A Scalable Framework for Advanced Driver Assistance Systems Simulation

Kareem Abdelgawad, Mohamed Abdelkarim, Bassem Hassan, Michael Grafe, and Iris Gräßler
Heinz Nixdorf Institute
University of Paderborn, Germany
{Kareem.Abdelgawad, Mohamed.Abdelkarim, Bassem.Hassan, Michael.Grafe, Iris.Graessler}@hni.uni-paderborn.de

***Abstract*—Advanced Driver Assistance Systems (ADAS) are mechatronic vehicle systems that contribute to improving road safety and increasing driving comfort. Apart from all technical challenges regarding control algorithms and sensor quality, customer acceptance of ADAS is an important concern to automobile manufacturers. Demonstrating new ADAS to customers in real traffic environments is impractical and leads to significant efforts and costs. This paper presents the structure of a scalable framework used to implement ADAS virtual prototypes. The design approach ensures maximum flexibility and scalability for integrating new ADAS functions. The framework is composed of modular functional units that enclose real-time capable simulation models developed with MATLAB/Simulink. The design of the functional units and the input-output relationships of their models are presented. Prototypical implementation of two innovative ADAS is presented to show the usability and validity of the framework for ADAS demonstration and training purposes. The developed framework was integrated in an existing PC-based driving simulator used to interactively demonstrate ADAS by means of a simulated traffic environment.**

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

I. INTRODUCTION

Improving road safety standards is one of the main concerns in the automotive industry. Advanced Driver Assistance Systems (ADAS) are complex mechatronic vehicle systems that monitor vehicle surroundings, as well as driving behavior. 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 [1].

By utilizing diverse sensor technologies (camera, radar, ultrasonic, etc.) and decision algorithms, different levels of assistance are achieved [2]. On the one hand, some ADAS, like, e.g., Lane Departure Warning [3], 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 [4], which applies full braking if driver fails to respond to obstacles in front of the vehicle.

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. Firstly, a flexible test environment is required in order to validate ADAS concepts and assess their decision logic. Secondly, a clear concept for driver-vehicle interface has 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 [5]. For demonstration and training purposes, they can be utilized to make drivers familiar with new ADAS, and hence, accelerate the learning phase.

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 [6]. It 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.

One objective of the project TRAFFIS is the development of a platform for ADAS demonstration and training purposes. As shown in Figure 1, 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 incorporates a complex motion platform and a surround projection system. The motion platform can fully simulate vehicle lateral and longitudinal accelerations. Moreover, real vehicle cabins can be used, so that drivers experience realistic control cues. The TRAFFIS-Portable variant has a pneumatic motion platform and a four-wall projection system. The TRAFFIS-Light variant is simple a PC-based driving simulator with no motion platform. These driving simulator variants along with an innovative configurability concept offer a flexible test and training environment for various in-vehicle systems [6]. However, the focus is given mainly to the development of ADAS.

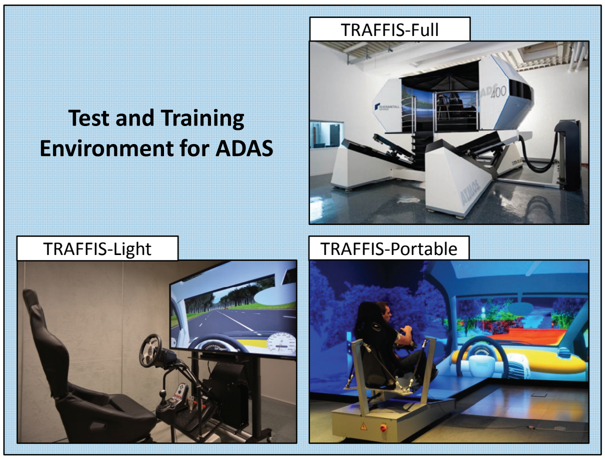

Figure 1. Driving simulators with different complexity levels.

A simulation framework with flexible prototyping concepts is required for easy and convenient ADAS demonstration and training. This paper presents the structure of a virtual prototyping framework for implementing ADAS and demonstrating their benefits with driving simulators. The developed 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. 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.

As a first step, the developed ADAS simulation framework was integrated with an existing simulation environment of the PC-based driving simulator, i.e., TRAFFIS-Light, 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 developed framework for ADAS demonstration and training purposes.

This paper is structured as follows: Section 2 shows briefly the driving simulation environment with which the developed framework was integrated. Section 3 presents the design approach of the developed ADAS virtual prototyping framework along with the concepts of its functional units and models. Section 4 demonstrates two ADAS prototypes realised with the developed framework and demonstrated using the PC-based driving simulator. Finally, Section 5 derives the conclusion and summarizes the benefits of the presented approach.

II. DRIVING SIMULATION ENVIRONMENT

The simulation environment of the PC-based driving simulator consists of two main functional units: a vehicle dynamics model and a traffic model. Figure 2 illustrates its structure and the direction of information flow.

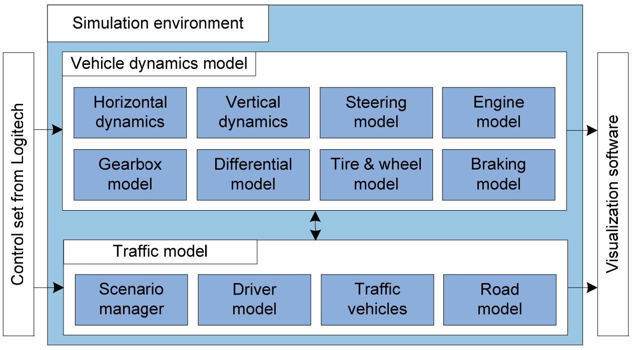

Figure 2. Simulation environment of the PC-based simulator.

Each functional unit consists of real-time capable sub-models implemented with MATLAB/Simulink. The visualization software represents the main feedback cue of the PC-based 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 [7]; a development engine that provides rich and easy functionalities for creating interactive 3D tools. The PC-based driving simulator incorporates a racing wheel-transmission-pedals set from Logitech and a racing seat from Speedmaster. It is still fully interactive with respect to steering, gears, acceleration and brake controls. This simulator variant and its simulation environment are considered in this work. The next sub-sections discuss each functional unit briefly.

A. Vehicle dynamics model

Modeling realistic vehicle dynamics is essential for the development of different in-vehicle systems. 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) [8]. The chassis has three translational and three rotational motions [9]. 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 [9]. It implements the blocks shown in Figure 2 as modular Simulink subsystems.

B. *Traffic model*

The traffic model is used to simulate the surrounding vehicles and the road [10]. 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, which composes arbitrarily different traffic situations. 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.

III. 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 [11]. The design approach of the developed ADAS simulation framework is based on an analogy between human driving behavior and the functionality of ADAS. Figure 3 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 [12] and [13]. 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.

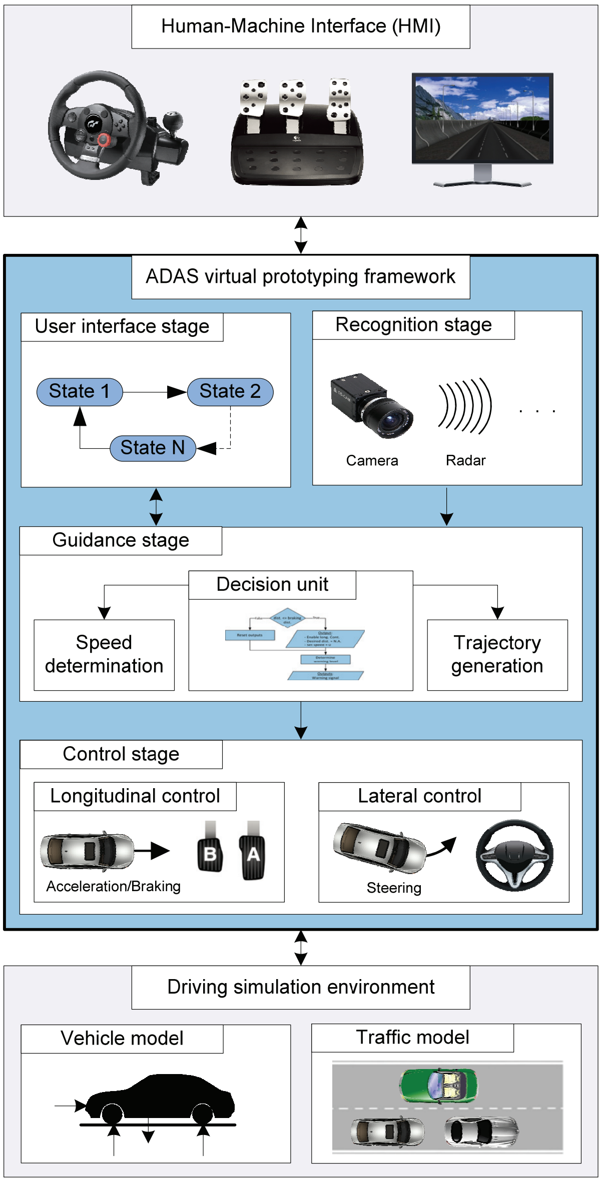

Figure 3. ADAS framework and its relation with the driving simulation environment and HMI.

As shown in Figure 3, 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 PC-based 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 4 shows the structure of the user interface stage and the main input-output signals.

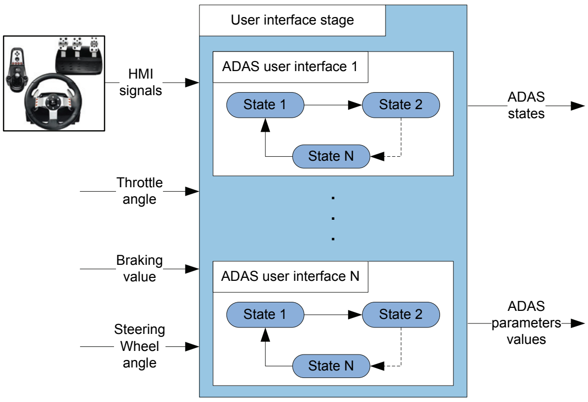

Figure 4. ADAS user interface stage.

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 [14]. 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 5 shows the structure of the recognition stage and the essential input-output signals.

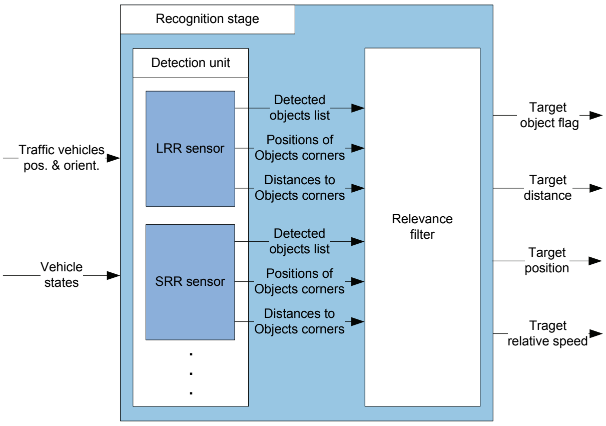

Figure 5. ADAS recognition stage.

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 [15]. 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 6 illustrates the selection functionality of the relevance filter unit.

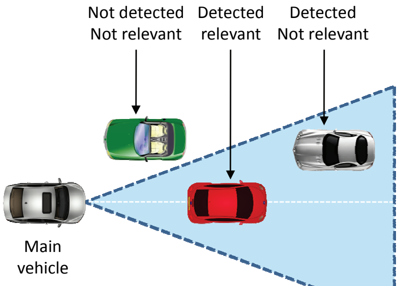

Figure 6. Target object selection of the relevance filter unit.

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. 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. Figure 7 shows the structure of the guidance stage and the main input-output signals.

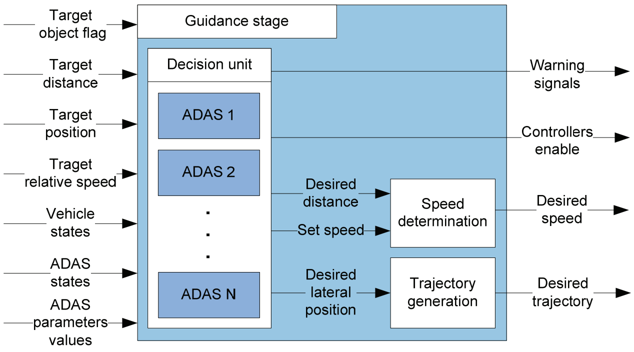

Figure 7. ADAS guidance stage.

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 8 shows a flow chart for the main function of the decision unit.

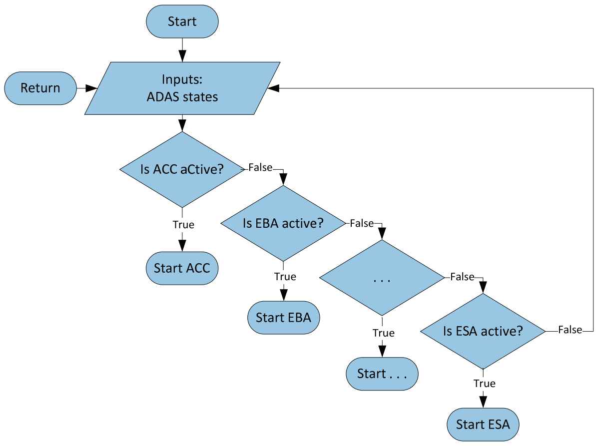

Figure 8. Transition logic between ADAS sub-routines.

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 [16]. 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 [17]. 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 9 shows the difference between the desired and actual headway distances.

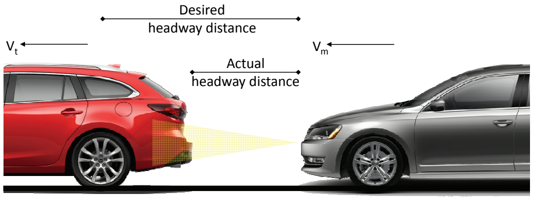

Figure 9. Headway distance control and speed determination.

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 10.

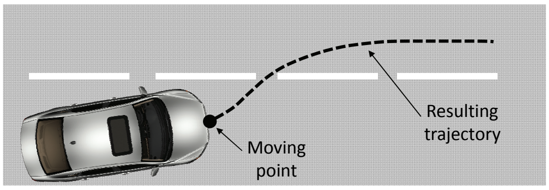

Figure 10. Moving point for trajectory generation.

The desired trajectory represented as postion updates is forwarded 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 11, decoupled longitudinal and lateral controllers were implemented to execute the orders of the guidance stage and guide the vehicle accordingly.

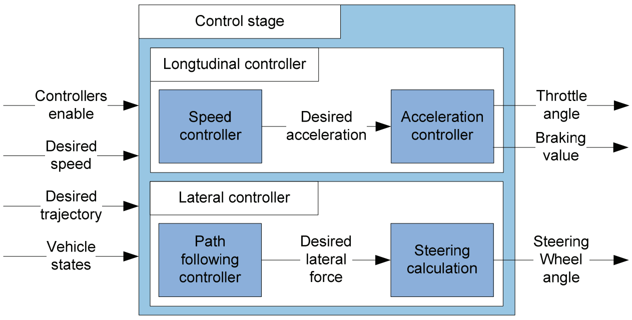

Figure 11. ADAS control stage.

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 [18]. 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 [19]. 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 [20]. The acceleration controller is composed mainly of three sub-models, as shown in Figure 12.

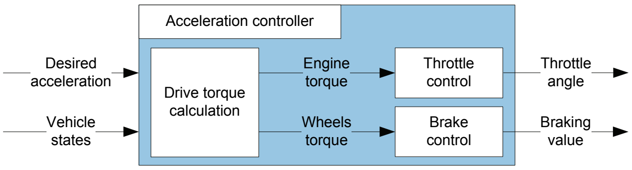

Figure 12. Sub-models of the acceleration controller.

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 [21]. 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 11, 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 [22]. 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 [8] 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.

IV. ADAS PROTOTYPICAL IMPLEMENTATION

To prove the usability of the developed ADAS virtual prototyping framework 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. Both functions represent the state of the art in ADAS development and have different types of intervention. However, 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. According to recognized moving or standing objects in front of the vehicle, it initiates automatic braking in the case of a potential rear-end collision provided that the driver has not responded to prior warnings signals [23]. Figure 13 shows a flow chart for a simplified version of the EBA sub-routine.

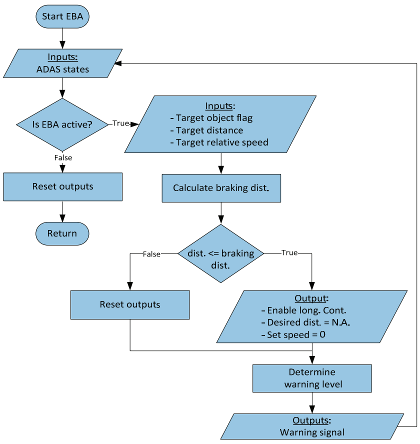

Figure 13. Simplified version of EBA logic within the decision unit.

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 14 illustrates the switching point between warnings and active intervention distances of the EBA sub-routine

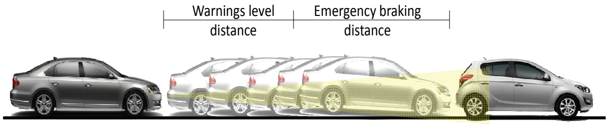

Figure 14. EBA intervention in case of no driver response.

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 [23]. 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 15.

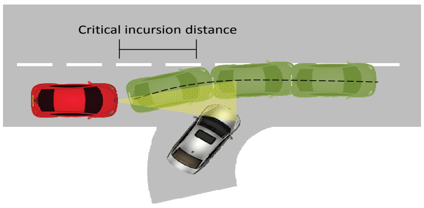

Figure 15. ESA intervention due to sudden road incursion.

Almost similar to Emergency Brake Assist function, the intention of the driver is observed through the user interface stage. 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. 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.

V. 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 an existing real-time simulation environment of a PC-based 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 [24]. 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.

REFERENCES

[1] 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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.