

Article

Development and Functional Validation Method of the Scenario-in-the-Loop Simulation Control Model Using Co-Simulation Techniques

Balint Toth *  and Zsolt Szalay 

Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem Rakpart 3, 1111 Budapest, Hungary; szalay.zsolt@kjk.bme.hu

* Correspondence: tothb.0920@edu.bme.hu

Abstract: With the facilitated development of highly automated driving functions and automated vehicles, the need for advanced testing techniques also arose. With a near-infinite number of potential traffic scenarios, vehicles have to drive an increased number of test kilometers during development, which would be very difficult to achieve with currently utilized conventional testing methods. State-of-the-Art testing technologies such as Vehicle-in-the-Loop (ViL) or Scenario-in-the-Loop (SciL) can provide a long-term solution; however, validation of these complex systems should also be addressed. ViL and SciL technologies provide real-time control and measurement with multiple participants; however, they require enormous computational capacity and low-latency communication to provide comparable results with real-world testing. 5G (fifth-generation wireless) communication and Edge computing can aid in fulfilling these needs, although appropriate implementation should also be tested. In the current paper, a realized control model based on the SciL architecture was presented that was developed with real-world testing data and validated utilizing co-simulation and digital twin techniques. The model was established in Simcenter Prescan© connected to MATLAB Simulink® and validated using IPG CarMaker®, which was used to feed the simulation with the necessary input data to replace the real-world testing data. The aim of the current paper was to introduce steps of the development process, to present the results of the validation procedure, and to provide an outlook of potential future implementations into the state of the art in proving ground ecosystems.

Keywords: autonomous driving; automated driving functions; vehicle simulation; Scenario-in-the-Loop; Vehicle-in-the-Loop testing; validation; co-simulation; virtual testing



Citation: Toth, B.; Szalay, Z. Development and Functional Validation Method of the Scenario-in-the-Loop Simulation Control Model Using Co-Simulation Techniques. *Machines* **2023**, *11*, 1028. <https://doi.org/10.3390/machines11111028>

Academic Editor: Domenico Mundo

Received: 30 September 2023

Revised: 28 October 2023

Accepted: 8 November 2023

Published: 17 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recent advances in automated driving technologies provide increased mobility, decreased pollution, and the vision of zero accidents. Therefore, the interest of society has a significant role among the main drivers of the developments. In recent years, advanced research has shaped the technical development and safe deployment and evaluation of highly automated systems [1–3]. However, the complexity of tasks still provides substantial space for future improvement due to advanced functions that have to be performed in more complex driving conditions [4,5]. These driving conditions (e.g., overcrowded roads) can still provide near-unsolvable challenges for even the most advanced systems [6]. With the expected appearance of highly automated driving functions and vehicles, the research of suitable testing procedures became one of the biggest challenges in today's automotive testing. With conventional methods, covering the increasing number of potential testing scenarios would be highly challenging [7]. Hence, the role of advanced simulation-based testing solutions is becoming more important. Simulations can help to identify the most critical driving situations and speed up the evaluation of potential use cases. The simulation environment must also be suitable for the development of different algorithms to control

vehicles, different traffic participants, and infrastructural elements. This could provide instant testing and validation capabilities [8–10].

In the closed-loop testing methodologies, simulation also plays an indispensable role, and its advantages in test scenario generation are already proven [11]. In parallel to the classical Model-in-the-Loop (MiL) and Software-in-the-Loop (SiL) testing phenomenon, simulation can also be effectively used in real-time Hardware-in-the-Loop (HiL), Vehicle-in-the-Loop (ViL) [12], in the already mentioned Scenario-in-the-Loop (SciL) concept or even in Driver-in-the-Loop (DiL) especially in the context of driver behavior in autonomous vehicles [13]. ViL simulation environment could be effectively used to develop different advanced driver assistance systems (ADASs). Moreover, these solutions, especially combined with augmented reality technology, can play a significant role in simulation-driven validation [14,15]. Using computer simulation to achieve a significant part of the required test kilometers is already a common practice in the automotive industry. More and more public databases and simulations become available to reproduce and create more realistic driving scenarios [16]. With the increasing complexity of these driving scenarios, the need for co-simulation technologies also arose, providing the opportunity to utilize the main advantages of different tools combining them into one simulation framework [17–21]. Simulations can be also used in some specific cases for the homologation procedures and type approvals of road vehicles and also nominated as a key pillar in the on-going discussions of the related working groups of the EU and the UNECE [22–24]. In tight relation with autonomous vehicles, the connectivity of these vehicles, smart objects, and infrastructure elements brings the issues of cyber-security, and even the UNECE developed mandatory regulations to lower the risk of undesired manipulation [25,26]. Albeit simulations will have a key role in the homologation, the fidelity, and credibility of the results still need to be inspected. Thus, several existing validation methods need to be subjected to thorough research [27]. Various advanced simulation-based virtual or semi-virtual testing methodologies have been developed in the last years intended to reduce the necessary public ground testing, provide even open-source solutions for testing AI-based functions, and facilitate the validation procedures of autonomous vehicles [28]. Entire vehicles with all their subsystems can be tested on roller benches challenged by using virtual environment-based sensor spoofing or sensor simulation [29–31]. Mixed reality testing solutions with real-time connected but separated tested and target objects in space were also developed [32,33]. Digital twin technologies have started to be more common, and already, the classical proving ground-based testing approaches consider their usage [34]. Advanced methods may significantly reduce the necessary testing time on proving grounds. However, the role of proving ground testing cannot be completely replaced [35].

To overcome the limitations of current state-of-the-art methods, our group has proposed the so-called Scenario-in-the-Loop (SciL) framework, which is a novel proving ground-based simulation-supported testing approach [36]. The SciL was designed as a development and validation method for simulation- and control software [36]. With real-time connection between real and virtual testing environments, the SciL framework is capable of blurring the boundaries. However, this system has to be further investigated and developed with the use of real-world input data. Furthermore, if there is a lack of suitable recorded data, simulation tools can be applied to generate them.

Hence, the objective of the current investigation was to integrate Simcenter Prescan© into the SciL simulation framework and to present the combined application of IPG CarMaker® and Simcenter Prescan© as a co-simulation environment. Our additional aim was to present an example of a control algorithm utilizing the developed simulation environment based on a EuroNCAP AEB protocol. During our work, we realized that the following research gap can be filled using our development:

- Various simulation tools can be used for facilitating vehicle testing, but every tool has its own advantages. Using co-simulation techniques, we can exploit the advantages of the connected tools. To connect simulation tools to provide appropriate real-time connection and operation is always challenging. Hence, we hope that our presented so-

lution for the coupling of the two mentioned vehicle simulation tools can also be useful for researchers who decide to use these software solutions for similar approaches.

- Following the common methods, simulations are used before the real-world testing, but in our approach, we decided to use real-world measurement data to build a simulation framework to ensure the compatibility between the simulation and real-world testing. This helps to realize a more flexible simulation environment that can also interact with real-world test objects more easily in the later stages.

In this paper, following a brief introduction of the SciL framework used as a foundation of our applied methodology, the integration of an industrial simulation software (Simcenter Prescan©) as the core component of the SciL simulation framework was demonstrated. Contrary to other methods, in the current approach, the acquisition of real testing data and its injection into the simulation software are the first two essential parts of generating the appropriate simulation model environment.

Thereafter, an example of control algorithm development for test targets is presented utilizing the developed SciL simulation environment based on a EuroNCAP AEB protocol. As part of this process, a special co-simulation environment was realized by combining two automotive simulation software, IPG CarMaker® and Simcenter Prescan©, through MATLAB Simulink® to validate our virtual test environment with simulated data. In this co-simulation environment, Prescan© was used for SciL control algorithm implementation, and IPG CarMaker® was utilized to substitute the signals from the vehicle under test (VUT). As a result, and advantage of the co-simulation, we can evoke the movement of the VUT in Prescan© as a real road user and activate all the potential disturbances and obstacles that can influence the behavior of the VUT, thus creating a reproducible development environment facilitating the testing of advanced and highly automated vehicle functions, without conducting time-consuming and expensive proving ground testing.

2. Materials and Methods

2.1. The SciL Simulation Framework

The SciL concept is an advanced testing technique intended to be used in proving ground environments and was designed based on the classical ViL architecture. The ViL was originally developed to investigate the interaction between the driver and advanced driver assistant systems (ADASs) [37]. The introduction of the SciL concept is closely connected to the ZalaZONE proving ground (Zalaegerszeg, Hungary) and was conducted in cooperation with the Budapest University of Technology and Economics. The concept is suitable for closed-loop testing of entire vehicles in a fully controlled proving ground environment using realistic vehicle- and vulnerable road user (VRU) targets [36].

The architecture and operation principle of the concept are shown in Figure 1, which is an updated visualization of the SciL concept introduced for the next generation X-in-the-Loop validation methodologies [38]. The main element of the SciL concept is the ‘Simulation and Control Software’, which acquires localization data from the tested vehicle. In the most optimal case, the VUT can be handled as a ‘black box’ during testing. All necessary data from the VUT can be acquired by external sensors, which can be equipped into the vehicle or implemented into the environment or infrastructure [39]. Based on the acquired data, the software can activate the different disturbances and redefine the scenario in real time if necessary. The virtual environment presented in the software is a digitalized copy of the real test environment where the use cases are executed. This virtual environment must be surveyed and built very accurately in order to avoid deviations between the movement of the real VUT and its virtual copy-so-called Digital Twin-during testing. When the VUT starts to drive on the real proving ground, its digital twin also behaves in the same way, influencing the simulation and subsequently activating different disturbances (e.g., pedestrians, cyclists, and other vehicles). Based on the simulation, the control software triggers the real target objects that can influence the VUT on the real proving ground. The continuously changing localization data will be registered again in the simulation, hence closing the loop. This concept provides opportunities to realize a

complex, fully controlled environment for the testing of ADASs functions. It can operate on connected and automated vehicles on high-tech proving grounds. With some of its key elements, SciL was already partially realized and presented with self-developed software solutions and interfaces during a proof-of-concept demonstration [40]. Additionally, SciL provides possibilities for test scenario investigation, definition, and preparation before real-world tests are conducted. With the right tool, the predefined trajectories, speed profiles, and activation criteria can be converted to use real testing equipment (e.g., dummies, target carriers) [41].

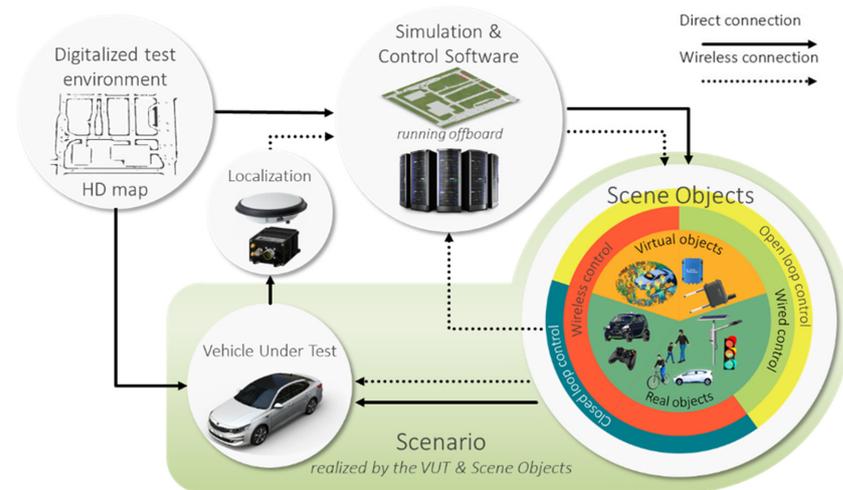


Figure 1. The architecture and operation principals of the SciL concept [38] (p. 35621).

The SciL concept can also be realized with widely used simulation tools like the previously mentioned software, providing a wider range of possible applications. In the following sections, a potential solution will be presented and tested using a co-simulation technology.

2.2. Applied Simulation Software and Their Application in the Simulation Framework

As it was introduced in the previous section, the key component of the SciL is the ‘Simulation and Control Software’. Based on previous research, we realized that none of the available simulation tools fulfill the needs of the SciL entirely. Therefore, we decided to select one of the available options and self-develop the missing functions. For this role, Simcenter Prescan© (version: 2019.1.0) was chosen. Simcenter Prescan© is a simulation software for automotive usage focusing primarily on testing and simulation of various sensors and other elements of the perception system. Consequently, the software contains accurate, idealized, and realistic sensor models. The software provides a user-friendly human–machine interface (HMI) for building different environments, road networks, and scenarios with various road users. In the simulation, all participants can be adjusted and handled on the same level, and dynamic calculations are available by every vehicle on demand. The software can be effectively used in ADASs and autonomous driving function-related simulations [42,43].

Simcenter Prescan© supports the open-source OpenDRIVE® format that can be considered as de facto standard in vehicle simulations. OpenDRIVE® was started to be developed in 2005 by VIREs Simulationstechnologie GmbH in cooperation with Daimler and currently maintained by ASAM (Association for Standardization of Automation and Measuring Systems) since 2020 and the latest available version was released in August 2021 [44,45].

Simcenter Prescan© is strongly based on MATLAB Simulink®. The simulation can run only in this software environment, and all of the changes applied in the HMI will modify the blocks and signals in the Simulink®. As an advantage of this connection, Prescan© can be extended with additional simulation models and interfaced simply through MATLAB with other software or hardware components.

As mentioned above, in the SciL concept, the VUT can be handled as a ‘black box’; therefore, the use of external sensors is recommended that function independently from the internal networks of the VUT (e.g., CAN, FlexRay). The acquired data should be sent to the ‘Simulation and Control Software’, ideally in real-time with low latency using wireless communication technologies. However, for the development process, we decided to use previously measured data to generate an offline database for developing the necessary connections and interface within the Simulink® environment of Prescan©. To ensure that the established connections would not need to be redefined during each test, an interface block was developed that can provide the data in the right format to Simcenter Prescan©. For further development, the establishment of an alternative data source was necessary in order to reduce the time and costs of proving ground testing. As an alternative source, IPG CarMaker® (version: 9.1.5) was utilized, which can be connected through the developed interface.

The software includes detailed vehicle models, intelligent driver models, and the opportunity to create detailed virtual environments with complex road networks flexibly. With various add-ons, the IPG CarMaker® is able to perform open- or closed-loop testing methods such as SiL, MiL, or even ViL configurations. Additionally, IPG CarMaker® can implement various sensor models to simulate the realistic perception systems of vehicles. The software can also handle the OpenDRIVE® format, which allows road networks to be imported from external sources. The software also allows multiple potential road users, which could be activated via events, although the simulation is always focusing on the appropriate dynamic calculation of the VUT [46,47].

The vehicle models are easily parameterized and very accurate, providing realistic data about the behavior of vehicles. Through the optional link to MATLAB Simulink®, these data can be used as real signals from the VUT. With the help of this connection, the IPG CarMaker® can provide the necessary inputs for the SciL architecture in the presented co-simulation method.

To ensure the right cooperation between the two software, the OpenDRIVE® virtual environment of the ZalaZONE Smart City was imported into the vehicle simulation tools. The model was generated using laser scanning, but manually built environments can also be used in both software [48].

To connect the two software the 2020a version of MATLAB Simulink® were used. The established connection between the two software can facilitate the development of SciL control algorithms. The activation commands and triggers can also be connected back to IPG CarMaker®, providing the closed-loop architecture. Based on this connection, the IPG CarMaker® can act as a real environment from the Prescan© point of view. Figure 2 shows the different roles of used software solutions used in the SciL operation principle.

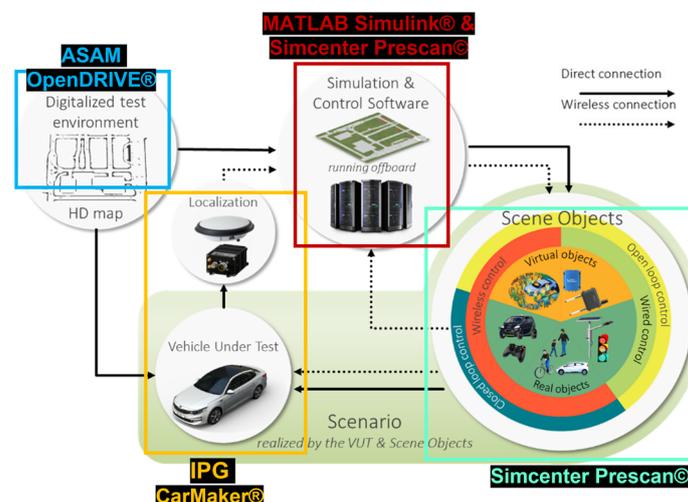


Figure 2. The role of the different simulation software solutions in the architecture of SciL.

2.3. Data Acquisition from the Real VUT

Acquiring the real VUT data, the development car (Figure 3) provided by TÜV Rheinland-KTI Ltd. (Budapest, Hungary) was equipped with the necessary sensors to be tested on the ZalaZONE proving ground. During measurements, data regarding vehicle movement were measured by independent sensors, ensuring the ‘black box’ concept; however, in some instances, the native CAN of the vehicle was also used for specific purposes.

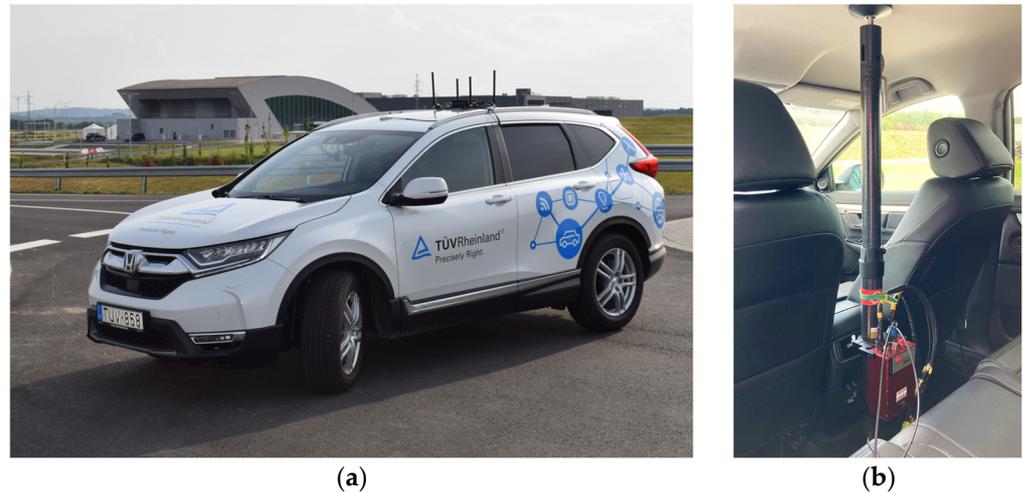


Figure 3. The used VUT and measurement system provided by TÜV Rheinland-KTI Ltd.: (a) Honda CR-V 2021 passenger car as the VUT; (b) OXTS RT3000 v3 INS unit as the main measurement system.

The most important data for the realization of the digital twin is the moving of the VUT along and around the three axes. During vehicle testing, usually six degrees of freedom are defined: (i) longitudinal, (ii) lateral, (iii) vertical along the axes, and (iv) pitch, (v) roll, and (vi) yaw around the axes (Figure 4). These values can be measured both in a local and a global coordinate frame. In most instances, the longitudinal and lateral values are measured as absolute positioning data; however, in the current investigation, the yaw can be replaced with the absolute heading of the vehicle and registered for further evaluation. In proving ground testing practice, these values are mainly measured with a high precision inertial measurement system (INS), which is a combination of a differential global navigation satellite system (DGNSS) and an inertial measurement unit (IMU). These devices also can accurately measure other relevant values, such as different accelerations and the velocity of the VUT. During the measurement, an OXTS RT3000 v3 INS unit was equipped into the VUT that was supported with NTRIP RTK differential correction using the local base station and the service of the ZalaZONE proving ground. The NTRIP correction can be provided using mobile network-based communication; subsequently, the interference with other proving ground users that are using regular 400–800 MHz frequency-based base stations can be avoided.

The testing equipment must be installed with high precision into the vehicle to ensure the 1 cm positioning accuracy, 0.05 km/h velocity, and 0.03° – 0.05° pitch, roll, and heading accuracy in the measurements. During installation, relative positions of the antennas and the main unit in relation to the wheels, steered and non-steered axles as well as to the ground, must be measured accurately [49].

The exact positioning data ensures that the digital twin can accurately follow the movement of the real VUT in the simulation. The appropriate movement of the vehicle bodywork is essential for the appropriate virtual sensor alignment, which, in this case, can provide more realistic simulated perception sensor data. However, the installed INS can provide measurements covering all six degrees of freedom; for this research, we only used lateral, longitudinal, heading, and velocity values. The OpenDRIVE[®] based virtual environment does not contain an elevation profile. Therefore, registration of the vertical

movement was not necessary. During measurement, only a local coordinate system was utilized in which the 2D movement with relative positions-in meter-could be registered. The software tools provided by OXTS were used to export data into a .csv file format, which was suitable for generating the necessary offline database for MATLAB Simulink[®]. Measurements were executed with a 100 Hz data sampling, significantly prolonging the duration of file export (30–60 s for each test run) and resulting in a huge amount of data. To reduce the timeframe of file export, a lower data sampling value was selected. Based on our experience, the optimal data sample value was found to be 25 Hz.

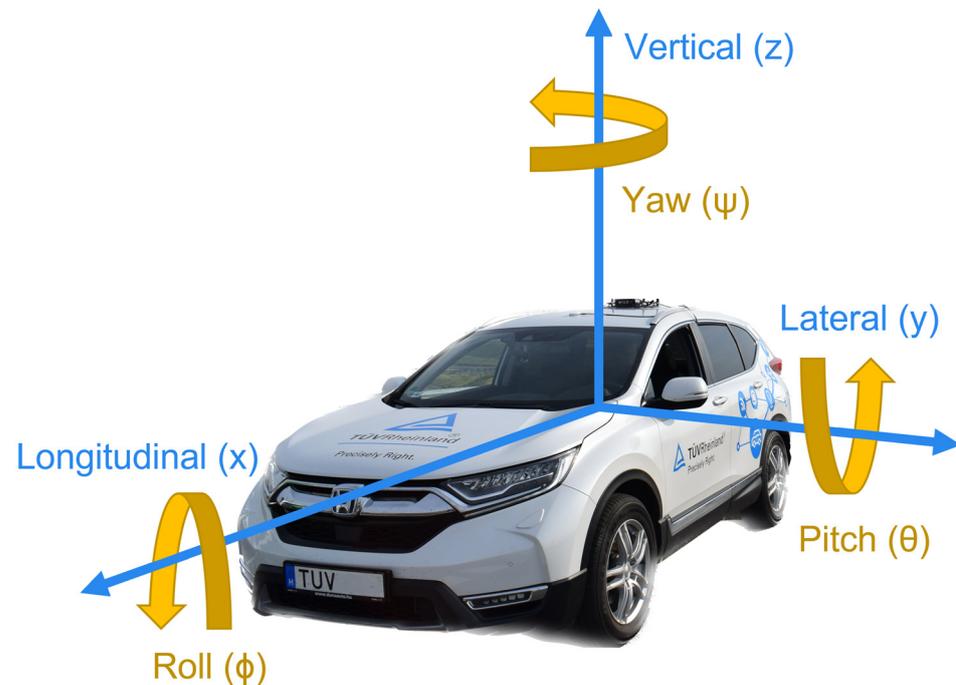


Figure 4. The six degrees of freedom in the vehicle testing.

Besides VUT movement, additional measurements can be performed to ensure proper visualization of the digital twin in the simulated environment. Measurements of the wheel radius are necessary to adequately animate wheel rotation animation. Also, the steering angle was measured for the accurate visualization of the steered wheel movement. The angle of the steered wheels was calculated from the steering wheel angle. Although it is possible to utilize an independent sensor, the internal CAN of the VUT provides sufficient information regarding these measures. In order to record data from the internal CAN, a new breakout had to be implemented, bypassing the built-in gateway in the OBD2 connector. An additional database file (.dbc) was necessary to be able to identify relevant CAN messages. The CAN was accessed with National Instruments (NIs) CAN equipment, and data were recorded using a self-developed software solution (in LabVIEW[®]). To identify the steering wheel/steered wheels angle ratio, a classical turntable equipped with a scale for wheel alignment was utilized. As a constant value, this was programmed into the simulation software during the parametrization of the vehicle.

For appropriate visualization, the status of the light and signaling devices installed on the VUT were also recorded. In this research, the most important light device was the brake lights. To check the status of the brake lights, a self-developed independent photo diode-based sensor was used. The sensor provided an analog voltage signal to an NI datalogger that was also recorded in LabVIEW[®].

Based on the above-mentioned real-world measurements, a database was created with the following values shown in Table 1:

Table 1. The measured values from the real VUT.

Measured Value	Dimension	Variable Name
Local longitudinal coordinate	m	PosX
Local lateral coordinate	m	PosY
Velocity	m/s	Velocity
Heading	°	Heading
Steering wheel angle	°	Steering_Angle
Status of the brake light	N/A	Brake_Light

2.4. Preparing Simcenter Prescan© for the Handling of External Data

Firstly, using the GUI of Prescan©, the test environment, at least one vehicle (e.g., the VUT), and one disturbance (e.g., pedestrian) were selected. Connections and activation criteria between these two participants were established in a later stage. The developed links could be copied and added to new participants later in Simulink®. In later stages, it is possible to modify connection links once the control algorithm is developed. The procedure of this will be introduced in Section 3.1. Prescan© offers the option to choose between different vehicle models; however, due to the lack of an exact model, one of the base models in the software was parametrized according to real vehicle parameters. After generating the scenario for the first time, the Simulink® building blocks were also generated and became available. Generated links can be modified utilizing Simulink® tools. Initially, all the different participants had predefined trajectories and speed profiles that were based on the initial setup in the GUI and followed during the simulation. Figure 5 shows the structure of the Simulink® model of the digital twin following modifications described below.

The first step was to override the original speed profiles and trajectories with our input data gathered from an independent source. For that, the original trajectory (or 'Path') signal was disconnected and terminated. By the division of the 'Path' signal, 15 sub-signals could be visualized, including: (i) positions, (ii) velocities, (iii) acceleration, (iv) rotations, and (v) angular velocities along and around all three axes covering the six degrees of motion. To override the original signal, a new signal had to be created with the same sub-signal structure. For the proper trajectory following the capability of the digital twin, three signals from the real car (longitudinal position, lateral position, and heading) and a constant altitude value were sufficient. Therefore, the other 11 signals were grounded. Instead of the original signal, the new trajectory signal was connected to the right input of the vehicle model. Subsequently, the digital twin was able to move on the same trajectory with the same heading as the real vehicle.

For better visualization, the wheel movements of the digital twin were aligned with the real car as well. This requires the acquisition of accurate velocity and steering angle values. Firstly, the velocity was converted into wheel angle, which provides the proper rotation measure of the wheels. For that, a 'Velocity to Wheel Angle' block was used, which needs the measured wheel radius as an input. For the appropriate steering angle, the degree was converted into radian. These two modified signals were used as inputs of the original 'WheelDisplacement' block, which provided the correct wheel movements calculated with the new signals. The brake light status signal can be easily connected to the original brake light blocks (left and right lamps separately), overriding the original set of 0 or 1 values.

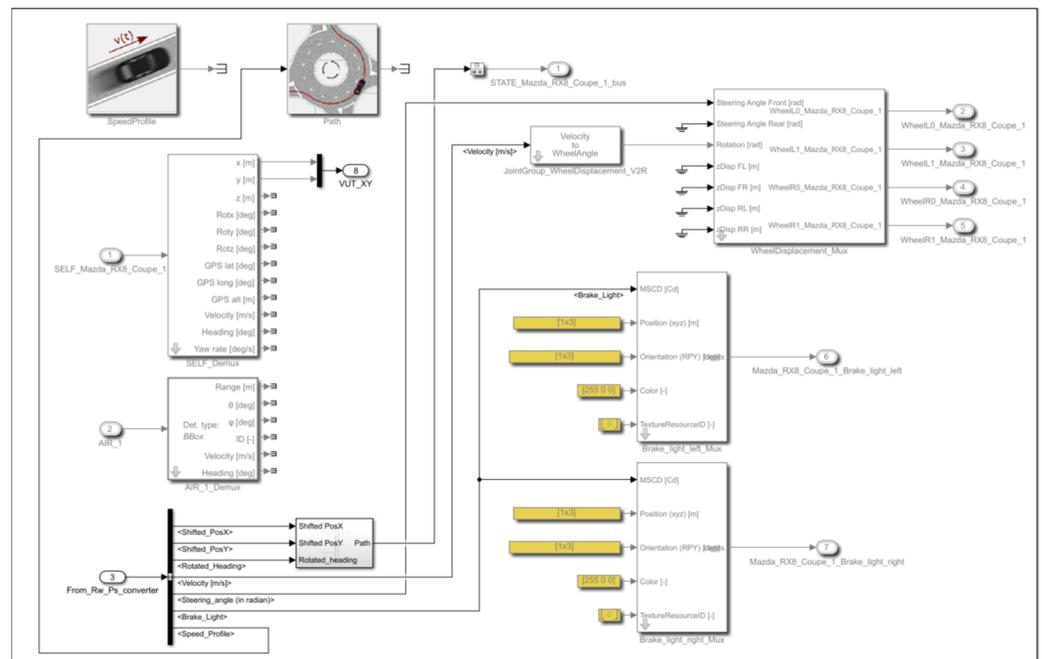


Figure 5. The modified Simulink® model of VUT block in Simcenter Prescan©.

The original ‘SpeedProfile’ signal, initially directly connected to the ‘Path’ block, was also replaced with modified signals, as it was necessary as an initial condition for starting the simulation. Although the ‘Path’ signal was already terminated, it did not influence the simulation. The ‘SpeedProfile’ contains the distance, velocity, and acceleration values, which can be calculated from the vehicle speed using the integrator and derivative blocks.

Following modifications, the Prescan© simulation was run based on external data; However, it was necessary to connect the external data through the interface block, described in the next section.

2.5. Developing the Necessary Interface to Connect Prescan© with External Data

The initial version of the ‘Simulation and Control Software’ required various data conversions to be able to run testing scenarios properly. To facilitate this process, an interface block was developed that converts input data into the appropriate structure for the simulation (Figure 6).

This block can read the data provided by the vehicle, including lateral position, longitudinal position, heading, velocity, steering angle, and status of the brake lights. The blocks contain the necessary conversation to provide the data in the right structure to the VUT block of Prescan©. The interface block shifts positions first if the virtual- and the real test environments are defined with different coordinate system origins. The heading value must also be modified. This required a mirroring, a rotation with 90 degrees, and a degree-to-radian conversion. The velocity was acquired in km/h from the INS unit. Hence, this had to be converted into m/s. Based on this value, the previously mentioned ‘SpeedProfile’ replacement with derivative and integrator blocks was also created. Last but not least, the steering angle was converted from degree to radian, and the ratio may also be included, which was identified between the steering wheel angle and the angle of the steered wheels.

With the help of this interface block, the links and calculations in the remaining part of the simulation can be independent of the input data that were acquired from the real VUT. These data were imported into MATLAB Workplace, and with the help of a Real-time ‘Signal From Workspace’ built-in Simulink® block, this could be read with the same sampling that is defined for the simulation.

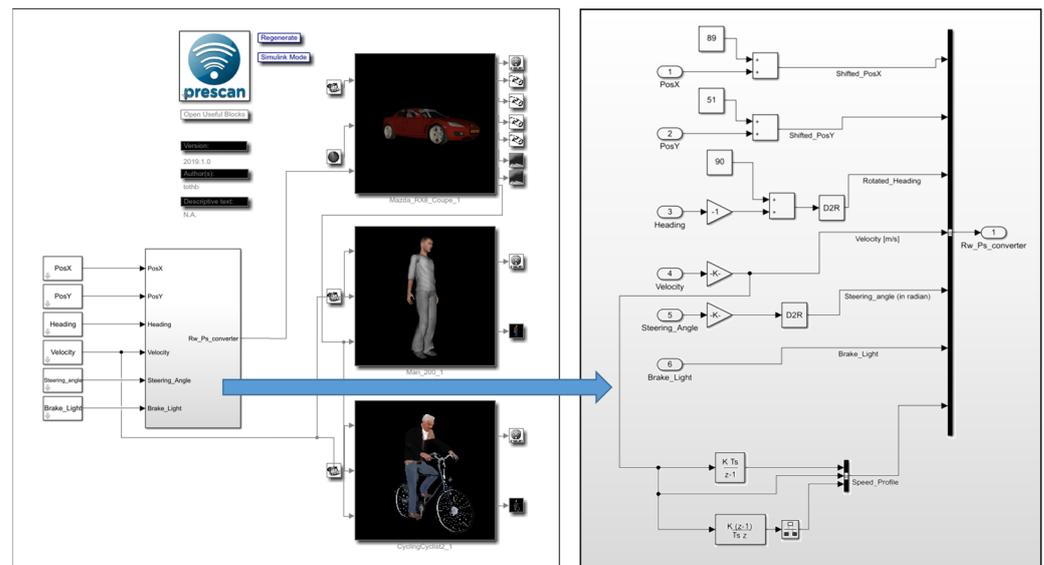


Figure 6. The developed interface block and its location into the entire Simulink® model.

3. Results

The developed simulation framework can be used for developing our required control algorithms for the different disturbances and targets introduced by the SciL concept. In this chapter, we present how a EuroNCAP-based control algorithm can be developed and validated with real VUT data and with simulated input data. For the latter, the connection of the IPG CarMaker® as an alternative source is essential, which is also briefly described in the following.

3.1. Developing an Example Control Algorithm for Test Target Activation

After the real VUT data were successfully imported into the simulation, we had to create the activation criteria for the disturbances based on the movement of the digital twin. In this paper, the EuroNCAP Car-to-Pedestrian Nearside Adult (CPNA) scenario was chosen as an example. This requires a pedestrian target, although we can also implement more targets into the same simulation model. The schematic view of the previously mentioned scenario can be seen in Figure 7. The necessary parameters and equations can be defined based on the related EuroNCAP protocol [50].

According to the protocol, this test scenario is applicable for AEB testing within the speed range of 10–60 km/h with 5 km/h incremental steps. The scenario needs to be tested with 25% and 75% overlap. For this specific example, the 75% overlap was chosen. Based on this parameter, the activation criteria can be defined. In the scenario, the dummy needs to start from a 4 m distance measured from the centerline of the VUT. Within the first 1 m (s_{acc}), the dummy needs to reach the 5 km/h target speed (v_{target}) from a standstill. The duration of the acceleration (t_{acc}) can be calculated from these values with the following Equation (1):

$$t_{acc} = \frac{2 \times s_{acc}}{v_{target}} \quad (1)$$

Using the Formula (1), the result is 1.44 s. In the case of 75% overlap, the dummy needs to travel 3 m plus 25% of the vehicle width to reach the ‘K’ impact point. The width of the VUT is 1.855 m. Hence, the necessary travel distance while the dummy is traveling with a constant 5 km/h velocity is 3.46 m (total travel distance of the dummy until the collision point is 4.46 m). Based on this value, the ‘Time to Collision’ (TTC) can be calculated

and used as a trigger criterion for the dummy activation. The TTC value for the dummy (TTC_{dummy}) can be calculated with the following Equation (2):

$$TTC_{dummy} = t_{acc} + \frac{s_{const} + 0.25 w_{vehicle}}{v_{target}} \quad (2)$$

The t_{acc} is the time component of the acceleration, $w_{vehicle}$ is the width of the vehicle, s_{const} is the remaining distance from 4 m that the dummy needs to travel with constant speed after the acceleration, and v_{target} is the speed of the dummy, which is the constant in the denominator. Using Equation (2), the calculated value is 3.9312 s. The trigger needs to be sent when the calculated TTC for the VUT reaches this value. This TTC value can be calculated based on the longitudinal distance of the VUT from the meeting point measured on the centerline of the VUT. In the local coordinate system of the simulation, the 'x' and 'y' coordinates of the VUT and the VRU target can be acquired. Hence, the absolute distance of the two objects can be calculated. Figure 8 demonstrates how the longitudinal distance can be calculated. The absolute distance can be calculated with the Equation of the distance between two points (3).

$$d = \sqrt{(DummyX - PosX)^2 + (DummyY - PosY)^2} \quad (3)$$

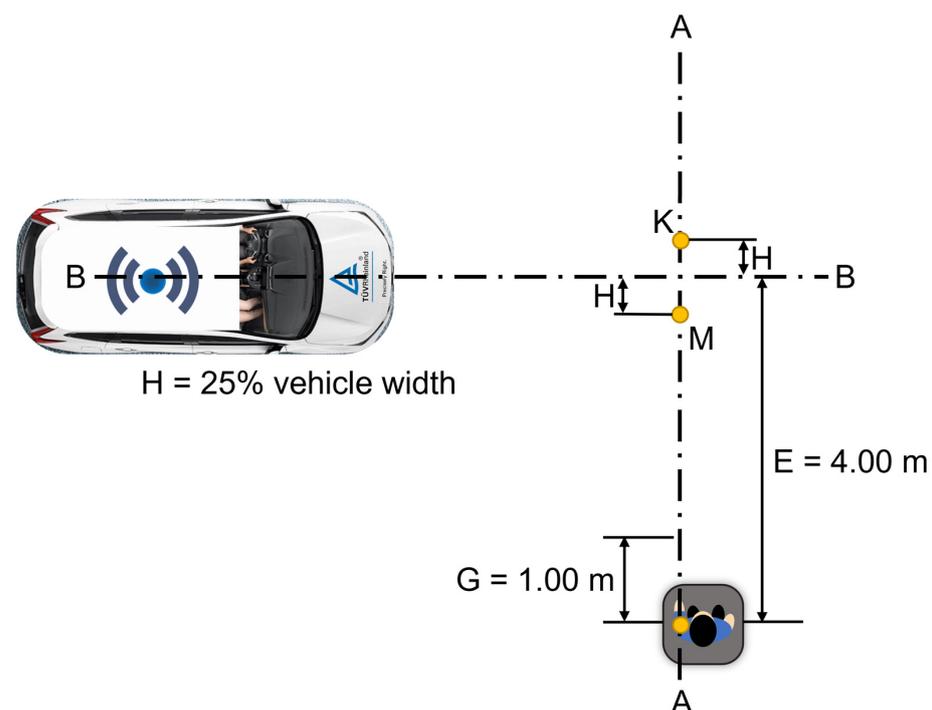


Figure 7. EuroNCAP Car-to-Pedestrian Nearside Adult (CPNA) scenario layout. Axes: AA—Trajectory of pedestrian dummy H-point, BB—Axis of centerline of VUT; Distances: E—Dummy H-point, start to 50%-impact (near side), G—Dummy acceleration distance (walking), H—Impact point offset for 25% or 75%; Points: K—Impact position for 75% near-side scenario, M—Impact position for 25% near-side scenario [50].

DummyX and DummyY are the coordinates of the VRU target, and PosX and PosY are the coordinates of the VUT in the local coordinate system. The same naming convention is used also in the simulation model shown in Figure 9. The start distance of the dummy is well-defined in the case of CPNA 75 testing. Thus, the longitudinal distance can be calculated using the Pythagorean theorem. However, the actual value needs to be counted from the closest point of the front bumper. Because the coordinates are provided from the center point of the vehicle geometry, we need to exclude half of the vehicle length from the

values using the Pythagorean theorem. Based on previously used formula, the necessary longitudinal distance can be calculated using the following Equation (4):

$$d_{\text{long}} = \sqrt{d^2 - \text{Dummy}_{\text{lat}}^2} - d_{\text{COG}} \quad (4)$$

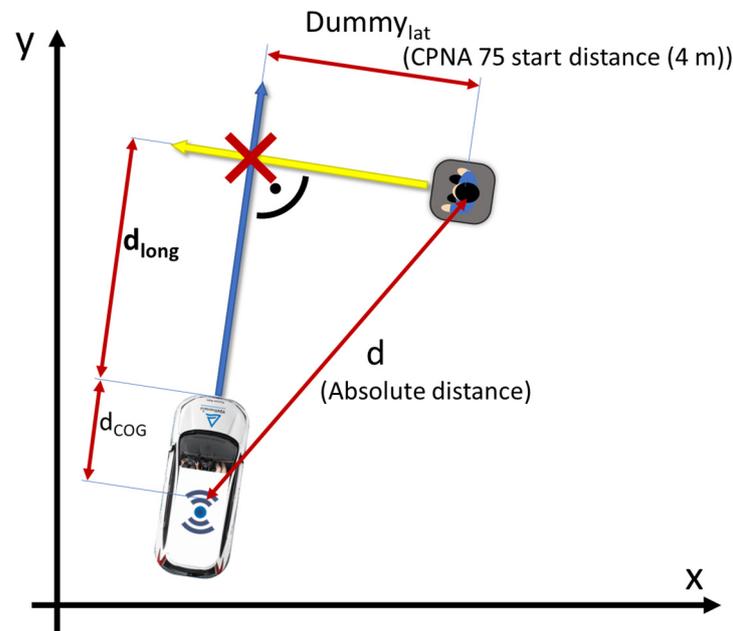


Figure 8. Calculation method for the longitudinal distance in the simulation coordinate system.

The d_{long} is the longitudinal distance, d the absolute distance, $\text{Dummy}_{\text{lat}}$ is the lateral distance of the dummy's start position measured from the vehicle center line, and d_{COG} is the distance of the distance of the vehicle's Center of Gravity (COG) point and its front bumper. The aimed TTC value can be calculated by dividing the d_{long} by the velocity of the vehicle. If the TTC is equal to or less than the $\text{TTC}_{\text{dummy}}$, the target can be activated. Based on the mentioned procedure, the Simulink® control algorithm model shown in Figure 9 can be developed.

The original inputs of the model are the 'Velocity' and the local coordinates of the VUT ('PosX', 'PosY') and the local coordinates of the VRU dummy ('DummyX', 'DummyY'). Using these values, the first 'User Defined Function' block in the model calculates the Absolute distance of the two objects, the second calculates the longitudinal distance, and the third one provides the TTC value. This TTC value is compared with the previously calculated $\text{TTC}_{\text{dummy}}$ value calculated using Equation (2) that belongs to the CPNA 75 scenario (3.9312 s). Once the TTC value is lower than the $\text{TTC}_{\text{dummy}}$ value, the trigger can be activated.

In the target's case, we can replace the original 'SpeedProfile' in the same way as it was described in Section 2.4. When the trigger is activated, we can send a user-defined signal into the 'SpeedProfile' output. The signal contains the desired speed profile of the dummy, which is created based on the EuroNCAP protocol. In this case, the 'Path' can be defined in the Prescan© GUI. Hence, we do not need to modify it. With this control algorithm, the dummy can be activated in line with the EuroNCAP protocol.

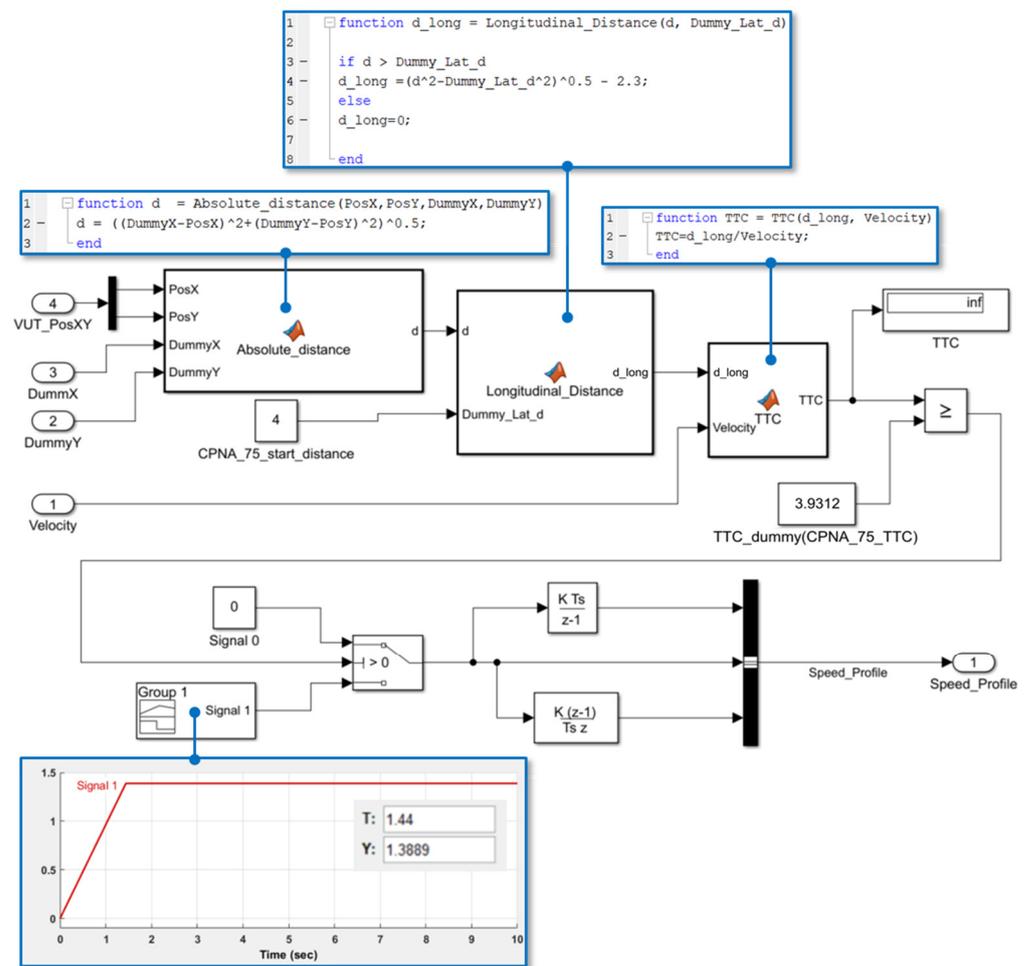


Figure 9. The control algorithm used for activating the dummy in the CPNA 75 scenario.

3.2. Connecting IPG CarMaker® to Provide External Data through the Developed Interface

Although the Prescan© based ‘Simulation and Control Software’ was developed and tested using measured real data to validate it in several scenarios, we can also create input data using different sources. In our case, IPG CarMaker® was chosen as a potential external source. For the appropriate implementation of this simulation tool, we need to identify, find, and convert the data which are provided by the IPG CarMaker® into a structure that can be used as input for the interface block. Similar to Prescan©, the scenarios also have to be defined in the GUI of the IPG CarMaker®. We must also use the same OpenDRIVE® road network that we are using in the Prescan©.

The IPG CarMaker® also has a Simulink® add-on, in which we can find the building blocks and signals of the running simulations. The IPG CarMaker® contains more sub-blocks that are responsible, e.g., for the calculation of the driver behavior (‘DrivMan’), vehicle control (‘VehicleControl’), and the dynamics of the tested vehicle (‘IPG Vehicle’). Most of the required data for feeding the interface can be found in ‘IPG Vehicle’; only the Brake Light Status is provided by the ‘DrivMan’ (Figure 10). These signals can be connected to the interface block, where only the correct positioning of the environment has to be adjusted in order to make the two coordinate systems match each other.

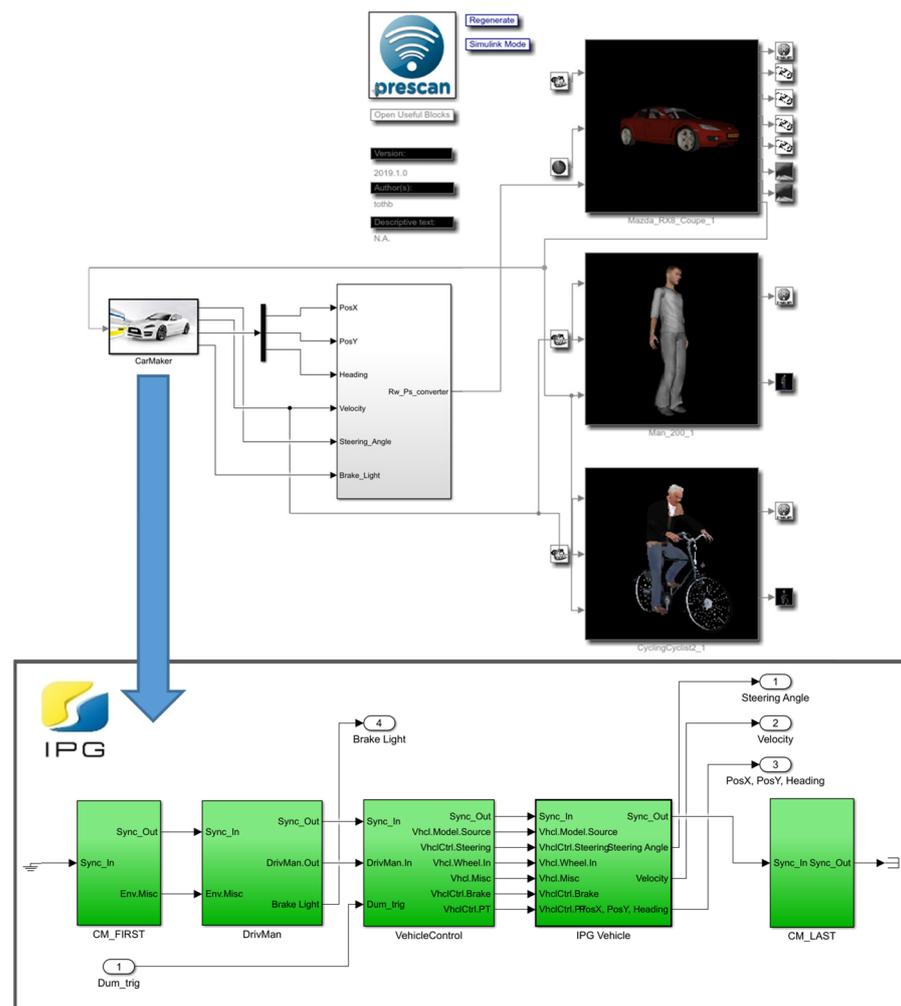


Figure 10. The structure of the IPG CarMaker® Simulink® model and its connection to Prescan®.

The IPG CarMaker® also provides the opportunity to implement VRU dummies into the simulation; however, the path and the speed profile of the dummy can be defined in the GUI. As a start condition, a so-called ‘VC.UserSignal’ can be defined, in which we can set a ‘0’ or ‘1’ value to activate the VRU target. With this method, we can connect the trigger value from the control algorithm back to the source IPG model, and we can influence the behavior of the source VUT in the IPG, which also affects the behavior of the digital twin.

3.3. Comparing the Result of the Simulation with the Real World Testing Data

In order to check the correct operation of the model, we compared the results of the real-world testing data, the Simcenter Prescan® simulation-based trigger data, and the IPG CarMaker® co-simulation-based data to inspect whether all the necessary testing parameters can be reproduced by using the developed co-simulation environment. During the real data acquisition, the measurement data from the real dummy target system was also provided, but this was not imported into the simulation environment, although these were also visualized on the figures that presented the results. Based on the EuroNCAP protocol, three different VUT testing speeds (10, 30, 50 km/h) were selected within the defined 10–60 km/h speed range, considering a step-by-step safe approach.

The results are presented in Table 2 and in the following figures (Figures 11–13), which show the results of the real-world measurement, the simulated trigger based on real VUT data, and the simulated data provided by the co-simulation. The results are organized by VUT testing speeds. In the real measurement data, a higher fluctuation of the VUT speed can be observed, especially in the case of 10 and 30 km/h testing speeds.

This phenomenon is caused by the control strategy of the driverless test system (DTS). Compared to this, the simulated data can provide a more even signal, which decreases the number of potential invalid test runs. In the real measurements, the speed profile of the dummy can be observed, which is a predefined value, hence independent from the VUT speed. With the simulated trigger data, the dummy provides a similar nature, but the signal is smoother compared to the real data. This could also be observed in the case of the dummy’s behavior in the co-simulation. The TTC values are really close to the calculated theoretical value (3.9312 s in Section 3.1); the slight difference is mainly caused by the sampling value of the simulation. In the case of the simulated data, a built-in dummy control was chosen in IPG CarMaker®. Hence, the dummy’s speed profile cannot be defined and controlled directly from the developed control algorithm. However, the TTC values can also be identified, and it can be seen that the dummy is triggered with proper timing. This also highlights the difference between different simulation approaches. In the case of the simulated trigger in the Prescan®, the whole dummy speed profile and control model can be sent in the trigger signal. Contrary to this, the IPG CarMaker® only needs a 0/1 start trigger, and then the dummy can be operated on built-in motion models. Choosing and parametrizing the appropriate control model in the IPG CarMaker® GUI, a similar dummy behavior can also be realized in the co-simulation. This effect can also be realized in later stages when the activation strategy for the real targets needs to be defined from the SciL control algorithm.

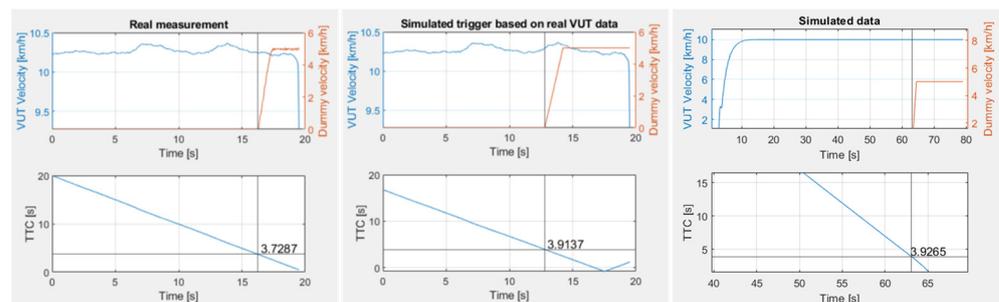


Figure 11. Results from the real measurements and simulated testing with 10 km/h VUT speed.

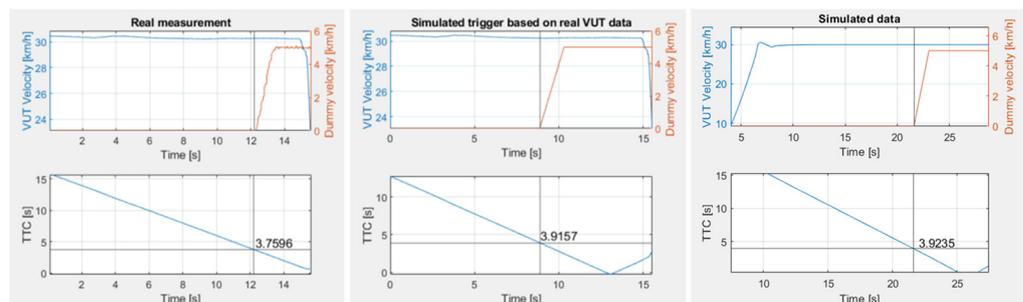


Figure 12. Results from the real measurements and simulated testing with 30 km/h VUT speed.

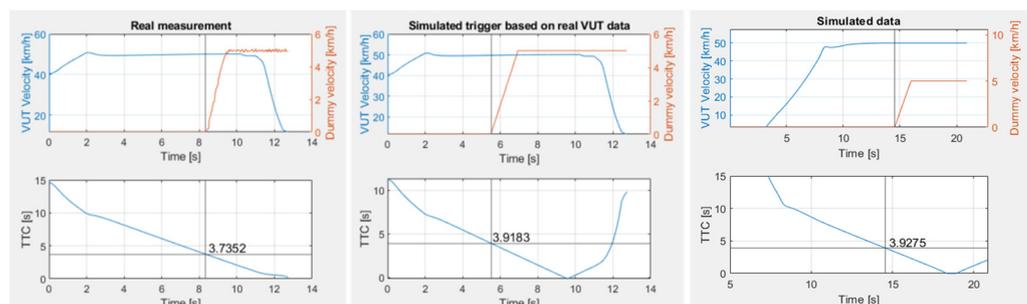


Figure 13. Results from the real measurements and simulated testing with 50 km/h VUT speed.

Table 2. Comparison of the key parameters measured in the case of different VUT speeds.

VUT Testing Speed		10 km/h	30 km/h	50 km/h
TTC when the dummy is activated [s]	Real-world measurement	3.7287	3.7596	3.7352
	VUT data-based simulated trigger measurement	3.9137	3.9157	3.9183
	Simulated measurement	3.9265	3.9235	3.9275
d_{long} when the dummy is activated [m]	Real-world measurement	10.61	31.62	51.94
	VUT data-based simulated trigger measurement	11.19	32.89	54.08
	Simulated measurement	10.91	32.69	54.55
Traveled distance by the dummy when $TTC \approx 0$ [m]	Real-world measurement	4.48	4.45	4.44
	VUT data-based simulated trigger measurement	4.43	4.44	4.42
	Simulated measurement	4.51	4.51	4.52

Additionally, for the TTC values when the dummy is activated, Table 2 also contains information about the actual longitudinal distance of the VUT at the same moment. It can be observed that with higher speeds, the trigger is sent from a higher distance. The values in the case of the simulated data are closer, similar to the TTC values. The slight difference in the case of TTC and activation distance is caused by the adaptive control strategy of the real dummy. In the case of the simulated trigger and co-simulation data, the acceleration section is always close to 1.44 s, but the real target can differ from this value. Based on the data from real-world measurements, this acceleration time is usually less than 1.44 s (mainly between 1.21 and 1.26 s). Hence, the trigger can be sent later, which means lower TTC and VUT longitudinal distance values.

Table 2 also contains the dummy's traveled distance until TTC reaches zero, i.e., until the collision. In this case, it can be observed how the values are differed from the theoretical 4.46 m. The real-world testing data are close to this value, caused by the controller's adaptive strategy. In the case of the simulated trigger, these values are also within 4 cm, but usually, the dummy arrives a bit later, which can be caused by the fluctuation of the VUT's speed. In the case of the co-simulation, these values are closer to each other, caused by the more stable VUT speed. However, the dummy is usually arriving earlier. That can be adjusted with the further parametrization of the simulated dummy.

4. Conclusions

The developed solution can help to prepare for real-time testing of the SciL concept using an industrial vehicle simulation software, Simcenter Prescan©. The developed and presented simulation environment shows an example of the potential usages. However, there are various possibilities for further developing the concept, and using co-simulation will always facilitate the testing and validating process of the new functions. Using some modification and the developed interface block, IPG CarMaker® can provide realistic input data for the 'Simulation and Control Software' in real-time, replacing the necessity of real-world measurements in the early stages of the development. In the common practice, simulation technologies are used prior to the real-world measurements. In the presented unique approach, real-world testing data were acquired first and were utilized to create a simulation control model to ensure the capability for handling real-world testing data even in real-time conditions in the later development stages. The built development environment provides the possibility to be fed with simulated data in the same structure, facilitating the development of the SciL control software.

In the future, we would like to finalize this core component of the SciL concept and then extend the system with real-time measured real-car data and target objects using low latency radiocommunication. Connecting different simulation tools is always challenging; hence,

we hope that the presented method to connect IPG CarMaker[®] with Simcenter Prescan[®] can also be useful for researchers who decide to use these tools for similar approaches. With the continuous development of the system, we believe Simcenter Prescan[®] can become the main control component of the SciL concept, accelerating the testing of future driving functions and automated vehicles.

Author Contributions: Conceptualization, B.T. and Z.S.; methodology, B.T.; software, B.T.; validation, B.T.; formal analysis, B.T. and Z.S.; investigation B.T. and Z.S.; resources, B.T.; data curation, B.T. and Z.S.; writing—original draft preparation, B.T.; writing—review and editing, B.T. and Z.S.; visualization, B.T.; supervision, Z.S.; project administration, B.T.; funding acquisition, B.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development, and Innovation Fund, financed under the KDP-2021 funding scheme.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Relevant data for the result are contained within the article.

Acknowledgments: The authors are grateful to TÜV Rheinland-KTI Ltd. for providing resources, including personnel, vehicle and measurement equipment, and organizing the necessary proving ground usage.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ADASs	Advanced Driver Assistance Systems
AEB	Automated/Autonomous Emergency Braking
AEB	Automatic Emergency Braking
CPNA	Car-to-Pedestrian Nearside Adult
COG	Center of Gravity
DGNSS	Differential Global Navigation Satellite System (DGNSS)
DTS	Driverless Test System
EuroNCAP	European New Car Assessment Programme
GUI	Graphical User Interface
HMI	Human–Machine Interface
INS	Inertial Measurement Unit
MiL	Model-in-the-Loop
NIs	National Instruments
SciL	Scenario-in-the-Loop
SiL	Software-in-the-Loop
TTC	Time-to-Collision
ViL	Vehicle-in-the-Loop
VRU	Vulnerable Road User
VUT	Vehicle Under Test

References

1. Nowakowski, C.; Shladover, S.E.; Chan, C.-Y. Determining the Readiness of Automated Driving Systems for Public Operation: Development of Behavioral Competency Requirements. *Transp. Res. Rec.* **2016**, *2559*, 65–72. [[CrossRef](#)]
2. Wang, W.; Wu, L.; Li, X.; Qu, F.; Li, W.; Ma, Y.; Ma, D. An Evaluation Method for Automated Vehicles Combining Subjective and Objective Factors. *Machines* **2023**, *11*, 597. [[CrossRef](#)]
3. Chen, L.; Li, Y.; Huang, C.; Li, B.; Xing, Y.; Tian, D.; Li, L.; Hu, Z.; Na, X.; Li, Z.; et al. Milestones in Autonomous Driving and Intelligent Vehicles: Survey of Surveys. *IEEE Trans. Intell. Veh.* **2023**, *8*, 1046–1056. [[CrossRef](#)]

4. Zhao, D.; Lam, H.; Peng, H.; Bao, S.; LeBlanc, D.J.; Nobukawa, K.; Pan, C.S. Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 595–607. [[CrossRef](#)] [[PubMed](#)]
5. Zhao, D.; Huang, X.; Peng, H.; Lam, H.; LeBlanc, D.J. Accelerated Evaluation of Automated Vehicles in Car-Following Maneuvers. *IEEE Trans. Intell. Transport. Syst.* **2018**, *19*, 733–744. [[CrossRef](#)]
6. Vaio, M.D.; Falcone, P.; Hult, R.; Petrillo, A.; Salvi, A.; Santini, S. Design and Experimental Validation of a Distributed Interaction Protocol for Connected Autonomous Vehicles at a Road Intersection. *IEEE Trans. Veh. Technol.* **2019**, *68*, 9451–9465. [[CrossRef](#)]
7. Koopman, P.; Wagner, M. Challenges in Autonomous Vehicle Testing and Validation. *SAE Int. J. Trans. Saf.* **2016**, *4*, 15–24. [[CrossRef](#)]
8. Altamimi, H.; Varga, I.; Tettamanti, T. Urban Platooning Combined with Dynamic Traffic Lights. *Machines* **2023**, *11*, 920. [[CrossRef](#)]
9. Weissensteiner, P.; Stettinger, G.; Rumetshofer, J.; Watzenig, D. Virtual Validation of an Automated Lane-Keeping System with an Extended Operational Design Domain. *Electronics* **2022**, *11*, 72. [[CrossRef](#)]
10. Coppola, A.; Costanzo, L.D.; Pariota, L.; Santini, S.; Bifulco, G.N. An Integrated Simulation Environment to Test the Effectiveness of GLOSA Services under Different Working Conditions. *Transp. Res. Part C Emerg. Technol.* **2022**, *134*, 103455. [[CrossRef](#)]
11. Feng, S.; Feng, Y.; Sun, H.; Zhang, Y.; Liu, H.X. Testing Scenario Library Generation for Connected and Automated Vehicles: An Adaptive Framework. *IEEE Trans. Intell. Transport. Syst.* **2022**, *23*, 1213–1222. [[CrossRef](#)]
12. Fayazi, S.A.; Vahidi, A.; Luckow, A. A Vehicle-in-the-Loop (VIL) Verification of an All-Autonomous Intersection Control Scheme. *Transp. Res. Part C Emerg. Technol.* **2019**, *107*, 193–210. [[CrossRef](#)]
13. Buzdugan, I.-D.; Butnariu, S.; Roşu, I.-A.; Pridie, A.-C.; Antonya, C. Personalized Driving Styles in Safety-Critical Scenarios for Autonomous Vehicles: An Approach Using Driver-in-the-Loop Simulations. *Vehicles* **2023**, *5*, 1149–1166. [[CrossRef](#)]
14. Feng, Y.; Yu, C.; Xu, S.; Liu, H.X.; Peng, H. An Augmented Reality Environment for Connected and Automated Vehicle Testing and Evaluation. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018. [[CrossRef](#)]
15. Kallweit, R.; Prescher, P.; Butenuth, M. Vehicle-in-the-loop: Augmenting real-world driving tests with virtual scenarios in order to enhance validation of active safety systems. In Proceedings of the 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV), Detroit, MI, USA, 5–8 June 2017.
16. Kang, Y.; Yin, H.; Berger, C. Test Your Self-Driving Algorithm: An Overview of Publicly Available Driving Datasets and Virtual Testing Environments. *IEEE Trans. Intell. Veh.* **2019**, *4*, 171–185. [[CrossRef](#)]
17. Shi, Y.; Liu, Z.; Wang, Z.; Ye, J.; Tong, W.; Liu, Z. An Integrated Traffic and Vehicle Co-Simulation Testing Framework for Connected and Autonomous Vehicles. *IEEE Intell. Transp. Syst. Mag.* **2022**, *14*, 26–40. [[CrossRef](#)]
18. Varga, B.; Ormándi, T.; Tettamanti, T. EGO-Centric, Multi-Scale Co-Simulation to Tackle Large Urban Traffic Scenario. *IEEE Access* **2023**, *11*, 57437–57447. [[CrossRef](#)]
19. Panossian, N.V.; Laarabi, H.; Moffat, K.; Chang, H.; Palmintier, B.; Meintz, A.; Lipman, T.E.; Waraich, R.A. Architecture for Co-Simulation of Transportation and Distribution Systems with Electric Vehicle Charging at Scale in the San Francisco Bay Area. *Energies* **2023**, *16*, 2189. [[CrossRef](#)]
20. Wang, Z.; Zheng, O.; Li, L.; Abdel-Aty, M.; Cruz-Neira, C.; Islam, Z. Towards Next Generation of Pedestrian and Connected Vehicle In-the-Loop Research: A Digital Twin Co-Simulation Framework. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2674–2683. [[CrossRef](#)]
21. Palmieri, M.; Quadri, C.; Fagiolini, A.; Bernardeschi, C. Co-Simulated Digital Twin on the Network Edge: A Vehicle Platoon. *Comput. Commun.* **2023**, *212*, 35–47. [[CrossRef](#)]
22. UNECE 1958 Agreement: Addendum 139–Regulation No. 140: Uniform Provisions Concerning the Approval of Passenger Cars with Regard to Electronic Stability Control (ESC) Systems. 2017. Available online: <https://unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/2017/R140e.pdf> (accessed on 3 April 2023).
23. Commission Implementing Regulation (EU) 2022/1426 of 5 August 2022 Laying down Rules for the Application of Regulation (EU) 2019/2144 of the European Parliament and of the Council as Regards Uniform Procedures and Technical Specifications for the Type-Approval of the Automated Driving System (ADS) of Fully Automated Vehicles (Text with EEA Relevance). Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32022R1426> (accessed on 4 April 2023).
24. UNECE: New Assessment/Test Method for Automated Driving (NATM) Guidelines for Validating Automated Driving System (ADS). 2022. Available online: <https://unece.org/sites/default/files/2022-04/ECE-TRANS-WP.29-2022-58.pdf> (accessed on 4 April 2023).
25. Sun, X.; Yu, F.R.; Zhang, P. A Survey on Cyber-Security of Connected and Autonomous Vehicles (CAVs). *IEEE Trans. Intell. Transport. Syst.* **2022**, *23*, 6240–6259. [[CrossRef](#)]
26. UNECE 1958 Agreement: Addendum 154–Regulation No. 155: Uniform Provisions Concerning the Approval of Vehicles with Regards to Cyber Security and Cyber Security Management System. Available online: <https://unece.org/sites/default/files/2023-02/R155e%20%282%29.pdf> (accessed on 20 September 2023).
27. Donà, R.; Ciuffo, B. Virtual Testing of Automated Driving Systems. A Survey on Validation Methods. *IEEE Access* **2022**, *10*, 24349–24367. [[CrossRef](#)]
28. Razdan, R.; Akbaş, M.İ.; Sell, R.; Bellone, M.; Menase, M.; Malayjerdi, M. PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration. *IEEE Access* **2023**, *11*, 28343–28354. [[CrossRef](#)]
29. Donà, R.; Vass, S.; Mattas, K.; Galassi, M.C.; Ciuffo, B. Virtual Testing in Automated Driving Systems Certification. A Longitudinal Dynamics Validation Example. *IEEE Access* **2022**, *10*, 47661–47672. [[CrossRef](#)]

30. Siegl, S.; Ratz, S.; Düser, T.; Hettel, R. Vehicle-in-the-Loop at Testbeds for ADAS/AD Validation. *ATZ Electron. Worldw.* **2021**, *16*, 62–67. [[CrossRef](#)]
31. Solmaz, S.; Rudigier, M.; Mischinger, M. A Vehicle-in-the-Loop Methodology for Evaluating Automated Driving Functions in Virtual Traffic. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1465–1471. [[CrossRef](#)]
32. Drechsler, M.F.; Peintner, J.; Reway, F.; Seifert, G.; Riener, A.; Huber, W. MiRE, A Mixed Reality Environment for Testing of Automated Driving Functions. *IEEE Trans. Veh. Technol.* **2022**, *71*, 3443–3456. [[CrossRef](#)]
33. Drechsler, M.F.; Seifert, G.; Peintner, J.; Reway, F.; Riener, A.; Huber, W. How Simulation based Test Methods will substitute the Proving Ground Testing? In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022; pp. 903–908. [[CrossRef](#)]
34. Katzorke, N.; Vinçon, C.; Kolar, P.; Lasi, H. Fields of Interest and Demands for a Digital Proving Ground Twin. *Transp. Res. Interdiscip. Perspect.* **2023**, *18*, 100782. [[CrossRef](#)]
35. Katzorke, N.; Moosmann, M.; Imdahl, R.; Lasi, H. A Method to Assess and Compare Proving Grounds in the Context of Automated Driving Systems. In Proceedings of the IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6. [[CrossRef](#)]
36. Németh, H.; Hány, A.; Szalay, Z.; Tihanyi, V.; Tóth, B. Proving Ground Test Scenarios in Mixed Virtual and Real Environment for Highly Automated Driving. In *Mobilität Zeiten Der Veränderung*; Springer: Wiesbaden, Germany, 2019; pp. 199–212. [[CrossRef](#)]
37. Bock, T. Vehicle in the Loop–Test und Simulationsumgebung für Fahrerassistenzsysteme. In *Audi Dissertationsreihe*; Vieweg: Göttingen, Germany, 2008; Volume 10.
38. Szalay, Z. Next Generation X-in-the-Loop Validation Methodology for Automated Vehicle Systems. *IEEE Access* **2021**, *9*, 35616–35632. [[CrossRef](#)]
39. Tihanyi, V.; Rövid, A.; Remeli, V.; Vincze, Z.; Csonthó, M.; Pethő, Z.; Szalai, M.; Varga, B.; Khalil, A.; Szalay, Z. Towards Cooperative Perception Services for ITS: Digital Twin in the Automotive Edge Cloud. *Energies* **2021**, *14*, 5930. [[CrossRef](#)]
40. Szalay, Z.; Szalai, M.; Tóth, B.; Tettamanti, T.; Tihanyi, V. Proof of concept for Scenario-in-the-Loop (SciL) testing for autonomous vehicle technology. In Proceedings of the IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, 4–8 November 2019; pp. 1–5. [[CrossRef](#)]
41. Horváth, Á.; Tettamanti, T. Test Automation—from Virtual Scenario Creation to Real-world Testing. In Proceedings of the First Conference on ZalaZONE Related R&I Activities of Budapest University of Technology and Economics, Zalaegerszeg, Hungary, 31 March 2022; pp. 48–52. [[CrossRef](#)]
42. Wang, P.; Ye, R.; Zhang, J.; Wang, T. An Eco-Driving Controller Based on Intelligent Connected Vehicles for Sustainable Transportation. *Appl. Sci.* **2022**, *12*, 4533. [[CrossRef](#)]
43. Yan, Y.; Li, H. Machine Vision-Based Method for Estimating Lateral Slope of Structured Roads. *Sensors* **2022**, *22*, 1867. [[CrossRef](#)]
44. Dupuis, M.; Grezlikowski, H. OpenDRIVE®—An open standard for the description of roads in driving simulations. In Proceedings of the Driving Simulation Conference, Paris, France, 4–6 October 2006; pp. 25–36.
45. ASAM OpenDRIVE®Standard. Available online: <https://www.asam.net/standards/detail/opendrive/> (accessed on 22 August 2023).
46. Hegedűs, T.; Fényes, D.; Németh, B.; Gáspár, P. Improving Sustainable Safe Transport via Automated Vehicle Control with Closed-Loop Matching. *Sustainability* **2021**, *13*, 11264. [[CrossRef](#)]
47. Nalic, D.; Pandurevic, A.; Eichberger, A.; Fellendorf, M.; Rogic, B. Software Framework for Testing of Automated Driving Systems in the Traffic Environment of Vissim. *Energies* **2021**, *14*, 3135. [[CrossRef](#)]
48. Gangel, K.; Hamar, Z.; Hány, A.; Horváth, Á.; Jandó, G.; Könyves, B.; Panker, D.; Pintér, K.; Pataki, M.; Szalai, M.; et al. Modelling the ZalaZONE Proving Ground: A Benchmark of State-of-the-Art Automotive Simulators PreScan, IPG CarMaker, and VTD Vires. *Acta Tech. Jaurinensis* **2021**, *14*, 488–507. [[CrossRef](#)]
49. User Manual for OXTS RT 3000 v3 and RT500 models (Revision 231106). Available online: https://www.oxts.com/wp-content/uploads/2023/11/OxTS-RT500-RT3000-Manual_231113.pdf (accessed on 18 July 2023).
50. EuroNCAP TEST PROTOCOL—AEB/LSS VRU Systems, Version 4.4. 2023. Available online: <https://cdn.euroncap.com/media/77299/euro-ncap-aeb-lss-vru-test-protocol-v44.pdf> (accessed on 18 July 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.