ARTICLE INFO
Article ID: 12-05-04-0029
© 2022 The Authors
doi:10.4271/12-05-04-0029

Dynamic Vehicle-in-the-Loop: A Novel Method for Testing Automated Driving Functions

Maikol Funk Drechsler,[1] Varun Sharma,[1] Fabio Reway,[1] Christoph Schütz,[2] and Werner Huber[1]

[1]Technische Hochschule Ingolstadt, CARISSMA Institute of Automated Driving, Germany
[2]BMW AG, Germany

Abstract

In automated driving functions (ADF) testing, novel methods have been developed to allow the combination of hardware and simulation to ensure safety in usage even at an early stage of development. This article proposes an architecture to integrate an entire test vehicle—denominated Dynamic Vehicle-in-the-Loop (DynViL)—in a virtual environment. This approach enables the interaction of a real vehicle with virtual traffic participants. The vehicle is physically tested on an empty track, but connected to the CARLA simulator, in which virtual driving scenarios are created. The simulated environment is transmitted to the vehicle driving function which controls the real vehicle in reaction to the virtual objects perceived in simulation. Furthermore, the performance of the DynViL in different test scenarios is evaluated. The results show that the reproducibility of the tests with the DynViL is satisfactory. Furthermore, the results indicate that the deviation between simulation and DynViL variates according to the vehicle speed and the complexity of the scenario. Based on the performance of the DynViL in comparison to simulation, the DynViL can be implemented as a complementary test method to be added to the transition between hardware in the loop (HiL) and proving ground. In this test method, erratic or unexpected behavior generated by the driving function and controllers can be detected in the real vehicle dynamics in a risk-free manner.

History

Received: 31 Jul 2021
Revised: 23 Oct 2021
Accepted: 24 May 2022
e-Available: 23 Jun 2022

Keywords

Automated driving, Safety, Test, Track, Vehicle dynamics, Virtual, Driving functions, Automated emergency braking

Citation

Drechsler, M., Sharma, V., Reway, F., Schütz, C. et al., "Dynamic Vehicle-in-the-Loop: A Novel Method for Testing Automated Driving Functions," *SAE Int. J. of CAV* 5(4):367–380, 2022, doi:10.4271/12-05-04-0029.

ISSN: 2574-0741
e-ISSN: 2574-075X

This article is part of a Special Issue on Emerging Simulation Tools and Technologies for Testing and Evaluating Connected and Automated Vehicles: Part 1.

© 2022 The Authors. Published by SAE International. This Open Access article is published under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits distribution, and reproduction in any medium, provided that the original author(s) and the source are credited.



Introduction

The actual driving assistance systems available in the market correspond to SAE Levels 1 (L1) and 2 (L2) [1]. In this case, the driver is responsible for taking decisions while the system operates under specific conditions. The development and validation of these systems have been done with the support of scenario-based testing in controlled proving grounds. These tests use a soft vehicle and dummy targets to represent the traffic vehicles and Vulnerable Road Users (VRUs) around the vehicle under test [2, 3].

The use of soft targets for validating SAE L2 vehicles is effective because these functions are triggered under strict conditions. However, the implementation of more complex driving functions such as SAE L3 or higher also requires testing in more complex scenarios [4]. In this case, collaborative maneuver among multiple vehicles or preindicators for the pedestrian path, like the change of chest angle can be barely reproduced in real test tracks.

Different regulations such as the ISO 26262-1:2018 [5] and the ISO/PAS 21448 [6], as well as the automotive industry [7], emphasize the importance of simulation as a testing and assessment tool. According to the new assessment/test method for automated driving provided by the European Council [8], the use of virtual testing helps developers to understand the outcomes of the system under test when it is triggered by simulated models. This allows the automated vehicle to be efficiently tested in various complex and safety-critical scenarios, which are otherwise difficult to be realized on test tracks. However, simulation-assisted testing depends largely on the quality and comprehensiveness of the implemented models. In this case, the limited fidelity of simplified models generates a limited representation of the vehicle and/or the environment, while complex models require extensive parametrization and can present limitations to run in real time.

Vehicles equipped with highly Automated Driving Functions (ADF) need to have precise control over the vehicle movement, including comfortable accelerations and adequate reaction under safety-critical situations. This correlates with the vehicle dynamics and fine-tuning of decision-making and control algorithms. Therefore, faults can be only observed after their implementation in a real test vehicle. However, complex scenarios can only be performed on real test tracks with the help of dummies that do not depict reality entirely, especially concerning the movement patterns of pedestrians. To address this challenge the stimulation of a real vehicle by synthetic data can be implemented. This approach combines the high validity of real tests and the flexibility of the simulation. In the context of testing ADF, virtual traffic participants can be used to trigger the decision-making algorithms, while the control acts on the real automated vehicle. This enables the verification and validation of the system considering the real driving dynamics in response to a particular virtual driving scenario without any risk to equipment, vehicle, or people.

This work describes the architecture of the Dynamic Vehicle-in-the-Loop (DynViL). In this case, the real vehicle moves on an empty test track, while its position is reproduced in the virtual environment as presented in Figure 1. Data from the vehicle surroundings, captured in the virtual domain, are fed back to the control system of the real vehicle, which can react to the virtual scene. This novel approach uses a central simulation computer outside the vehicle while the vehicle commands and positioning are exchanged between the vehicle and the central simulation.

FIGURE 1 Virtual surroundings created in the simulation (top) and DynViL on the empty test track (bottom).



© The Authors

The main objective of this work is to evaluate the feasibility and the potential of the proposed test environment. This includes an analysis of the divergences between pure simulated testing and the proposed DynViL in different scenarios. Besides that, the repeatability of tests executed with the DynViL is investigated. The main contribution of this article includes the proof of the DynViL concept, in which distributed vehicles and the virtual environment can exchange data by wireless communication. Furthermore, acceptable repeatability of the proposed test method is demonstrated and the performance of the DynViL in comparison to SiL is investigated. The results contribute to a better understanding of the impact of vehicle dynamics on the testing of ADF. Besides that, the DynViL presents itself as a suitable method for testing ADF in scenarios where the non-modeled phenomena from the vehicle dynamics model can affect the performance of the driving functions.

This article is organized as follows: Initially, an overview of the related work is presented. The section methodology describes the DynViL and its evaluation method. After that, the results regarding the Wireless Local Area Network (WLAN) latency, repeatability of tests, and the performance comparison between DynViL and Software-in-the-Loop (SiL) in different scenarios and speeds are presented. The section discussions include an overview of the limitation, potential implementations, and further development of the DynViL. Finally, in the conclusion section, the contributions and conclusions of this work are presented.

Related Work

The V-model is a widely used methodology in the automotive industry for defining the entire product development cycle [9]. According to its guidelines, automotive technologies must be tested to verify and validate their functionalities to ensure the safety in use of individual components and integrated systems. In the initial development phase, SiL approach is applied, considering a purely virtual environment for the test and evaluation of the implemented algorithms [10]. SiL enables the earliest detection of faults and failures caused by the implemented code. It is an essential test phase before the code deployment in specific hardware. Furthermore, testing can be extended to also include the final hardware and sensors, such as cameras [11, 12] and radars [13]. This approach, denominated Hardware-in-the-Loop (HiL) combined hardware, software, and stimulation by synthetic data in laboratory test benches. This enables the test of real hardware components and integrated systems in an early stage of development [14]. However, classical HiL approaches still lack the real vehicle dynamics, which is simulated by mathematical models. Depending on the fidelity of these models, the controller behavior and tune can present deviations between simulators and tests including the real vehicle.

The Vehicle-in-the-Loop (ViL) methodology includes the mechanical systems besides the electronic components in the test loop. In this case, real suspension, turning, and acceleration performance are integrated with the test, and simulated models of the vehicle are no longer required. Initial ViL tests were focused on human-machine interaction and the behavior of the driver during the development of driving assistance systems [15]. This research area is being further developed including "Virtual Reality" concepts in the architecture [16].

Because of the rise of automation in actual vehicles, ViL is implemented as a tool to evaluate the behavior of the complete automated system under controllable and reproducible virtual environments. The ViL methodology has benefits against the proving ground tests because of the higher reproducibility of the virtual scenarios, which can include complex target movements and larger quantities of participants in the scenario. Thus, ViL is evaluated as a possible methodology for the homologation of highly automated vehicles [17].

To permit the mechanical movement of the vehicle, some test benches couple the vehicle over rolls, permitting the evaluation of hybrid engines [18] and the prototyping and validation of Advanced Driver-Assistance Systems (ADAS) [19, 20]. Based on this method, the company AVL GmbH developed the DRIVINGCUBE™ test bench for validation and homologation of automated vehicles. In this approach, the whole car can be integrated on a roller test bench, and radar and cameras are stimulated by the synthetic data provided by the virtual environment [21]. A different approach was taken by Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (TNO; English: Netherlands Organisation for Applied Scientific Research) in the so-called Vehicle-Hardware-in-the-Loop (VeHiL) [22]. In this implementation, the vehicle is attached to a roller test bench and real targets are controlled by the simulation, moving around the vehicle under test (ego vehicle) to trigger the safety functions. Despite the easy adaptation of the platform to different vehicles, the roller test bench approach has a limitation on the vehicle dynamics since the vehicle is not moving, and the resistances are estimated by the simulation.

In other implementations, the ego vehicle is driven in an empty track and the position is sent to a virtual environment, which uses sensor models to feed the necessary perception data in the automated system [23]. Lately, this approach is finding similar implementations used by various companies such as IPG Automotive GmbH [24], and MORAI Inc./dSPACE GmbH [25]. Furthermore, academic projects have been implementing this approach to develop and test lane-keeping assistance and adaptive cruise control functions [26]. Similar methods were used to demonstrate the performance of driving functions at left turns in mixed-traffic areas [29], in roundabout navigation, collision detection [27, 28], and automated parking [30]. In most of these implementations, ideal or statistical sensor models are used to generate the necessary synthetic data from the virtual environment in the form of an object list. These sensor models are limited due to the physical simplifications and cannot be used to test the system including the perception units. This approach is, however, useful to evaluate the behavior of the vehicle regarding the outputs of the decision-making and control algorithms.

In most cases, the simulation runs on computers installed in the vehicle. Consequently, high computational power and space in the vehicle are required to process the virtual environment. The necessary performance and amount of processing units are even higher when the raw camera or lidar data [31] is used to feed the automated system. The challenges can be even more complex if electromagnetic stimulation of radar sensors is necessary, owing to the big space occupied by radar target simulators. For this reason, this approach is not suitable for modular testing of vehicles and is usually restricted to prebuild platforms. The use of distributed X-in-the-Loop testing can be implemented to overcome the limitations. In this approach, complementary test benches can be connected through the network, and the necessary data is exchanged. Different research studies evaluate the application of world-wide distributed and internet-connected test benches for applications such as anti-lock braking systems [32] and electric vehicle powertrain [33] testing. However, the interaction between test benches situated in different countries induces a high delay that needs to be compensated by prediction models.

Mixed-reality testing of ADF with central simulation outside the vehicle was implemented for evaluating an intersection management system, combining virtual traffic and a real vehicle [34]. In this case, instead of sending the positioning of other entities to the vehicle, the intersection management algorithm performs the decision-making and a request confirmation returns to the vehicle in case it can turn left at the intersection. Approaches including the interaction of several real traffic participants in a virtual environment were also implemented [35, 36]. In [35] the movement of a person is tracked and reproduced in the virtual

environment while a head-mounted display is used to show the virtual environment to the person. At the same time, an automated vehicle is connected to the virtual environment enabling a safe interaction between the vehicle and real pedestrian in a virtual environment. The architecture proposes possible combined testing with Pedestrian-in-the-Loop and ViL; however, for this implementation, dedicated instances of the simulator are indicated and roadside units are necessary to exchange the positioning data between the simulators. In other research [36], object lists are transmitted to the ViL and only the vehicle control occurs inside the vehicle. The data exchange uses 5G communications, and according to the authors, even then the latency between the systems is the bigger issue for the implementation. Because of the exchange of object lists, a complex environment with a large number of traffic participants can further increase this challenge.

Based on the actual limitations, the present research proposes and evaluates a novel DynViL method. In this system, the real vehicle moves on an empty track, but only the vehicle commands (accelerator and brake pedal positions and steering wheel angle) and vehicle positioning are exchanged between the vehicle and the virtual environment; thus, the necessary computational power remains outside the vehicle. This permits an efficient communication of the virtual environment with co-simulations and other traffic participants, without loss of the real behavior of the vehicle. This approach is designed to enable mixed-reality testing, in which other test benches can be integrated into the same environment, such as stimulation of real sensors and Human-in-the-Loop simulators. In the proposed method, the exchange of data between vehicle and simulation is independent of the number of participants, enabling the test of highly complex scenarios with efficient use of the wireless communication between vehicle and simulation. Lastly, the proposed architecture enables fast replacement of the ego vehicle as the computational resources do not need to be exchanged. In this context, DynViL can be used for research and development of automated driving systems, including improvement of sensor setups, tuning of controllers, evaluation of the algorithm performance, cooperative driving, and many others. In this research, the DynViL will be analyzed with simplified scenarios and sensor models to evaluate the repeatability of the method without the influence of perception units or complex scenarios. In a second step, four different scenarios with four different speeds are compared with pure simulated tests to understand the impact that the real vehicle dynamics has on the test results.

Methodology

This section describes the implemented SiL architecture as well as the specifications and architecture of the DynViL. Furthermore, the system under test is presented. An evaluation of the WLAN impact on the DynViL is performed, and finally, the methods for evaluating the repeatability and deviations between SiL and DynViL are explained.

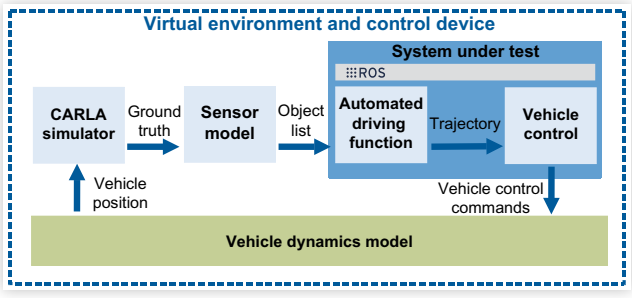
A. SiL Architecture

The SiL architecture is implemented to permit a modular exchange of the driving functions and easy development and detection of faults and failures in the implemented codes. In this way, a pure simulated test can be conducted to analyze the capability of the driving functions and to serve as a base for the evaluation of the DynViL performance. The architecture of the SiL is presented in Figure 2.

As presented in Figure 2, the virtual environment provides ground truth (GT) data to the sensor model, which processes the information from the virtual environment and sends it to the system under test. The system under test includes all functionalities of the ADF and the control algorithms. The ADF and control algorithms computed the necessary accelerator/brake pedal and steering angle values and make them available to the vehicle dynamics model. The vehicle model runs in co-simulation with the CARLA simulator processing the vehicle commands. The outputs from the vehicle model correspond to the measured data from a real car, including position, velocity, acceleration, and yaw angle. In the next subsections, the virtual environment, the implemented sensor model, and vehicle dynamics models are explained in more detail.

1. *Simulator:* The open-source CARLA simulator [37] is integrated within the architecture to provide the virtual environment and visualization. CARLA is used to move the virtual VRUs and to reproduce the movement of the ego vehicle in a virtual environment. The dynamic models of pedestrians and bicyclists available in the simulator are used to move the targets, and a trigger based on the ego vehicle position is used to initiate the target trajectory. CARLA runs asynchronously and parallelly to the system under test and a Python API is used to exchange the GT data and ego vehicle position.
2. *Sensor Model:* An ideal sensor model is implemented to generate a sensor-like object list. In this case, the GT data, including the position, absolute velocity, acceleration, classification, and size of all moving objects, are accessed from the virtual environment.

FIGURE 2 SiL architecture.



© The Authors

The sensor model rotates and transposes the GT from the map to the sensor frame, which is attached to the simulated vehicle. In addition, the positioning of the object corners is calculated and the entities outside the sensor field of view (FOV) are removed from the object list. As a simplification, the objects which contain at least one of the corners inside the FOV are considered as detected; otherwise, they are removed. The implemented sensor has a FOV of 120° and a range of 100 m, and it is fixed in the center of the vehicle.

3. *Vehicle Dynamics Model:* A nonlinear two-track dynamic vehicle model is implemented in MATLAB. In comparison to a kinematic model or a linear single-track model, this is a very comprehensive model that considers the individual tire characteristics by using the Magic Formula tire model [38]. The formula establishes the relationship between the normal force acting on each tire and the lateral and longitudinal forces $F_{y[1-4]}$, $F_{x[1-4]}$, using the sideslip angle $\alpha_{[1-4]}$, the steering angle δ, and the tire coefficients presented in Figure 3.

To design the model of the real vehicle, the required parameters such as the center of gravity, total mass, and dimensions of the vehicle are measured in the laboratory. However, because the accurate tire coefficients are difficult to be obtained, the values have been iteratively optimized for the ego vehicle. Finally, based on the input data provided by the vehicle controller, the model can update the velocity $v_{[1-4]}$ of each tire and, hence, the velocity v, the yaw rate ω, position, and orientation β of the vehicle through the computed forces using the Newton-Euler equations. A comprehensive mathematical description can be found in [39].

B. DynViL Requirements and Architecture

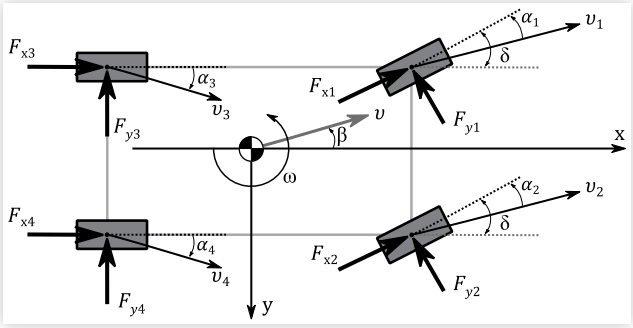
DynViL presents itself as a complementary test method, following the V-model development process, that lays in between the transition from HiL and the tests on the proving ground. The proposed implementation aims to enable rapid testing of highly ADF in complex environments mainly during the development phase. Because of that, the communication with multiple test benches as Human-in-the-Loop or sensor stimulation needs to be considered. Furthermore, considering the strict correlation between the ADF, vehicle control, and vehicle dynamics, high fidelity of the vehicle dynamics behavior needs to be maintained. Time-consuming parametrization of complex vehicle dynamics model needs, on the other hand, to be saved. Furthermore, the modularity of the components including the vehicle, hardware, and software under test needs to be addressed. The communication between the different test benches and the vehicle cannot be significantly affected by latency issues even if various traffic participants are included in the scenario.

Based on the defined requirements, the DynViL architecture includes a real vehicle that has its positioning reproduced in a virtual twin inside the simulated environment. Based on the interaction of the virtual vehicle and the virtual traffic participants, the real vehicle is controlled by the driving function. It guarantees that the real vehicle behavior is considered without the necessity of parametrizing the vehicle dynamics models. Furthermore, complex scenarios can be built in the virtual environment permitting the testing of highly ADF without any material or personal risk.

Differently from most of the ViL approached, the simulation and control systems run on a test bench outside the vehicle. In this approach, just the vehicle control commands (accelerator and brake pedal positions, and steering wheel angle) and the vehicle positioning are exchanged through wireless communication. In this way, complex scenarios can be run without affecting the amount of data exchanged between the real vehicle and the simulation. An overview of the implemented architecture is presented in Figure 4.

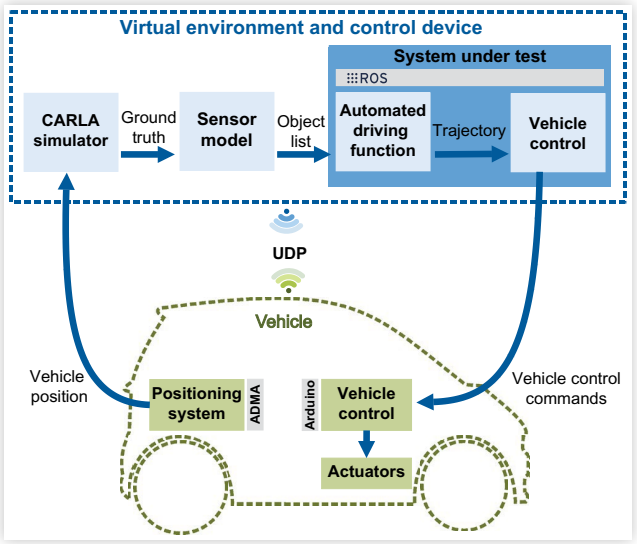
As shown in Figure 4, the virtual environment, sensor model, and system under test remain in the same setup as in the SiL architecture from Figure 2. In DynViL architecture, however, the vehicle dynamics model is replaced by the real

FIGURE 3 The two-track dynamic vehicle model.



© The Authors

FIGURE 4 DynViL architecture.



© The Authors

vehicle. In this approach, the ADF and control algorithms compute the necessary accelerator/brake pedal and steering angle values and send them by wireless communication to the ego vehicle, wherein the final maneuver is performed. A GNSS/IMU system is used to capture the vehicle movement, which is sent back to the virtual environment.

The test vehicle integrated within the DynViL architecture consists of an electric SMART Fortwo equipped with self-designed linear and rotational actuators that operate as driving robots. The actuators are powered by controller area network (CAN)-based stepper motors, which mechanically actuate on the original pedals and steering column. An Arduino board connected to the motors receives the WLAN information from the vehicle controller and converts the commands into respective CAN message and transmits them via a CAN I/O shield.

Furthermore, to make the control closed loop, the information about the vehicle position, orientation, velocity, and acceleration is obtained through the Automotive Dynamic Motion Analyzer (ADMA) over the WLAN as a UDP stream at 100 Hz. The ADMA receives the actual GPS position corrected by the Differential Global Navigation Satellite System with a precision of ±2 cm [40]. The sensor also includes an inertial measurement unit with an internal Kalman Filter for providing other vehicle parameters.

C. System under Test

A prototype driving function and controller are implemented in the Robot Operating System (ROS) framework [41]. This is the system under test that will be tested: first, in an SiL approach and, later, with the help of the DynViL test environment. In the next subsections, the ADF and the controller will be explained in more detail so that the understanding of the functions can support the identification of faulty behaviors in the results.

1. *Automated Driving Function:* A simplified ADF was developed and implemented in the architectures described in Figures 2 and 4. The function drives the car longitudinally on the desired path with the requested speed. The ADF generates a straight path-line, which starts at the initial position of the vehicle along with the vehicle orientation. The total length of the path-line to be driven in the scenario is taken by connecting the start point with a goal point at the end of the virtual map. At each time step, the actual vehicle position is measured and the nearest point on the initial path-line is calculated. From this point, the next 200 points and the respective timestamps are calculated based on the desired acceleration and speed profiles. These position and velocity points are sent at a frequency of 30 Hz to the vehicle controller. The acceleration and speed profiles are based on the maximum acceleration of the vehicle, set speed, and triggered decelerations of the ADF.

According to the data provided by the sensor model, the nearest object inside a 3.2-m-wide lane is defined as the target. The longitudinal distance *dist* is calculated between the center of the ego vehicle and the center of the target. Three different levels of deceleration and warning modes can be triggered according to the comparison between the module of the actual time to collision *TTC* and the necessary braking distance $time_{brake}$. The *TTC* can be calculated based on the longitudinal distance *dist*, the safety offset with a 6 m range, and the relative velocity between the ego vehicle and Target in the vehicle orientation vel_{rel} as shown in Equation 1.

$$TTC = \frac{dist - Offset}{vel_{rel}} \quad \text{Eq. (1)}$$

$$time_{brake} = \frac{speed_{ego}}{decel} + time_{react} \quad \text{Eq. (2)}$$

The calculation of the braking time $time_{brake}$ is realized by Equation 2, where $speed_{ego}$ is the speed of the ego vehicle and *decel* is the desired deceleration of each one of the decelerations stages, which are defined in Table 1. The $time_{react}$ is the reaction time of the system. In the case of automated braking and the driver reaction, their values are 0.5 s and 1.2 s, respectively.

The braking stage is triggered if $|TTC| < time_{brake}$ and $TTC < 0$, the second equation ensures that the relative velocity is negative and that the vehicle is approaching the target. Once triggered, the trajectory is updated to include the desired deceleration until the vehicle reaches the standstill. If the ego vehicle is next to the target (less than 6 m), the TTC is not calculated anymore. In this case, if the longitudinal velocity of the target is lower than 0.5 m/s the ego stops; otherwise, it follows the target at the same speed.

2. *Vehicle Controller:* The trajectory generated by the ADF is sent to the vehicle controller, which calculates the necessary steering wheel, brake pedal, and acceleration pedal positions for the ego vehicle. The vehicle commands are generated such that the vehicle can reach the desired location at the time instance requested in the trajectory.

The control algorithm consists of two parts, the longitudinal and the lateral controllers. The longitudinal controller is a PID implementation that controls the accelerator and brake pedal positions as a function of the vehicle speed error. This error is calculated based on the difference between the trajectory speed and the predicted speed at a certain anticipated time instance, which is estimated by a constant acceleration kinematic equation. The lateral controller acts on the steering angle position as a function of two types of errors: The yaw error is based on the deviation from the predicted yaw angle at the anticipated time instance regarding the yaw angle of the trajectory. The lateral deviation is based on the deviation of the vehicle laterally from its trajectory at the anticipated time instance. The implementation of this controller can be found in [39].

TABLE 1 Implemented deceleration.

Stage	3	2	1	Warning
Deceleration	9.81 m/s²	5.00 m/s²	3.00 m/s²	4.00 m/s²

© The Authors

D. Diagnosis of the WLAN Communication

In contrast to other ViL implementations, this research proposes a centralized simulation and control center which runs outside the vehicle. To understand the possible impact of this approach, the so-called roundtrip time (RTT) for the WLAN, complete DynViL system, and SiL implementation are evaluated. The RTT of the WLAN is calculated by the time required for a 64-byte package to be sent to the Arduino board in the vehicle plus the necessary time to return this package to the central computer. It is executed 12 times using the "ping" utility of the operating system Ubuntu 18.04. The RTT of the DynViL and SiL system considers the difference between the triggering of the function and the first deceleration of the vehicle. In this way, the complete system including WLAN and CAN communications, actuators, vehicle, and positioning sensors are evaluated in the DynViL method. In the SiL approach, the result includes the latency of the ROS messages along with the controller and vehicle model. To allow a more comprehensive evaluation, the final result includes an average and the standard deviation of the 12 executed tests.

E. Repeatability of the DynViL

During the testing of ADF, the repeatability of the test method is important to guarantee similar results at each test execution. To have a base of comparison for the repeatability of the DynViL, the same test is realized in SiL and DynViL test methods. The chosen scenario for the repeatability evaluation is an extended version of the Car-to-Car Rear Stationary (CCRS) from the Euro New Car Assessment Programme (NCAP) catalog presented in the upper left corner in Figure 5. The other test scenarios in Figure 5 are explained in Subsection F. The CCRS test simulates a stopped car (Target) in the same lane of the ego vehicle, between the start point and the final point of the calculated trajectory. To avoid a collision, the ego vehicle needs to perform an emergency maneuver. The test is repeated 12 times and is performed at 40 km/h. However, instead of evaluating only the collision avoidance capability, the DynViL is focused on the comprehensive tests of highly ADF, including the decision-making, trajectory planning, and trajectory following. This is further combined with the abilities of the implemented ADF to follow the computed trajectory automatedly, without any human interference. Thus, the complete acceleration and speed maintenance maneuvers are considered for evaluation.

Herein, the data of the vehicle position in X and Y directions, speed of the vehicle, and the status of the ADF are saved and compared among all tests. The collected data is evaluated from the initial acceleration until the vehicle reaches a standstill. If necessary, downsampling of the data is performed so that all tests have the same amount of data to be compared. To obtain the unbiased estimator for the standard deviation, s to all N *samples* at each timestamp t, the calculation is implemented separately to each variable r, according to Equation 3.

$$s_t = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(r_t - \overline{r}_t\right)^2} \qquad \text{Eq. (3)}$$

In Equation 3, $\overline{r}_t$ is the average of the variable r in all N samples of a specific timestamp t. To obtain an average deviation of each variable r during the complete testing time T, the average standard deviation $\overline{s}$ is calculated by Equation 4.

$$\overline{s} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} s_t^{\,2}} \qquad \text{Eq. (4)}$$

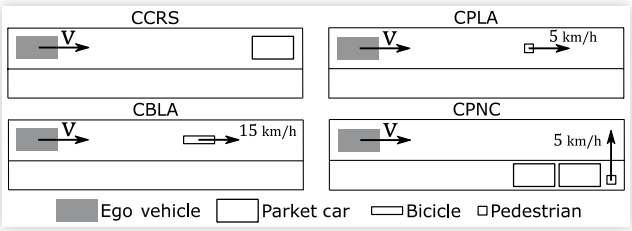
F. DynViL Performance in Different Test Scenarios

To understand the potential and limitations of the DynViL, the test method will be evaluated under four different scenarios. The same tests will be also conducted with the SiL architecture to have a basis for comparison. In these tests it is expected that the different challenges of each scenario permit to identify positive and negative points of the use of a real vehicle instead of the vehicle dynamics model. This evaluation is based on four scenarios of the Euro NCAP catalog for testing automated emergency braking functions. As in the repeatability test, the Euro NCAP scenario is extended so that the complete trajectory following can also be evaluated. The tests are executed with four different speeds for the ego vehicle (20, 30, 40, and 50 km/h). The scenarios are shown in Figure 5, including the CCRS explained previously. Besides CCRS the tests are executed in the following scenarios:

- **Car-to-Pedestrian Longitudinal Adult (CPLA):** An adult pedestrian moves longitudinally in the center of the lane at a speed of 5 km/h.
- **Car-to-Bicyclist Longitudinal Adult (CBLA):** An adult bicyclist moves longitudinally in the center of the lane at a speed of 15 km/h.
- **Car-to-Pedestrian Nearside Child (CPNC):** A child hidden by two parked cars runs into the street perpendicularly to the Ego vehicle movement at a speed of 5 km/h.

To compare, each specific combination of speed and scenario is conducted in SiL and DynViL methodologies, and

FIGURE 5 Implemented Euro NCAP scenarios.



© The Authors

the significant data are selected and downsampled if necessary. The collected variable r is compared at each time step t by a normalized root-mean-square deviation (NRMSD) as shown in Equation 5.

$$P_{NRMSD} = \frac{100\%}{r_{max}} \sqrt{\frac{\sum_{t=1}^{T} \left(r_t^{DynViL} - r_t^{SiL} \right)^2}{T}} \qquad \text{Eq. (5)}$$

In Equation 5, r_{max} is the maximum value that a variable r reaches in all realized tests and r_t^{DynViL} and r_t^{SiL} are values of the variable r on a specific timestamp t for DynViL and SiL, respectively. To permit a summarized evaluation of the results, the NRMSD obtained from Equation 5 are divided into two groups:

a. *Vehicle dynamics data:* X-position, Y-position, speed, and acceleration;
b. *ADF data:* ADF status, relative velocity, and distance between the Ego vehicle and the target.

An arithmetic mean of the NRMSD is calculated for the variables of each group, generating the vehicle dynamics gap and the ADF gap. The resulting values indicate the gap between SiL and DynViL in each test scenario, in a range from 0 to 100%. This permits a normalized comparison between the different variables and scenarios.

Results

This section presents the results obtained by the implementation of the methodology defined above. The results are divided into the evaluation of the WLAN latency, the DynViL repeatability, and the performance of SiL and DynViL in different scenarios.

A. Diagnosis of the WLAN Communication

The results obtained by the RTT measurements on the SiL, WLAN, and DynViL approaches are shown in Table 2.

The RTT from SiL and DynViL represents the time between sending the brake command and the first deceleration of the vehicle. The WLAN RTT includes the time to send and receive back a package with 64 bytes. The results of Table 2 indicate that the SiL reacts 14 times faster than the real vehicle. This is an expected result since the latency of the vehicle system is not modeled. Another point to be highlighted in the SiL result is the relatively high deviation on the result, represented by the 5 ms standard deviation, which is caused by the asynchronous implementation of the model.

TABLE 2 Normal distribution of the RTT.

System	SiL (ms)	WLAN (ms)	DynViL (ms)
Average	16	5	225
Standard deviation	5	6	27

© The Authors

Comparing the results of the WLAN and DynViL, it is observed that the WLAN latency represents approximately 2.3% of the complete system latency. This concludes that the onboard installation of the simulator, sensor models, and controllers does not improve the system performance since the latency of the actuators, sensors, and internal communications cannot be further reduced. However, the WLAN includes instability of the system observed by the high standard deviation of the WLAN RTT which causes a limitation to applications that require real-time responses. Considering the modularity benefits of using the virtual environment outside the vehicle and the evaluated results, the proposed implementation is feasible and can support the development and research of automated driving vehicles, along with its interaction with other traffic participants.

B. Repeatability of the DynViL

The repeatability of DynViL must be evaluated to ensure that test results generated after similar test runs are consistent. The positioning, speed, and the ADF status data obtained from the repetition of the DynViL and SiL tests for the CCRS scenario are shown in Figure 7. The ADF status corresponds to

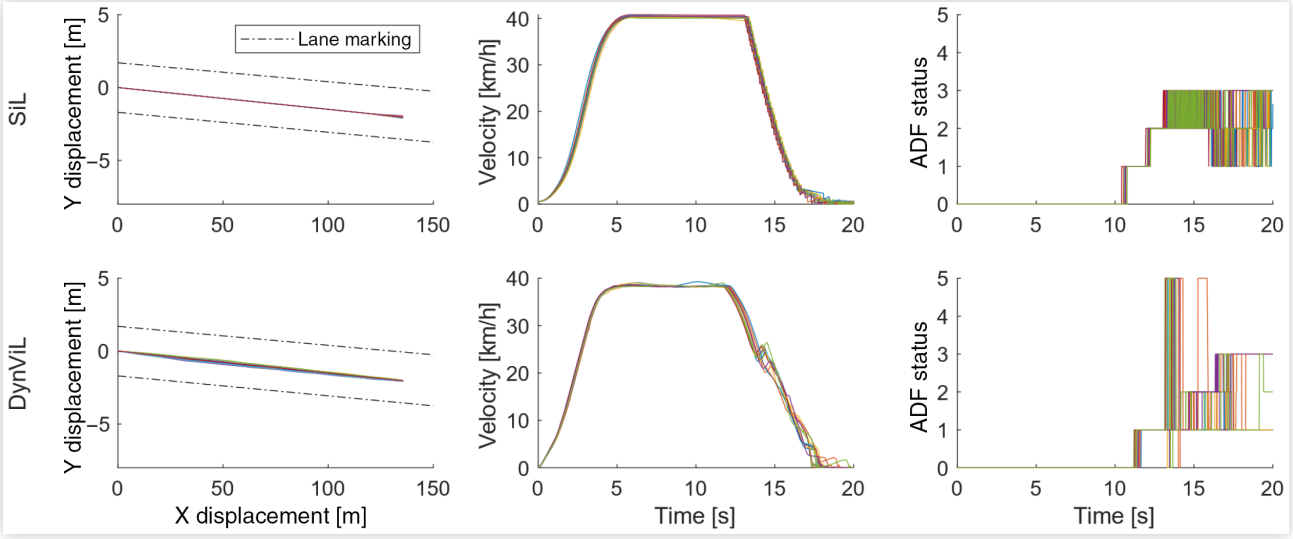
0: ADF **not** triggered.
1, 2, 3: Braking **stages** described in Table I;.
4: Ego vehicle is **following** the target.
5: Braking **warning**.

Analyzing the movement of the vehicle in the left graphs of Figure 6, the behavior in the SiL (bottom) corresponds to the expectations of the maneuver, in which the car drives completely straight. On the other hand, the results of the DynViL implementation present lateral deviations which are derived from physical phenomena that are not modeled, such as steering friction, complex tire-road interaction, motor and transmission losses, the lateral inclination of the street, dealignment of the suspension, and many others.

The speed data presented at the center of Figure 6 indicates a higher variation in SiL than in DynViL during the vehicle acceleration. This unexpected behavior is generated by the variance in the SiL RTT. For example, in case the vehicle model takes a long time to update the position, it may not match the desired trajectory and the controller can accelerate abnormally to synchronize the next position. This limitation can be improved by the synchronization between simulation and the vehicle model. In addition, a real-time operating system, as applied in HiL testing, can be used to run the vehicle model.

When evaluating the deceleration profile in the speed graph of the DynViL test, misbehaviors are identified at around 15 s with a minimal acceleration of the vehicle. The same behavior is not observed in the SiL test, which shows a completely smooth deceleration. In the ADF status of the

FIGURE 6 Repeatability of displacement, speed, and ADF status in 12 test executions of the CCRS scenario in SiL (top) and DynViL (bottom).



DynViL, at the right bottom of Figure 6, this misbehavior can be again observed at around 15 s, when the ADF status switches to "warning" mode and consequently permits the acceleration, generating the different behavior of the vehicle speed. The variation of the status occurs by a limitation on precise controlling the vehicle deceleration, due to which the vehicle brakes more than desired and the excessive reduction of the speed increases the TTC, bringing the ego vehicle to a safe condition ($|TTC| > time_{brake}$). Once the vehicle accelerates, the function is again triggered, and it brakes until a standstill. As the brake system in the vehicle model of the SiL method is composed of a linear correlation between the brake pedal position and the braking torque, the virtual vehicle requires a higher value of brake pedal input to reach a significant deceleration. For this reason, this misbehavior cannot be observed in the SiL approach. The correlation between distance and relative velocity additionally generates a high-frequency variation of the ADF status for both methods as observed on the right side of Figure 6. Furthermore, the DynViL does not reach the desired speed of 40 km/h, while in simulation, the vehicle achieves the desired speed. This behavior results in a different trigger time in each test method because the virtual vehicle drives faster and reaches the target earlier.

To achieve an overall comparison between SiL and DynViL, the average standard deviation of the evaluated variables is shown in Table 3. The results include the final position which was not evaluated in the previous graph; nonetheless, it represents an important variable to validate the test method.

TABLE 3 Average standard deviation in the repeatability tests.

Std. deviation	Pos X (m)	Pos Y (m)	Speed (km/h)	Status	Final pos (m)
SiL	0.60	0.03	0.55	0.35	0.01
DynViL	0.35	0.06	0.73	0.61	0.16

© The Authors

The results presented in Table 3 support the effects shown in Figure 6 in which the velocity profile of the SiL environment has a higher deviation than the DynViL. Due to that, the vehicle reaches a different X-positioning during the time in each repetition, and consequently, a more significant deviation is observed in Position X of the SiL test. The larger deviations in the DynViL in all other aspects are expected since in the SiL the vehicle is simplified by a physical model. In comparison to Position X, Position Y and speed deviations are relatively small. The standard deviation of the ADF status in DynViL is almost twice the deviation in the SiL method, which is largely contributed by the different braking capabilities of the real vehicle as compared to the virtual vehicle model. In the DynViL, 95% of the tests have an assertiveness better than 64 cm while, in SiL, 95% of the cases have an assertiveness of 4 cm at the final position of the ego vehicle. This behavior is associated with the different braking behavior of the real and virtual vehicles and with the simplified vehicle dynamics model used in SiL. In this case, when the vehicle is closer to the final braking position, the ADF maintains the actual vehicle speed until its standstill. This may lead to speeds slower than 5 km/h which cannot be managed by the controller on the real car, thus bringing the vehicle to a standstill before the expected point.

By an overall evaluation of the results shown in Figure 6 and Table 3, it is possible to observe that a controller perfectly tuned to the implemented simulated vehicle model can generate unexpected behaviors when implemented in the real vehicle, such as lateral deviation of the trajectory, undesirable change of the ADF trigger, and deviation from the desired speed. It highlights the necessary fine-tuning of the controller and driving functions with the final vehicle platform, which can be supported by the DynViL approach.

C. DynViL Performance in Different Test Scenarios

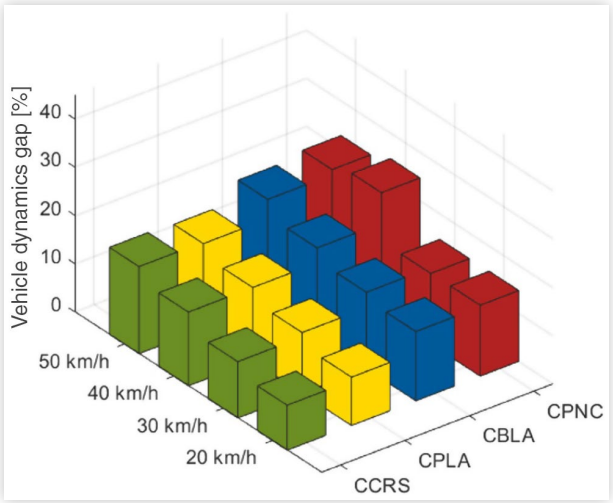
The comparison between SiL and DynViL in different scenarios aims to understand how DynViL performs under different test scenarios and how the complexity of the scenario can impact the DynViL performance. The average of the NRMSD of X-position, Y-position, speed, and acceleration is calculated and represents the vehicle dynamics gap shown in Figure 7.

The average of the NRMSD of the ADF status, relative velocity, and distance between the ego vehicle and the target are summarized in the driving function gap which is represented for different scenarios and speeds in Figure 8.

To simplify the understanding of the results from Figures 7 and 8, each one of the scenarios is evaluated separately in the next subsections and a final overview of the results is given after that.

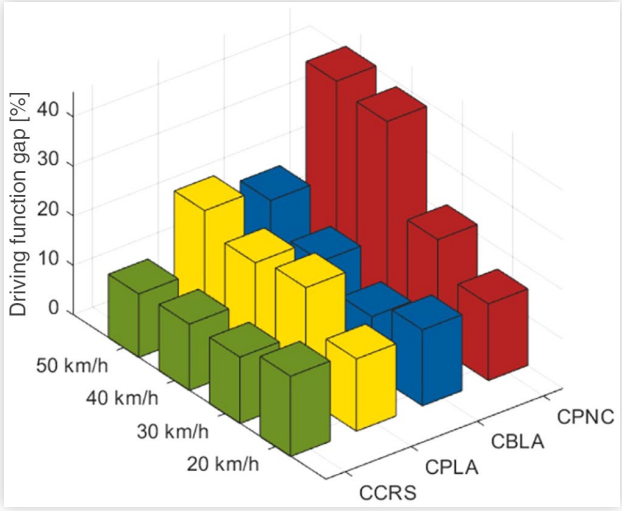
1. CCRS: The CCRS represents the simplest test in this comparison. The target vehicle stays stationary on the road while the ego vehicle approximates smoothly reducing the velocity. When analyzing the results of the CCRS test in Figure 7, it is possible to identify that the deviation between simulation and DynViL in the data correlated with the vehicle dynamics increases with the velocity. This behavior is explained by the simplifications of the vehicle model which are more evident in higher velocities. Furthermore, in Figure 8 the deviation between DynViL and SiL in the ADF-correlated variables is mainly constant in all velocities. The velocity of 20 km/h has a higher deviation than the other velocities. This is occasioned by a different maximal velocity in DynViL and SiL. In the 20 km/h test, the maximum speed is reached before the target is detected (at 100 m) and, consequently, a shift in all ADF-related variables generates the greater deviation.
2. CPLA: In this case, the vehicle dynamics gap increases with the velocity because of the same reasons as in CCRS. Additionally, the difference in the maximum speed between DynViL and SiL generates a different interaction between the moving target and the ego vehicle, increasing the ADF gap with the increment of the speed. This scenario represents a pedestrian who walks longitudinally at 5 km/h along the road and the limitation in maintaining the ego vehicle at this speed promotes a sinusoidal acceleration and deceleration in the DynViL. Consequently, the ADF status and triggering levels fluctuate in the DynViL testing generating greater deviations of the ADF gap for this scenario.
3. CBLA: Given that CPLA and CBLA are similar scenarios with the target in different velocities, the phenomena observed in CBLA are similar to those observed in CPLA. However, in CBLA the target moves at 15 km/h instead of the 5 km/h from CPLA, the restriction in maintaining the vehicle in lower velocities is not so predominant in CPLA. Due to that, the deviations in the ADF-related measurements in the CBLA scenario are smaller than the deviation presented in CPLA.
4. CPNC: The CPNC has a completely different approach than the previous scenarios. In this case, the vehicle has a very short reaction time to detect the pedestrian crossing the road and proceeding with an emergency braking. On the right side of the street are parked vehicles that hide the pedestrian. Because of the lateral deviations of the DynViL, as shown in the displacement graph of Figure 7, the parked vehicles are detected as a target. Mostly at 40 km/h and 50 km/h, the lateral movement in the DynViL causes the triggering of the ADF earlier, resulting in partial

FIGURE 7 Vehicle dynamics gap between SiL and DynViL in different speeds and scenarios.



© The Authors

FIGURE 8 ADF gap between Sil and DynViL in different speeds and scenarios.



© The Authors

braking of the vehicle at the beginning of the test. After a short deceleration, the vehicle aligns again and reaches the desired speed of the test. However, it causes the high gaps between SiL and DynViL in the vehicle dynamic-related data. Finally, in the CPNC the lateral deviation of the DynViL trajectory permits that the function defines the parked car as a target, generating the greatest disparity of the ADF-related variables in SiL and DynViL.

By a holistic evaluation of the results presented in Figure 7, it is possible to observe the influence of the vehicle speed on the gap between SiL and DynViL in all scenarios. This comportment probably originated from the limitation of the implemented vehicle dynamics model in reproducing complex phenomena, such as complex tire-road interaction, motor, transmission losses, and dealignment of the suspension. The effect of the non-modeled phenomena increases with the speed, consequently amplifying the deviation between the behavior of the real vehicle and the vehicle dynamics model.

Based on Figure 8, the scenario variation has a small influence on the ADF gap between SiL and DynViL at low speeds (20 km/h). However, with the speed increment, the behavior varies among the different scenarios, including an increase or decrease of the gap according to the scenario. More challenging scenarios for the control of the vehicle, such as CPLA or CPNC, show greater deviations between the implemented SiL and the DynViL.

Comparing the results from Figures 7 and 8, it is possible to identify that the response of the ADF is more affected by the change of the scenarios, while the vehicle dynamics gap is strongly correlated with velocity applied during the test. However, the straight correlation between the vehicle dynamics, controller, and driving function generates different results on the ADF side in case simplified vehicle dynamics models are used to represent the ego behavior.

Discussions

The applicability of a distributed approach to connect a real vehicle to a virtual environment is evaluated in this work. The proposed DynViL architecture maintains simulators as well as the control devices connected to a test bench outside the vehicle. It enables an easy exchange of the vehicle under test without the need to integrate simulation hardware on board. Besides that, the high fidelity of the vehicle dynamics is maintained without parameterizing complex vehicle dynamics models.

The distributed implementation of the DynViL enables efficient communication with other X-in-the-Loop systems as radar target simulators, or Human-in-the-Loop approaches, without a large exchange of data between the vehicle and the simulator. The DynViL architecture implemented in this research indicates that this approach can be applied for the testing of highly automated driving functions. Additionally, the latency resulting from the wireless exchange of the commands and positioning of the vehicle does not have a significant effect on the latency of the whole system, even if asynchronous co-simulation is implemented.

The repeatability of the test method guarantees that the systems under test will be submitted to the same conditions at each interaction of the test. It is especially important in case of comparison of different systems or to evaluate the improvement of the system along with the further development of the ADF. The implemented SiL environment shows minimal deviations between the tests, mainly derived from the asynchronous execution of the virtual environment and the system under test. The DynViL, on the other hand, has inferior repeatability resulting from a combination of factors such as uncontrollable noises in the vehicle system, latency of the internal system of the vehicle, and the variation of the latency, resulting from the WLAN connection. Better mitigation of this factor needs to be performed, including the comparison with real testing in which the control device is installed inside the vehicle. This approach will enable the identification of the deviation correlated with the DynViL implementation and the deviations correlated with the internal vehicle system and driving robots.

The vehicle dynamics model implemented in the SiL testing is a comprehensive two-track model with nonlinear tire models described by the Magic Formula. Despite that, the performance of the ADF during the DynViL testing includes misbehaviors such as slight deviation of the trajectory, inadequate fluctuations in the ADF status, and abrupt braking, which are not evident in the SiL testing. Further research will evaluate more comprehensive vehicle dynamics models, such as multibody systems. Furthermore, the DynViL will be used to evaluate the level of complexity necessary in the vehicle dynamics models so that ADF and vehicle control can be tested with enough precision in virtual environments.

The SiL and DynViL are compared under different test scenarios, and the results indicate that the deviation between these test methods varies according to the ego vehicle speed and the complexity of the scenario. With the increment of the ego vehicle speed, the variables related to the vehicle dynamics present a higher deviation between the DynViL and SiL. Furthermore, more challenging scenarios show greater deviations in the ADF-related variables. It indicates that in more complex and challenging scenarios, the vehicle dynamics model can be taken to the limit and non-modeled phenomena can have a significant influence on the results. In these cases, the DynViL can be used to include the real vehicle system in high-complexity scenarios without any risk to the participants. The SiL and HiL, on the other hand, accelerate the testing of a highly automated vehicle in comparison to DynViL. The necessary phenomena that need to be modeled so that the vehicle dynamics behave more realistically, maintaining real-time capabilities in SiL and HiL environments, also need to be investigated.

Based on the advantages and limitations investigated in this work, the concept of the DynViL is feasible and its implementation can take place between the HiL and the proving ground testing. Even the HiL test benches already in use can

be updated to exchange data with the real vehicle. In this way, the realistic vehicle behavior can be rapidly included in the evaluation. More benefits can be reached with the DynViL since safety-critical scenarios that are just limited implemented in real testing can be reproduced faster with this approach.

In future research, the authors aim to compare SiL, HiL, DynViL, and proving ground testing in the same scenario so that the limitation and concrete application of the DynViL can be further mitigated. Furthermore, a seamless chain to enable the easy transition between all test methods will be considered. The identification of the main phenomena that need to be included in the vehicle dynamics models used in SiL also needs to be investigated. Considering that the focus of the DynViL is the testing of the next generation of automated vehicles, highly automated driving functions need to be investigated in the future to evaluate the proposed architecture. In addition, the stimulation of the perception unit and the interaction with other traffic participants as Pedestrian-in-the-Loop or Driver-in-the-Loop will be considered in a Mixed-Reality Environment.

Conclusions

This work proposes a test method that includes the combination of a real vehicle and a simulated environment in a distributed fashion. The proposed DynViL architecture is described and implemented in this research article. Additionally, a proof of concept is performed. Aiming to evaluate the applicability of the method, the latency of the WLAN communication between the real vehicle and the virtual environment is investigated. Furthermore, the repeatability of this mixed-reality approach is compared with the SiL. Finally, the DynViL is subjected to four different Euro NCAP scenarios to comprehend how the performance of the SiL and the DynViL is affected by the complexity of the scenario.

The results of the research indicate that the DynViL is a suitable test method to be implemented between the HiL and the proving ground testing, following the V-model development process. Furthermore, the repeatability analysis indicates that the proposed implementation shows adequate repeatability, and higher deviations are correlated with the limited control of the braking system of the real vehicle. However, future comparison with real testing needs to be implemented to understand the effects of the vehicle system and the DynViL on the results.

By the comparison of the ADF performance in different scenarios and on both SiL and DynViL test methods, it is possible to identify deviations in the behavior of the implemented vehicle dynamics model and the real vehicle. The first observation indicates that the deviation between the DynViL and the SiL regarding the variables related to the vehicle dynamics increases with the speed of the ego vehicle. This behavior is derived from the limitation of the vehicle dynamics model in reproducing complex phenomena that are not modeled on the actual virtual twin. Furthermore, more challenging scenarios lead to greater deviations between simulation and DynViL. It suggests that tests in which unexpected maneuvers occur can bring the vehicle dynamics to their limits, and unmodelled phenomena can present a higher influence on the vehicle behavior. In these specific corner cases, the DynViL can facilitate the testing, including the real vehicle dynamics while the complex environment is generated by the virtual environment.

The possibility of triggering the ADF by simulated targets facilitates the testing under highly complex environments and safety-critical situations, which are hardly performed in real test tracks. Thus, the DynViL can potentially support the research and development of driving functions with SAE automation L3 and higher, which require full control of the vehicle in complex scenarios. Furthermore, the DynViL approach enables a gradual replacement of virtual models with real components, including the vehicle, sensors, and processing units. It allows the evaluation of each module limitation and facilitates the integration of a whole system during the development process. The distributed implementation of DynViL guarantees the integration of sensor stimulation and other X-in-the-Loop test benches in the same architecture. In this way, the vehicle sensor can be stimulated, and the larger amount of data traffic remains on the HiL testbench. Besides that, the interaction with other real traffic participants in a mixed-reality environment can be established. In this case, Human-in-the-Loop or Driver-in-the-Loop methods can be used to evaluate the interaction of the automated driving vehicle and a real pedestrian in the same virtual environment.

Future research on the DynViL will be concentrated on the validation of the DynViL performance against real track testing. This step will include the integration of a perception unit and sensor stimulation methods so that the same test can be performed in a real test track and mixed-reality test environment. Additionally, improvements in the vehicle dynamics model will be implemented to reduce the reality-simulation gap and to improve the real-time limitations of the virtual model. Furthermore, a Pedestrian-in-the-Loop will be integrated with the DynViL to evaluate the interaction between pedestrians and automated vehicles.

Acknowledgments

This work is supported under the FH-Impuls program of the German Federal Ministry of Education and Research (BMBF) under Grant No. 13FH7I01IA (SAFIR IP1) and Grant No. 13FH7I06IA (SAFIR IP6).

References

1. SAE International, "SAE J3016:202104—Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE International, Warrendale, PA, 2019.

2. Rigling, A., "Testing and Assessment of AEB Systems for Consumer Protection," in Pfeffer, P.E., (Ed.), *5th International Munich Chassis Symposium 2014* (Wiesbaden: Springer Fachmedien Wiesbaden, 2014), 623-632.
3. Lemmen, P., Stoll, J., Bergelt, U., Seiniger, P. et al., "Evaluation of Pedestrian Targets for Use in Autonomous Emergency Brake System Testing—A Report from the Harmonisation Platform 2 Dealing with Test Equipment," in *23rd Enhanced Safety of Vehicles Conference (ESV)*, Seoul, Republic of Korea, 2013.
4. Economic Commission for Europe (ECE), "Proposal for a New UN Regulation on Uniform Provisions Concerning the Approval of Vehicles With Regards to Automated Lane Keeping System," 2020.
5. ISO, "26262-1:2018—Road Vehicles—Functional Safety," International Organization for Standardization, Geneva, Switzerland, 2018.
6. International Organization for Standardization (ISO), "PAS 21448:2019 Road Vehicles—Safety of the Intended Functionality," 2019.
7. Aptiv, Audi, Baidu, BMW, Continental, Daimler, Fiat Chrysler Automobiles, HERE, Infineon, Intel and Volkswagen, "Safety First for Automated Driving," 2019.
8. Economic Commission for Europe (ECE), "New Assessment/ Test Method for Automated Driving," Geneva, Switzerland, 2021.
9. Winner, H., Hakuli, S., Lotz, F., and Singer, C., *Handbook of Driver Assistance Systems* (Cham: Springer International Publishing, 2016)
10. Jeong, S. and Lee, W.J., "An Automated Testing Method for AUTOSAR Software Components Based on SiL Simulation," in *Ninth International Conference on Ubiquitous and Future Networks*, Milan, Italy, 278-283, 2017.
11. Reway, F., Huber, W., and Ribeiro, E.P., "Test Methodology for Vision-Based Adas Algorithms with an Automotive Camera-in-the-Loop," in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Madrid, Spain, 2018.
12. Sievers, G., Seiger, C., Peperhowe, M., Krumm, H. et al., "Driving Simulation Technologies for Sensor Simulation in SIL and HIL Environments," in *Driving Simulation Conference Europe 2018 VR*, Antibes, France, 2018.
13. Kizilay, A. and Becker, A., "76-79 GHz Automotive Radar Test System Based on Radar Target Simulator," in Großmann, U. and Kunold, I., (Eds.), *Smart Energy and Systems* (Glückstadt: Verlag Werner Hülsbusch, 2019), 176-181.
14. Feilhauer, M., Haering, J., and Wyatt, S., "Current Approaches in HiL-Based ADAS Testing," *SAE Int. J. Commer. Veh.* 9, no. 2 (2016): 63-69, https://doi.org/10.4271/2016-01-8013.
15. Bock, T., Maurer, M., and Farber, A.G., "Validation of the Vehicle in the Loop (VIL); A Milestone for the Simulation of Driver Assistance Systems," in *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 612-617, 2007.
16. Karl, I., Berg, G., Rüger, F., and Färber, B., "Driving Behavior and Simulator Sickness While Driving the Vehicle in the Loop: Validation of Longitudinal Driving Behavior," *IEEE Intelligent Transportation Systems Magazine* 5 (2013): 42-57.
17. Riedmaier, S., Nesensohn, J., Gutenkunst, C., Duser, T. et al., "Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions," 2018.
18. Albers, A. and Düser, T., "Implementation of a Vehicle-in-the-Loop Development and Validation Platform," in *33. FISITA World Automotiv Congress*, Budapest, Hungary, 2010.
19. Galko, C., Rossi, R., and Savatier, X., "Vehicle-Hardware-in-the-Loop System for ADAS Prototyping and Validation," in *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV)*, Samos, Greece, 329-334, 2014.
20. Gao, Y., Xu, Z., Zhao, X., Wang, G. et al., "Hardware-in-the-Loop Simulation Platform for Autonomous Vehicle AEB Prototyping and Validation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020.
21. Siegl, S., Ratz, S., Düser, T., and Hettel, R., "Vehicle-in-the-Loop at Testbeds for ADAS/AD Validation," *ATZ Electronics Worldwide* 16, no. 7-8 (2021): 62-67.
22. Tideman, M. and Noort, M.V., "A Simulation Tool Suite for Developing Connected Vehicle Systems," in *IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Australia, 2013.
23. Zofka, M.R., Klemm, S., Kuhnt, F., Schamm, T. et al., "Testing and Validating High Level Components for Automated Driving: Simulation Framework for Traffic Scenarios," in *IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, 2016.
24. Überbacher, M., Wolze, P., and Burtsche, T., "Experiencing Safety Function Testing," *ATZ Worldwide* 119 (2017): 54-57.
25. dSPACE, "Augmented Reality," *dSPACE Magazine* 2 (2019): 14-15.
26. Solmaz, S., Rudigier, M., and Mischinger, M., "A Vehicle-in-the-Loop Methodology for Evaluating Automated Driving," in *IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, TX, 2020.
27. Chen, Y., Chen, S., Xiao, T., Zhang, S. et al., "Mixed Test Environment-Based Vehicle-in-the-Loop Validation—A New Testing Approach for Autonomous Vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, TX, 2020.
28. Szalai, M., Varga, B., Tettamanti, T., and Tihanyi, V., "Mixed Reality Test Environment for Autonomous Cars Using Unity 3D and SUMO," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herlany, Slovakia, 2020.
29. Varga, B., Szalai, M., Fehér, Á., Aradi, S. et al., "Mixed-Reality Automotive Testing with SENSORIS," *Periodica Polytechnica Transportation Engineering* 48, no. 4 (2020): 1-6.
30. Kneissl, M., Dorff, S.V., Molin, A., Denniel, M. et al., "Mixed-Reality Testing of Multi-Vehicle Coordination in an Automated Valet Parking Environment," *IFAC PapersOnLine* 53, no. 2 (2020): 17564-17571.
31. Zofka, M.R., Essinger, M., Fleck, T., Kohlhaas, R. et al., "The Sleepwalker Framework: Verification and Validation of

Autonomous Vehicles by Mixed Reality LiDAR Stimulation," in *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, Brisbane, Australia, 2018.

32. Schreiber, V., Ivanov, V., Augsburg, K., Noack, M. et al., "Shared and Distributed X-in-the-Loop Tests for Automotive Systems: Feasibility Study," *IEEE Access* 6 (2018): 4017-4026.
33. Ivanov, V., Augsburg, K., Bernad, C., Dhaens, M. et al., "Connected and Shared X-in-the-Loop Technologies for Electric Vehicle Design," in *the 32nd International Electric Vehicle Symposium*, Lyon, France, 2019.
34. Quinlan, M., Au, T.-C., Zhu, J., Stiurca, N. et al., "Bringing Simulation to Life: A Mixed Reality Autonomous Intersection," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.
35. Zofka, M.R., Ulbrich, S., Karl, D., Fleck, T. et al., "Traffic Participants in the Loop: A Mixed Reality-Based Interaction Testbed for the Verification and Validation of Autonomous Vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, 2018.
36. Varga, B., Tettamanti, T., and Szalay, Z., "System Architecture for Scenario-in-the-Loop Automotive Testing," *Transport and Telecommunication* 22, no. 2 (2021): 141-151.
37. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. et al., "CARLA: An Open Urban Driving Simulator," in *1st Conference on Robot Learning*, Mountain View, CA, 2017.
38. Pacejka, H.B. and Bakker, E., "The Magic Formula Tyre Model," *Vehicle System Dynamics* 21, no. 1 (1992): 1-18.
39. Botsch, M. and Utschick, W., *Fahrzeugsicherheit und automatisiertes Fahren: Methoden der Signalverarbeitung und des maschinellen Lernens* (Carl Hanser Verlag GmbH Co KG, München, 2020)
40. GeneSys Elektronik GmbH, "ADMA Family GPS/Inertial System Automotive/Railway," accessed July 27, 2021, https://www.genesys-offenburg.de/en/products/adma-family-gpsinertial-system-automotiverailway/.
41. Quigley, M., Conley, K., Gerkey, B., Faust, J. et al., "ROS: An Open-Source Robot Operating System," in *ICRA Workshop on Open Source Software*, Kobe, Japan, 2009.

© 2022 The Authors. Published by SAE International. This Open Access article is published under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits distribution, and reproduction in any medium, provided that the original author(s) and the source are credited.

Positions and opinions advanced in this work are those of the author(s) and not necessarily those of SAE International. Responsibility for the content of the work lies solely with the author(s).