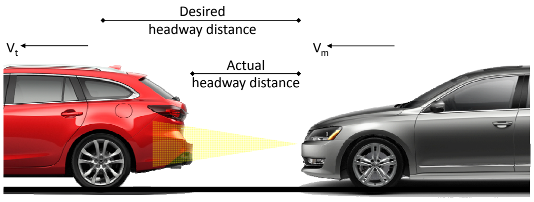

Figure 18. Headway distance control and speed determination.

Moreover, the function selects the minimum of the ADAS set speed, like, e.g., set speed of an adaptive cruise control, and that required for following a target object while preserving constant headway space. Finally, the desired speed is forwarded to the longitudinal controller discussed in a later section.

- Trajectory generation

This function generates the trajectory required to guide the vehicle through the road or to move it from one lateral position to another. The function encloses the mathematical description of the road, so that the generated trajectory reconsiles with road path. The trajectory is generated in the form of a moving point in front of the vehicle. The activated ADAS within the decision unit determines the desired lateral position required to adjust the vehicle path or to avoid a collision for example. The function limits the rate of lateral position change generated within the decision unit in order to obtain reasonable and realistic lateral transitions. Although it handles the orders of the decision unit mainly with respect to the lateral direction, the function adds a predetermined offset to the longitudinal component of current vehicle position. Hence, the location of the moving point is updated continuously and gradually to form the desired trajectory, as shown in Figure 19.

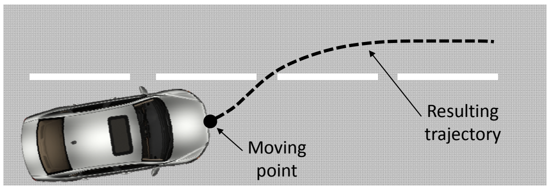

Figure 19. Moving point for trajectory generation.

The desired trajectory represented as postion updates is forwarded then to the lateral controller discussed in a later section.

D. Control stage

A motion controller is required in order to control the state of the vehicle in case of active ADAS intervention. As shown in Figure 20, decoupled longitudinal and lateral controllers were implemented to execute the orders of the guidance stage and guide the vehicle accordingly.

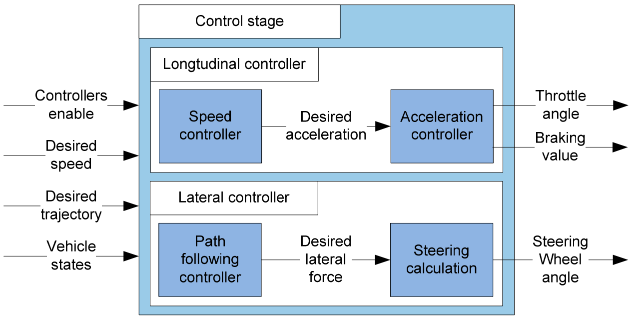

Figure 20. ADAS control stage.

The control stage gets an enable signal from the guidance stage that indicates which controller is to be activated, and hence, moving the vehicle with a desired speed in a desired direction. These controllers are discussed next.

- Longitudinal controller

The longitudinal controller is a cascaded speed-acceleration control loop system [30]. It is composed of two successive controllers: speed controller and acceleration controller. The speed controller is a Proportional-Integral (PI) type that constitutes the outer loop of the longitudinal controller. The speed command from the guidance stage is compared with the actual speed of the vehicle to generate a speed error. The speed controller generates an acceleration value required to overcome the speed error. It is followed by an anti-windup function to prevent output saturation [31]. The desired acceleration is forwarded then to the acceleration controller.

The acceleration controller constitutes the inner loop of the longitudinal controller. The desired acceleration is compared with the actual acceleration of the vehicle to generate an acceleration error. The acceleration controller implements the inverse form of vehicle dynamics and drivetrain of the vehicle model [32]. The acceleration controller is composed mainly of three sub-models, as shown in Figure 21.

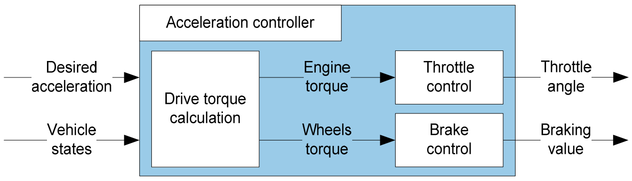

Figure 21. Sub-models of the acceleration controller.

The drive torque calculation sub-model generates the wheels torque and engine torque required to achieve the desired acceleration. It is based on the dynamics equations of the vehicle model. The throttle control sub-model generates the throttle angle according to the required engine torque. It is based on the engine model within the vehicle dynamics model. Similarly, the brake control sub-model generates the braking value according to the required wheels torque [33]. It is based on the braking model within the vehicle model. The longitudinal controller exports the throttle angle or braking value to the vehicle dynamics model. For comfort driving and realistic vehicle behavior, the throttle and brake control sub-models do not allow the acceleration and deceleration to exceed predetermined limits.

- Lateral controller

The lateral controller handles the path following control problem, i.e., how to control the vehicle, so that it can faithfully follow a prescribed path. As shown in Figure 20, it is composed mainly of two sub-models. The path following controller sub-model gets the trajectory generated by the guidance stage in the form of a moving point, i.e., a point directly in front of the vehicle that updates its location on a certain path. It calculates the front axle force required to let the vehicle adjust its orientation, and hence, follow the moving point to pursue the desired trajectory. The path following controller is based on the feedback linearization control method [34]. The basic idea is to convert the closed-loop control system including the plant, i.e., the horizontal vehicle dynamics model in this case, into linear system dynamics. The method was applied to the bicycle vehicle model [20] and showed optimal robustness even at stability borders, such as rapid steering maneuvers or driving at relatively high speeds in sharp curves. According to the horizontal vehicle dynamics, the steering calculation sub-model determines the steering angle, which corresponds to the desired lateral force. Moreover, it calculates the steering wheel angle using the inversion of the steering model within the vehicle dynamics model. Finally, the lateral controller exports the steering wheel angle required to guide the vehicle in the desired direction to the vehicle dynamics model, and hence, following a certain trajectory.

The designed longitudinal and lateral controllers can serve a variety of active ADAS functions, where a spontaneous rapid maneuver or the whole driving task is taken over by an automated intervention. The generality and simplicity of the interface between the developed guidance and control stages make it convenient to develop and plug new ADAS functions. The following section presents the logic of two innovative ADAS functions implemented in the decision unit within the guidance stage.

V. ADAS Prototypical Implementation

The developed ADAS virtual prototyping framework can be used for simulating almost any ADAS function. The recognition stage can be extended for additional sensor models. The guidance stage is also extensible, so that any logic for new ADAS functions can be simply added as new separate sub-routines. The control stage covers the longitudinal and lateral directions, and hence, it can be used principally for any active ADAS.

To prove its usability in general and to show the benefits of its modular structure in particular, prototypes for two new ADAS were implemented: Emergency Brake Assist and Emergency Steer Assist. These systems aim to help drivers to avoid accidents by alerting them to a potential collision and initiating automatic braking or steering maneuver. They represent the state of the art in ADAS development [35]. Although they have different types of intervention, both functions have been implemented without any special interface adjustments due to the modularity and scalability of the ADAS simulation framework described in this paper.

A. *Emergency Brake Assist*

Emergency Brake Assist (EBA) is an ADAS sub-routine implemented within the decision unit of the guidance stage. The system compensates for failures in the driver's action on the brake pedal. In general, drivers in emergency situations tend to apply insufficient pressure or release braking pressure too early. Figure 22 shows a flow chart for a simplified version of the EBA sub-routine.

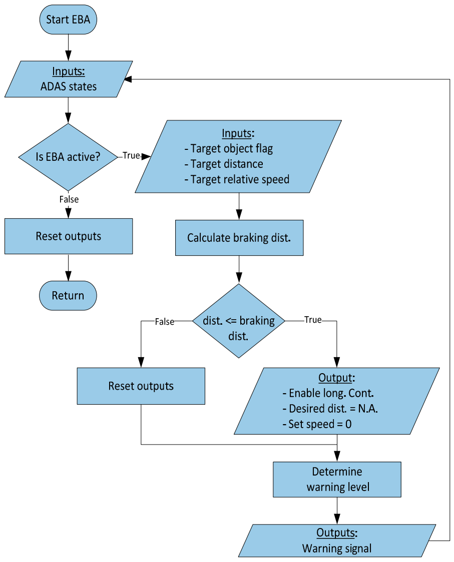

Figure 22. Simplified version of EBA logic within the decision unit.

According to recognized moving or standing objects in front of the vehicle, EBA initiates automatic braking in the case of a potential rear-end collision provided that the driver has not responded to prior warnings signals [36].

The intention of the driver is observed through the user interface stage, and hence, is embedded within the ADAS states signal. The EBA sub-routine gets the distance and relative speed of a target object existing in front of the vehicle from the guidance stage. The critical braking distance, i.e., safe distance, is calculated from the provided inputs. This means, the safe distance is variable and depends mainly on the relative speed of the target object. If the actual distance to the target object gets close to the safe distance within predefined limits, the function initiates optical and acoustic warning signals to be handled by the visualization software.

The optical warning has three levels: a green cautionary signal if the target object ahead is close, a yellow alert signal if the safe distance is reached and a red critical signal if the actual distance is equal to or fell below the safe distance. In the latter case, if the driver fails to take braking or steering actions, i.e., when an emergency situation is fully confirmed and the state of the target object flag does not change, the EBA sub-routine enables the longitudinal controller and sets the speed to zero. The sub-routine overrides the acceleration request of the driver who is effectively taken out of the loop. However, the driver still can retain control anytime by taking an appropriate steering action, and hence; changing the state of the target object flag.

The function was tested and validated with many test scenarios, where different values for the speed of the main vehicle and traffic vehicle ahead were considered. Figure 23 illustrates the switching point between warnings and active intervention distances of the EBA sub-routine

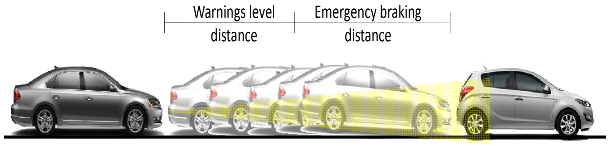

Figure 23. EBA intervention in case of no driver response.

B. *Emergency Steer Assist*

Emergency Steer Assist (ESA) is an ADAS function implemented within the decision unit of the guidance stage. The function supports the driver in the lateral driving task [36]. According to recognized sudden right or left incursion from a traffic object and if the driver has no time left for braking, the function initiates rapid automatic steering intervention in the case of predicted collision, as shown in Figure 24. ESA calculates the optimal trajectory around the appeared object and applies steering torque to help to follow the trajectory and stabilize the vehicle. However, the driver remains in control of the vehicle and can override the system at all times

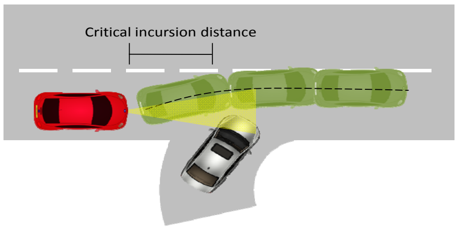

Figure 24. ESA intervention due to sudden road incursion.

Almost similar to the Emergency Brake Assist function, the intention of the driver is observed through the user interface stage. Figure 25 shows a flow chart for a simplified version of the ESA sub-routine.

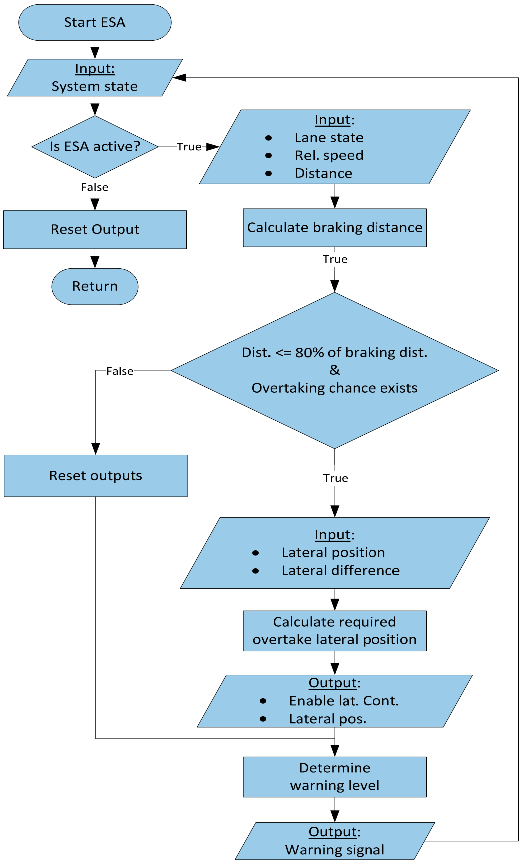

Figure 25. Simplified version of ESA logic within the decision unit.

If a target object appeared suddenly within the lane of the vehicle, the function decides to steer the vehicle abruptly in the opposite direction. This decision takes the form of a desired (x, y) point, which is exported to the lateral controller. The speed of the vehicle, the distance at which the target object appeared and the intention of the driver are factored in the decision of the function. The critical incursion distance is variable and depends mainly on the speed of the vehicle [36]. Figure 26 shows a screen shot while ESA performs a rapid maneuver to avoid a pedestrian.

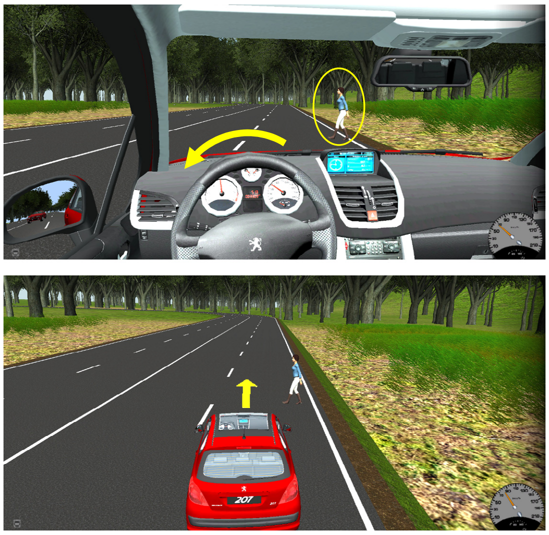

Figure 26. ESA performs a rapid maneuver to avoid a pedestrian.

The function was tested and validated with test scenarios, where different values for the speed of the main vehicle, as well as different distances to the incurring target vehicle were considered.

VI. Summary and Conclusion

Advanced Driver Assistance Systems (ADAS) gain importance due to their safety and comfort features. The ADAS virtual prototyping framework described in this paper offers a flexible solution to efficiently validate ADAS concepts and easily demonstrate their benefits to customers. The presented approach is based on an analogy between the functionality of ADAS and the human driving model. This resulted in a comprehensive architecture, which is composed of modular and extensible functional units.

The developed ADAS virtual prototyping framework was integrated with the real-time simulation environment of the TRAFFIS-Light driving simulator. To validate the approach and the capabilities of the developed ADAS simulation framework, prototypical implementation of two innovative ADAS functions was presented. Although both functions show different types of intervention, no special signal interface adjustments were necessary. The design of the other functional units of the simulation environment, i.e., vehicle dynamics model and traffic model, has not to be adjusted for any future ADAS prototypes.

A group of test persons were involved in the behavioral validation process of the driving simulator after integrating the ADAS virtual prototyping framework [37]. In other words, an assessment of how drivers react and perform with respect to the implemented ADAS prototypes has been made. The test persons have been subjected to near collision situations, where different values for the speed of the main vehicle and traffic vehicle ahead were considered. The behavioral validation process showed how the test persons could reasonably handle ADAS warnings and active interventions with very good learning curves. Effectiveness, proper operation and drivers' acceptance of the implemented ADAS were evaluated.

The presented approach added new capabilities to the PC-based driving simulator for assessing ADAS algorithms and performing drivers training by means of a driving simulation environment. In general, the modularity and scalability requirement of an ADAS training environment for the project TRAFFIS was fulfilled.

VII. Future Work

Driving simulators are built to address several aspects in the automotive and transportation fields. They are used for the development of in-vehicle systems, analysis of driving strategies, as well as for demonstration and training purposes.

The majority of available simulators are single-user stand-alone systems. However, as vehicle systems are becoming more complex, driving simulation must keep up in terms of scalability and flexibility. For instance, the significance of C2X-Communication systems has grown in the recent years [38]. These systems allow the vehicles to communicate with other each other, as well as with road infrastructure [39]. Similarly, cooperative advanced driver assistance systems, i.e., interconnecting driver assistance systems of different vehicles on the road, are gaining a lot of attention [40] [41]. These systems benefit from the new communication technologies and the utilization of GPS receivers in vehicles. They add new dimensions of safety, comfort, and optimized traffic flow.

Testing cooperative vehicle systems is even harder than testing traditional stand-alone driver support systems [42]. There are more vehicles and interaction possibilities with each other and road infrastructure. As a potent testing platform, future driving simulation should allow realistic cooperation between different interactive simulation entities, which represent their counterparts in real traffic situations.

Networked driving simulation systems can facilitate this challenge [43]. They allow developers to embed the logic of future cooperative vehicle systems into realistic and interactive traffic scenarios without the effort and costs of real test-drive [44]. Moreover, multi-user driving simulators that communicate with each other can demonstrate realistic effects of driver-driver interaction in more complex simulation scenarios. This is the major motive for extending the developed simulation framework to allow for the simulation of multi-user interactive driving simulation.

The development of such a networked driving simulation system includes many challenging tasks in order to provide a reliable and realistic simulation environment. For example, it requires utilization of the so-called global time management [45] [46]. That is, synchronizing the local time and event processing of each individual driving simulator in order to guarantee simulation reliability and test accuracy [47].

The next steps aim to develop a concept for a synchronization mechanism for networked driving simulators. This will allow the developed simulation framework to be utilized in a distributed driving simulation system. The synchronization mechanism should facilitate coordination among different participating simulators, which interact within one traffic simulation scenario. In particular, it should guarantee performance and efficiency of the driving simulation system. Finally, the new design approach has to ensure a modular structure that allows easy integration and exchange of driving simulators in a network.

REFERENCES

[1] K. Abdelgawad, M. Abdelkarim, B. Hassan, M. Grafe, and I. Gräßler, "A Scalable Framework for Advanced Driver Assistance Systems Simulation," In Proceedings of SIMUL 2014, the Sixth International Conference on Advances in System Simulation (IARIA), Nice, France, October 2014.

[2] T. Hummel, M. Kühn, J. Bende, and A. Lang, "Advanced Driver Assistance Systems – An investigation of their potential safety benefits based on an analysis of insurance claims in Germany," German Insurance Association - Insurers Accident Research, Research Report FS 03, Berlin, 2011.

[3] J. Golias, G. Yannis, and C. Antoniou, "Classification of driver assistance systems according to their impact on road safety and traffic efficiency," In Transport Reviews - Journal of Intelligent Transportation Systems, Taylor & Francis Group, vol. 22, 2002, pp. 179-196.

[4] P. Hsiao, K. Hung, S. Huang, W. Kao, and C. Hsu, "An embedded lane departure warning system," IEEE 15th International Symposium on Consumer Electronics (ISCE), Singapore, June 2011, pp. 162-165, ISSN: 0747-668X, ISBN: 978-1-61284-843-3.

[5] R. Zheng, K. Nakano, S. Yamabe, M. Aki, and H. Nakamura, "Study on Emergency-Avoidance Braking for the Automatic Platooning of Trucks," IEEE Transactions on Intelligent Transportation Systems, China, August 2014, Vol. 15, No. 4, pp. 1748-1757, DOI: 10.1109/TITS.2014.2307160, ISSN: 1524-9050.

[6] F. Colditz, L. Dragon, R. Faul, D. Meljnikov, and V. Schill, "Use of Driving Simulators within Car Development," In Proceedings of Driving Simulation Conference, North America, Iowa City, USA, September 2007.

[7] B. Hassan and J. Gausemeier, "Concept of a Reconfigurable Driving Simulator for Testing and Training of Advanced Driver Assistance Systems," IEEE Transl. ISAM 2013 China, vol. 2, July 2013, pp. 337-339.

[8] S. Pechberti, D. Gruyer, and V. Vigneron, "Radar simulation in SiVIC platform for transportation issues. Antenna and propagation channel modelling," IEEE Transactions on Intelligent Transportation Systems, Alaska, USA, September 2012, pp. 469 - 474, DOI: 10.1109/ITSC.2012.6338631, ISSN: 2153-0009.

[9] H. Chiang, Y. Chen, B. Wu, and T. Lee, "Embedded Driver-Assistance System Using Multiple Sensors for Safe Overtaking Maneuver," IEEE Systems Journal, November 2012, Vol. 8, Issue 3, pp. 681 - 698, DOI: 10.1109/JSYST.2012.2212636, ISSN: 1932-8184.

[10] L. Yong and L. Xia, "Safety Driving Decision-Making of the AVCSS," IEEE Transactions on Intelligent Computation Technology and Automation, Hunan, China, October 2008, pp. 477 - 481, DOI: 10.1109/ICICTA.2008.171, ISBN: 978-0-7695-3357-5.

[11] M. Horwick and K. Siedersberger, "Strategy and architecture of a safety concept for fully automatic and autonomous driving assistance systems," IEEE Intelligent Vehicles Symposium, California, USA, June 2010, pp. 955 - 960, DOI: 10.1109/IVS.2010.5548115, ISBN: 1931-0587.

[12] L. Genxian, W. Dongsheng, W. Haixia, and L. Zhenyu, "Image Processing Memory Optimization for Multi-camera Based Advanced Driver Assistance Systems," IEEE Transactions on Measuring Technology and Mechatronics Automation, Zhangjiajie, China, January 2014, pp. 313 - 318, DOI: 10.1109/ICMTMA.2014.78, ISBN: 978-1-4799-3434-8.

[13] dSPACE AutomationDesk, Real-Time Kernel (RTK) Real-Time Kernel (RTK) on Models (ASM), DS2211, dSPACE Catalog, dSPACE GmbH, 2009.

[14] D. Block, S. Heeren, S. Kühnel, A. Leschke, and B. Rumpe, "Simulations on Consumer Tests: A Perspective for Driver Assistance Systems," In Proceedings of Engineering Simulations for Cyber-Physical Systems Conference (ES4CPS), Dresden, Germany, March 2014.

[15] S. Ziegler and R. Höpler, "Extending the IPG CarMaker by FMI Compliant Units," In Proceedings of 8th International Modelica Conference, Dresden, Germany, March 2011.

[16] B. Lacroix, P. Mathieu, and A. Kemeny, "A Normative Model for Behavioral Differentiation," In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Sydney, Australia, 2008, pp. 96-99.

[17] A. Gloria, F. Bellotti, R. Berta, and E. Lavagnino, "Serious Games for Education and Training," International Journal of Serious Games, Vol. 1, No. 1, 2014, pp. 100-105, ISSN: 2384-8766.

[18] L. Rui, D. Burschka, and G. Hirzinger, "Real time landscape modelling and visualization," IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, July 2007, pp. 1820-1823, DOI: 10.1109/IGARSS.2007.4423175, ISBN: 978-1-4244-1211-2.

[19] J. Berssenbrügge, J. Stöcklein, A. Andre and I. Gräßler, "Procedural Generation of Vegetation for a Virtual Test Track," In Proceedings of the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference ASME 2014, Buffalo, NY, USA, August 2014.

[20] H. True, "The dynamics of vehicles on road and on tracks," Swets & Zeitlinger B.V., Lisse, Netherlands, vol. 37, April 2003, pp. 96–105.

[21] R. N. Jazar, "Vehicle dynamics: Theory and application," Springer Science+Business Media, LCC, New York, USA, 2008, pp. 37-279, e-ISBN 978-0-387-74244-1.

[22] J. Barcelo, "Fundamental of traffic simulation," Springer Science+Business Media, LCC, New York, USA, 2008, pp. 15-63, ISSN: 0884-8289, e-ISBN: 978-1-4419-6142-6.

[23] C. Macadam, "Understanding and modeling the human driver," Journal of Vehicle System Dynamics 49, vol. 40, nos. 1-3, 2003, pp. 101-134.

[24] D. T. Mcruer, R. W. Allen, D. H. Weir, and R. H. Klein, "New results in driver steering control models," Journal of Human Factors and Ergonomics Society, 19(4), SAGE Publications, California, USA, August 1977, pp. 381-397, DOI: 10.1177/001872087701900406.

[25] G. A. Bekey, G. O. Burnham, and J. Seo, "Control Theoretic Models of Human Drivers in Car Following," Journal of Human Factors and Ergonomics Society, 19(4), SAGE Publications, California, USA, August 1977, pp. 399-413, DOI: 10.1177/001872087701900406.

[26] R. Altendorfer, S. Wirkert, and S. Heinrichs-Bartscher, "Sensor Fusion as an Enabling Technology for Safety-critical Driver Assistance Systems," SAE International Journal of Passenger Cars - Electronic and Electrical Systems, October

2010, SAE International, USA, 2010, pp. 183-192, ISSN 0148-7191.

[27] T. Akenine-Möller, "Fast 3D triangle-box overlap testing," Journal of Graphics Tools archive, Vol. 6, No. 2, September 2001, pp. 29-33.

[28] N. Benalie, W. Pananurak, S. Thanok, and M. Parnichkun "Improvement of Adaptive Cruise Control System based on Speed Characteristics and Time Headway," IEEE/RSJ International Conference on Intelligent Robots and Systems, Missouri, USA, October 2009, pp. 2403-2408.

[29] J. E. Slotine and W. Li, "Applied nonlinear control," Prentice Hall Englewood Cliffs, New Jersey, USA, ISBN: 0-13-040890-5, 1991, pp. 276–307.

[30] V. V. Sivaji and M. Sailaja, "Adaptive Cruise Control Systems for Vehicle Modeling Using Stop and Go Manoeuvres," International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, vol. 3, Issue 4, July 2013, pp.2453-2456.

[31] C. Poussot-Vassala, O. Senameb, L. Dugardb, and S. M. Savaresic, "Anti-windup Schemes for Proportional Integral and Proportional Resonant Controller," In Proceedings of Power Electronics Conference, Roorkee, India, June 2010.

[32] K. Yi, Y. Cho, S. Lee, J. Lee, and N. Ryoo, "A Throttle/Brake Control Law for Vehicle Intelligent Cruise Control," In Proceedings of Seoul 2000 FISITA World Automotive Congress, Seoul, Korea, June 2000.

[33] C. Poussot-Vassala, O. Senameb, L. Dugardb, and S. M. Savaresic, "Vehicle Dynamic Stability Improvements Through Gain-Scheduled Steering and Braking Control," Journal of Vehicle System Dynamics 49, vol. 00, no. 00, January 2009, pp. 1597-1621, DOI: 10.1080/00423114.2010.527995.

[34] M. Abdelkarim, T. Butz, and A. Moutchiho, "A nonlinear path following controller for lateral vehicle guidance - Ein nichtlinearer Bahnfolgeregler zur Fahrzeugquerführung," Fahrermodellierung in Wissenschaft und Wirtschaft, Fortschritt-Berichte VDI, vol. 22, no. 35, VDI Verlag, Düsseldorf, Germany, June 2013, pp. 135-145, ISBN: 978-3-18-303522-9.

[35] A. Eckert, B. Hartmann, M. Sevenich, and P. Rieth, "Emergency Steer & Brake Assist – A Systematic Approach for System Integration of two Complementary Driver Assistance Systems," Continental AG. Germany: Paper Nr. 11-0111.

[36] C. Keller, T. Dang, H. Fritz , A. Joos, and C. Rabe, "Active Pedestrian Safety by Automatic Braking and Evasive Steering," IEEE Transactions on Intelligent Transportation Systems, December 2011, Vol. 12, Issue. 4, pp. 1292 - 1304, DOI: 10.1109/TITS.2011.2158424, ISSN: 1524-9050.

[37] Z. Mao, X. Yan, H. Zhang, and C. Wu, "Driving Simulator Validation for Drivers' Speed Behavior," In Proceedings of the Second International Conference on Transportation Engineering, Chengdu, China, July 25-27, 2009, pp. 2887-2892, ISBN: 9780784410394.

[38] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann, "Car2X Communication: Securing the Last Meter - A Cost-Effective Approach for Ensuring Trust in Car2X Applications Using In-Vehicle Symmetric Cryptography," In Proceedings of IEEE Vehicular Technology Conference, San Francisco, USA, Septemper 2011, pp. 1-5, DOI: 10.1109/VETECF.2011.6093081, ISSN: 1090-3038.

[39] S. Boskovich, K. Boriboonsomsin, and M. Barth, "A developmental framework towards dynamic incident rerouting using vehicle-to-vehicle communication and multi-agent systems," In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, September 2010, pp. 789-794, DOI: 10.1109/ITSC.2010.5625251, ISSN: 2153-0009.

[40] Y. Takatori and H. Yashima, "Performance evaluation of vehicle cooperative driving assistance systems that uses forward obstruction detecting sensors and inver-vehicle communication," In Proceedings of IEEE 9th International Conference on Intelligent Transport Systems Telecommunications, Lille, France, October 2009, pp. 622 - 627, DOI: 10.1109/ITST.2009.5399280, ISBN: 978-1-4244-5346-7.

[41] J. Fischer, A. Menon, A. Gorjestani, C. Shankwitz, and M. Donath, "Range sensor evaluation for use in Cooperative Intersection Collision Avoidance Systems," In Proceedings of IEEE Vehicular Networking Conference, Tokyo, Japan, October 2009, pp. 1-8, DOI: 10.1109/VNC.2009.5416389, ISBN: 978-1-4244-5685-7.

[42] Q. Wang and C. Phillips, "Cooperative collision avoidance for multi-vehicle systems using reinforcement learning," In Proceedings of IEEE 18th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, August 2013, pp. 98-102, DOI: 10.1109/MMAR.2013.6669888, ISBN: 978-1-4673-5506-3.

[43] T. Bando and T. Shibata, "Networked driving simulator based on SIGVerse and lane-change analysis according to frequency of driving," In Proceedings of 15th International IEEE Conference on Intelligent Transportation Systems, Alaska, USA, September 2012, pp. 1608-1613, DOI: 10.1109/ITSC.2012.6338804, ISSN: 2153-0009.

[44] Y. Zhao, A. Wagh, K. Hulme, C. Qiao, and W. Sadek, "Integrated Traffic-Driving-Networking Simulator: A Unique R&D Tool for Connected Vehicles," In Proceedings of International Conference on Connected Vehicles and Expo, Beijing, China, December 2012, pp. 203-204, DOI: 10.1109/ICCVE.2012.45, ISBN: 978-1-4673-4705-1.

[45] I. Tacic and M. Fujimoto, "Synchronized data distribution management in distributed simulations," In Proceedings of the 12th IEEE Workshop on Parallel and Distributed Simulation, Banff, Canada, May 1998, pp. 108-115, DOI: 10.1109/PADS.1998.685276, ISBN: 0-8186-8457-7.

[46] P. Galluscio, J. Douglass, A. Malloy, and A. Turner, "A comparison of two methods for advancing time in parallel discrete event simulation," In Proceedings of the IEEE Simulation Conference, Arlington, USA, December 1995, pp. 650-657, DOI: 10.1109/WSC.1995.478840, ISBN: 0-78033018-8.

[47] W. Xuehui , Z. Lei, X. Nong, and T. Yuhua, "Time Management in Parallel Discrete Event Simulation," In Proceedings of the IEEE International Forum on Information Technology and Applications, Chengdu, China, May 2009, pp. 209-212, DOI: 10.1109/IFITA.2009.96, ISBN: 978-0-7695-3600-2.