CarLink: A Real-Time V2X Validation Platform for a Safe and Sustainable Mobility

Babak Mafakheri^{*†}, Pierpaolo Gonnella[‡], Barbara Mavì Masini^{§†}, Alessandro Bazzi^{*†}

^{*}DEI, Università di Bologna, 40136 Bologna, Italy

[†]WiLab, CNIT, 40136 Bologna, Italy

[‡]FEV Italia, Italy

[§]IEIIT, CNR, 40136 Bologna, Italy

Abstract—Modern vehicles have an ever greater number of automation and connectivity capabilities, which promise extraordinary improvements in terms of safety and comfort, but they also imply new challenges in the design and validation process. In this article, we present CarLink, a new simulation platform with hardware-in-the-loop (HiL), designed and implemented to reduce the time spent on field tests through the emulation of a complex vehicular scenario in a controlled laboratory environment. Specifically, CarLink can simulate a generic traffic scenario and let each vehicle in it communicate with a vehicle under test (VUT), which is actually physically available HiL and equipped with long- and short-range wireless communication capabilities. Communication between simulated vehicles and the VUT is provided by an external management unit (EMU) that integrates the virtual world with the physical one. The architecture is also designed to allow the integration of advanced driver assistance systems (ADAS) testing for the validation of future connected and automated vehicles.

I. INTRODUCTION

With the aim of stopping the endless tragedies on the road, enabling greener mobility, and enhancing comfort, vehicles are increasingly being equipped with advanced sensors and communication devices. As an indication of this, it was recently forecast that in 2024 more than half of all the vehicles being produced will be equipped with some degree of automation,¹ and the market for vehicular connectivity is expected to grow by about 15% yearly from 2020 to 2027.²

This revolution that is taking place in the automotive industry means that car and truck manufacturers are facing new challenges in the design and validation

phase of so-called connected and autonomous vehicles (CAVs). It is, in fact, infeasible to solely rely on time-consuming and costly field tests. Hence, advanced simulation platforms become key elements in the entire production chain, where it is possible to validate and debug the hardware and software developed for what becomes a *vehicle under test (VUT)*. Such simulation platforms are required to reproduce the surrounding environment with the highest possible degree of realism and, at the same time, correctly simulate the connectivity devices, sensors, and applications (hereafter referred to devices and device functionalities, respectively) with which the VUT is being equipped.

The design and implementation of a simulation environment which includes the vehicle hardware and software in the loop and reproduces its interactions with the external world is thus a topic gaining increasing attention from the research community in both academia and industry. The hardware-in-the-loop (HiL) platform needs to deal, in real-time, with a large number of devices and features, including both connectivity and advanced driver assistance systems (ADAS). With regards to ADAS, the platform must include, in particular, cameras, RADAR, LIDAR, and their reaction to changes around the vehicle. When dealing with connectivity, i.e., vehicle-to-everything (V2X) communications, the area of interest expands to include a much greater space, which runs from the neighboring vehicles or other road users and devices, to entire regions under the coverage of cellular systems. In particular, at least two classes of wireless communication systems are envisaged, which we will refer to here as long-range and short-range connectivity. The former refers to classic cellular communications, or vehicle-to-network (V2N), involving the information exchange between the vehicle's on-board unit (OBU) and the base stations, while the latter refers to direct communications with other vehicles and road devices, i.e., vehicle-to-vehicle (V2V), vehicle-to-

¹Source: "Worldwide Autonomous Vehicle Forecast, 2020-2024", International Data Corporation, 2020, <https://www.idc.com/getdoc.jsp?containerId=prUS46887020>

²Source: "Connected Car Market Size and Forecast", Verified Market Research, 2019, <https://www.verifiedmarketresearch.com/product/connected-car-market/>

infrastructure (V2I), vehicle-to-pedestrian (V2P), etc., which are enabled by technologies based on IEEE 802.11p/bd and side-link cellular-V2X (C-V2X).

In this work, we present the design and prototype implementation of a real-time working platform with HiL that is designed to test connectivity aspects related to vehicular applications. Moreover, we introduce future developments toward integration of ADAS devices and functionalities in the VUT. In particular, in Section II, after a thorough discussion of the related works aimed at highlighting the difference and novelty of our proposal with respect to solutions already present in the literature, the overall architecture of the proposed platform is described in Section III. Relevant implementation details regarding short-range and long-range connectivity are presented in Sections IV and V, respectively, and some results are presented in Section VI. Our conclusion and discussion on future lines of work are included in Section VII.

II. RELATED WORK

The rapid growth of CAVs and the fact that large-scale field trials are costly and time-consuming, has given rise to the development of simulation platforms. This kind of platform might be totally software-based or it can include hardware devices in what is known as an HiL approach. It is easy to understand that it does not only reduce the need for the expensive and time-consuming field tests, but it also allows developers to perform a wider set of tests in various virtual environments. The main characteristics of recent works are summarized in Table I, which also highlighting the differences with respect to CarLink.

Focusing on software-based platforms, several simulators have been developed and used by the research community in the last few years and some of the main solutions are discussed in [1]. A sophisticated example of a simulation platform to test both ADAS and V2X connectivity is introduced in [2], in which the authors use the CarMaker vehicle simulator [3] and the robot operating system (ROS) to analyze the performance of ADAS. They introduce accurate modeling of IEEE 802.11p and side-link LTE-V2X for the connectivity aspect. Although the framework allows testing V2X applications when considering real traffic data and vehicle dynamics, it does not include HiL, i.e., the testing of real devices. Hence, this kind of simulator is useful for preliminary studies in large scenarios. However, it cannot be integrated with hardware and used for VUT validation, or adopted in real-time communication environments.

A different approach is the use of simulation platforms with HiL to test real devices and applications in realistic simulated environments through real-time processing capabilities. Interest in this kind of platform has increased recently, but most studies focus only on either communication or automation. For example, the platform presented in [4] adopts the well-known simulation of urban mobility (SUMO) [5] with a VUT virtually interacting through advanced sensors with surrounding simulated vehicles without considering any wireless communications between the vehicles. And the HiL-based platform in [6] takes the real hardware electronic control unit (ECU) of a self-driving vehicle as a part of the simulation platform to verify the validity of self-driving algorithms in virtual scenes, including perception, planning, decision-making, and control but no connectivity is considered.

Only a few HiL platforms designed to test ADAS systems also consider connectivity. One example can be found in [7], where a smartphone in a real car is interfaced with the SUMO simulator, which receives the car's position in real time from the smartphone. At the same time, a number of surrounding vehicles are simulated with SUMO for a particular traffic scenario. In the end, the smartphone receives all the vehicles' information via the cellular interface, which allows the testing of various applications in real time and for large-scale scenarios. In [7] only ideal long-range communications are considered. V2XSim [8] is another HiL-simulator for validating, in a virtual manner, connected and automated vehicles that are moving in a scenario with simulated static and dynamic objects; it relies on ROS for the modeling of the vehicles and on a virtual road side unit (RSU) to collect and redistribute messages that are compliant with the specifications of short-range communications. Although it addresses V2X and related standard messages, it does not include OBUs in the loop and cannot be used to test the hardware part of connectivity.

Given the importance of short-range V2X and the testing of the related hardware, a number of proposals incorporating HiL have been put forward that specifically focus on this aspect. As an example, an evaluation of a safety application, specifically for intersection collision warning (ICW), is performed through a HiL platform called TRUDI in [9]; TRUDI, again relying on SUMO for the vehicular traffic, was designed to test multiple communicating devices adopting short-range communication and does not seem suitable for the validation of the complete set of software and hardware of a connected and automated VUT. Additionally, it

does not include long-range communications. A similar platform, called the HI²LS framework, is proposed in [10]; in this case also, hardware for short-range communications is integrated with software that simulates the virtual position of the devices. Although it does not use a sophisticated traffic simulator, the architecture does appear to be fit for the validation of connected cars, as long as long-range connectivity and ADAS are not of interest. In [11] an approach for creating a reactive HiL testbed is presented, including the OMNET++ and Artery simulators to recreate compliant environments for short-range communication. The testbed reproduces both the mobility of the vehicle and the wireless exchanges in detail, but it encounters difficulties related to the non-real-time nature of the wireless network simulator. In [11], neither long-range communication nor ADAS are addressed.

A HiL simulator with IEEE 802.11p-based devices is implemented in [12], with the same limitations as the above solutions, to test lane-keeping control and cooperative cruise control. In this case, wireless communications are ideal and no losses are considered. A conceptual model that can simulate system-wide changes in traffic parameters on roadways involving both connected vehicles and regular vehicles is introduced in [13]. The authors aimed to consider both static and dynamic objects in a closed-loop HiL simulator in order to study certain important parameters that were missing in the literature at that time. However, the model still lacked a number of realistic features for the analysis of the wireless communication link conditions through dedicated modules.

These are some examples of HiL simulation platforms with the general goal of eliminating costly real-field tests. Although many of these works discuss connectivity and V2X communication links, they all have certain limitations if the aim is to validate the hardware and software of a VUT that is equipped with both ADAS and (short- and long-range) wireless communications devices. As summarized in Table I, unlike the above mentioned works, CarLink makes it possible to verify a VUT's capabilities and performance by considering in detail the impact of the different available communication interfaces and providing real-time processing to test different applications in a large simulated environment with tens of vehicles. Specifically, we characterize in detail long-range cellular communication and short-range communications based on both ITS-G5 and side-link C-V2X. The platform is also designed in such a way as to be ready to integrate ADAS systems.

TABLE I: Summary of the main related works.

References & their Strengths	Difference with CarLink
[2] Florian A. Schiegg et al. A simulation framework, based on CarMaker, to allow automatic testing of ADAS for a large number of real world scenarios.	[2] This platform considers ADAS and short-range communications, but not long-range. It is not integrated with hardware as it is not a real-time simulation.
[4] Marc René Zofka et al. A concept for realistic simulation scenarios with the capability of running at different integration levels.	[4] The platform includes HiL for ADAS only; short- and long-range communications are not included.
[7] Wynita M. Griggs et al. A HiL simulation platform for emulating large-scale intelligent transportation systems with a focus on applications needing feedback from the environment to the vehicle.	[7] A simulation platform with HiL for ADAS that also includes long-range communication. Short-range communication is not considered. The communication link between vehicles/objects is not realistic (e.g., the LOS, NLOS are not considered).
[8] Ethan Zhang et al. An integrated V2X simulation platform built with the Gazebo robot simulation engine focusing on precise vehicle mechanics, an accurate physical description of vehicle movement, and communications in the transportation network.	[8] The platform includes HiL for the ADAS and assumes ideal short-range communications. Long-range communication is not included. It relies on Gazebo, which limits its capability to model realistic traffic/scenario simulations.
[9] Alessandro Bazzi et al. A HiL platform for the analysis and evaluation of safety applications including real ITS-G5 compliant message exchanges, as well as a channel emulator with real signals.	[9] Only short-range communication is considered. Long-range communication and ADAS are not implemented. It lacks modeling of the communication link in realistic conditions (e.g., the LOS and NLOS are not distinguished).
[10] Seokil Song et al. A human-interactive HiL framework for evaluating C-ITS and road safety services allowing the evaluation of road safety system designs in a wide range of virtual traffic environments and road conditions.	[10] Only short-range communication is considered in this HiL-based platform. Long-range communication and ADAS are not included. It evaluates only a very small-scale scenario and the performance in large-scale scenarios is not discussed.
[11] Christina Obermaier et al. A holistic approach for testing devices and applications in a HiL environment to monitor the behaviour of a DUT FOR A WIDE RANGE OF SCENARIOS.	[11] Only short-range communication is considered in this HiL-based platform. Long-range communication and ADAS are not included. There are some constraints on working in real time due to the limitations of OMNET++.
[12] S. Y. GELBAL ET AL. A HiL-BASED SIMULATOR WITH THE GOAL OF DEVELOPING AUTOMATED DRIVING ALGORITHMS.	[12] Designed for short-range communication only. Neither ADAS nor long-range communication are considered. The communication link is assumed to be ideal and link losses are not analyzed.
[13] MOHAMMAD A. HOQUE ET AL. A CONCEPTUAL MODEL OF AN INTEGRATED DISTRIBUTED CONNECTED VEHICLE SIMULATOR (ID-CVS) THAT CAN EMULATE REAL-TIME TRAFFIC IN A LARGE AREA BY INCORPORATING HiL SIMULATION TOGETHER WITH THE CLOSED-LOOP COUPLING OF SUMO AND OMNET++.	[13] This conceptual model only considers short-range communication. Since the model is not implemented, the real-time features are not clear.

III. THE ARCHITECTURE OF THE CARLINK PLATFORM

The CarLink architecture is depicted in Fig. 1; two main parts can be distinguished, namely the vehicle under the test (VUT) in yellow, and the external world emulator (EWE) in green. Both parts include components related to connectivity (blue dashed-dotted rectangle) and ADAS (red dashed-dotted rectangle). More specifically, the VUT includes the hardware and software under test, whereas the EWE is responsible for simulating the virtual environment around the target vehicle in order to provide realistic environment to test the safety applications. The different component of the CarLink is discussed in details in the rest of the section. As can be observed in Fig. 1, ADAS is by design an integral part of the platform, although the main focus of this article is on the connectivity part.

A. Vehicle Under Test (VUT)

The VUT is composed of the following components.

1) *Hybrid control unit (HCU)*: This is, in general, the controller of the mechanical and energy systems of the vehicle.

2) *Vehicle central gateway (GTW)*: This collects information from various components to provide possible formulated indications to the hybrid control unit (HCU).

3) *Telecommunication control unit (TeCU)*: This is the component in charge of connectivity. It is connected to a virtual global navigation satellite system (GNSS) and multiple multi-radio access technologies (mRATs), which include both long-range and short-range communication technologies.

4) *ADAS control unit*: This manages the physical or simulated sensors and reports the information to the vehicle central gateway (GTW).

B. External World Emulator (EWE)

The EWE is composed of the following components.

1) *Scenario/traffic simulator (TRS)*: This simulates the whole scenario in which the VUT is virtually moving, with static and dynamic nodes including roads and buildings, and it updates the virtual GNSS with the position of the vehicles in real time. Since the scenario/traffic simulator (TRS) has a view of the whole environment around the VUT, it plays a crucial role and acts as the master of the entire platform. Therefore all the other components must be synchronized with it. Additionally, since it manages the position of all the vehicles in the scenario, it is also responsible for localization issues by injecting possible inaccuracies into the GNSS data.

2) *Vehicle under the test emulator (VTE)*: This is the component in charge of connecting the TRS to the ADAS of the VUT. Specifically, it receives information from the TRS and details the environment around the VUT to enable scanning from the simulated or real sensors. Moreover, it gives information to the GTW and HCU about the mechanical aspects of the vehicle.

3) *V2X external management unit (EMU)*: This is in charge of determining the availability of V2X links and controlling the V2X communications sent from the EWE to the VUT or vice-versa. In other words, the external management unit (EMU) is designed to model the wireless channel in real time and determine which of the transmitted messages are correctly received. A crucial aspect is the way the EMU is connected to the TRS to determine the result of message exchanges, which can be done in two different ways, as is described

Fig. 1: Block scheme of CarLink platform.

in more detail below. The ability of the EMU to simulate the channel by considering the location of static objects and the real-time positions of dynamic nodes to achieve real-time simulation is the major enabling factor of the platform.

We first introduced a detailed implementation of the EMU that is able to work at run time in [14]. In that work, the EMU was located between the scenario/traffic simulator and the hardware. However, we have extended its capabilities to also work in parallel, which brings further advantages to the whole implementation. We discuss this new approach in the next section.

IV. SHORT-RANGE CONNECTIVITY

The short-range connectivity architecture is illustrated in Fig. 2.

Fig. 2: Block scheme of the connectivity part of CarLink.

A. Integration of TRS with EMU

The EMU can be integrated with the TRS in two different ways: i) it can be located between the EMU

and the mRATs, or ii) it can work in parallel with the TRS and instruct it while the TRS is communicating directly with the mRATs.

1) *EMU between TRS and mRATs*: This architecture is illustrated in Fig. 3(a). In this configuration, the EMU acts as a filter for the exchange of information between the TRS (here represented by SUMO) and the mRATs by withdrawing the messages that are not correctly received by the VUT on the basis of the simulation conditions. Thus, it only passes valid messages on to the mRATs. Furthermore, the EMU is in charge of the calculation of a possible GNSS error to be introduced in the location-aware messages forwarded to the mRATs. The main drawback of this approach is that any processing time required in the EMU to determine the validity of a message or to modify the location fields represents artificial delays added to the end-to-end delay observed from packet generation to packet reception. For this reason, the EMU should process the information in the shortest possible time interval. On the other hand, the advantage of this model is the simplicity of its implementation and integration with the TRS. Examples of solutions for minimizing the processing delay with results can be found in [14].

2) *EMU instructing TRS*: To make CarLink effectively modular and as generally applicable as possible, we also implemented a different approach in which the EMU processes the data in parallel with the TRS, passing on information on message validity so that the TRS could stop invalid messages. This process allows the EMU to periodically inform the TRS about the link quality between the VUT and the other connected objects in the scenario. In this way, on the basis of the information received asynchronously from the EMU, the TRS determines the messages that are correctly received and passes them directly from the TRS to the mRATs as if they were sent from any object in the external world to the VUT, or they are used for the scenario update as if they came from the VUT to any object in the external world. With the same reasoning, the GNSS accuracy calculation is performed within the TRS. This approach allows the two processes to run in parallel, so no artificial delay is added to the end-to-end delay from the wireless communication at the cost of a more complex implementation and the need to move some components (such as the one acting on GNSS accuracy) inside the TRS. The integration of the EMU and TRS with the instructing mode is shown in Fig. 2 and 3(b), with the CANoe-Car2x simulator used as the TRS. The static objects are pre-loaded in the EMU in this case as well, exploiting

(a) EMU between TRS and mRATs.

(b) EMU instructing TRS.

Fig. 3: Possible implementations of the EMU

the Open Street Map platform. However, in contrast to the previous approach, information about dynamic objects is passed from CANoe to the EMU through the use of sockets to let the EMU and TRS work in parallel. As an additional advantage, this approach can be implemented in a multi-processing way, where the various links between the VUT and each of the other connected nodes is managed by a different occurrence of the EMU. This allows the platform to fully exploit the capabilities of multiprocessing.

B. Integration of HiL

The hardware needs to receive virtual GNSS information either: i) directly from the TRS, when the EMU is implemented in the instructing mode (Fig. 3(b)); or

Fig. 4: Long-range connectivity.

ii) from the EMU when it is implemented inbetween (Fig. 3(a)). The information injected into the devices replicates the real GNSS information coming from a real device, such as a global positioning system (GPS). To this end, a GPS data parser is implemented, as indicated in Fig. 2, to obtain the information from TRS or EMU and convert them to the National Marine Electronics Association (NMEA) formatted GPS. In this way, not only is it possible to accurately simulate the real GNSS information known by the devices, but also easier for developers to write software for a wide variety of GPS receivers instead of having to write a custom interface for each one.

V. LONG-RANGE LINK SIMULATOR

Fig. 4 represents the architecture of the long-range connectivity part, which is composed of simulated and real components. In more detail, the following main parts can be observed:

1) *GNSS Simulator*: This sends latitude and longitude information to the components implementing the channel model and to the VUT. Latitude and longitude are used in the channel model to compute the distance between the VUT and the evolved NodeB (eNB).

2) *eNB Location Data-set*: Base station locations are extracted from this data-set and loaded in a list before the execution of the simulation. The location points are used to calculate the distance between the VUT and the eNB, which is useful for the path loss computation and for determining the wireless communication link condition.

3) *Building Data-set*: This data-set contains the coordinates of the buildings possibly impairing wireless communications. The coordinates that are collected can be limited to a defined range in order not to affect the time of the simulation. Such a range should consider

the estimated maximum line-of-sight (LOS) distance including shadowing. The location of the buildings is used to define the LOS and non-line-of-sight (NLOS) condition of the links between the VUT and the eNBs.

4) *Channel Model for uplink and downlink*: This component calculates the received signal used in the simulation of uplink and downlink V2I communications for the given scenario (e.g., highway, rural, or urban), including path loss and shadowing with appropriate LOS or NLOS conditions and correlated shadowing.

5) *Proxy*: This is a server used as an intermediary between the VUT and the Internet services. The Proxy is used as a switch where the channel model determines the correctness of transmission by sending an ACK or a NACK to the Proxy that are called *Proxy Controls*. Depending on the Proxy Controls computed by the channel models, the Proxy allows or forbids communication between the VUT and the Internet services. The packet losses calculated in the simulated part can therefore be reproduced in the real link by means of interruptions commanded via software through the Proxy.

VI. SYSTEM PERFORMANCE

Details of our implementation and the results of the examples we employed are given below, focusing first on short-range communications and then on long-range.

A. Short-Range Communication

For the short-range implementation, we used devices from Cohda Wireless as the OBUs and RSUs. In particular, we used Cohda MK5 implementing ITS-G5 and MK6c implementing sidelink LTE-V2X.

Fig. 5 shows an example of a test of short-range communication. The TRS (implemented here with CANoe-Car2x), the EMU, and the GPS parser run on the host machine. As discussed in Section IV, the EMU filters the valid messages in an instructing mode and the TRS passes them onto the GPS parser. The valid messages are distinguished by finding the link condition (e.g., LOS or NLOS) between the VUT and the other vehicles, calculating the path loss and shadowing, and comparing the received power with the minimum threshold (see Fig. 2). The GPS parser, then, generates the NMEA format strings to be sent to the virtual machine that controls the OBU and the RSU.

For the implementation shown in Fig. 5, we used the MK6c implementing sidelink LTE-V2X. The RSU receives the data for all the vehicles, except the VUT, that are simulated by the TRS through the GPS parser, and transmits them to the OBU through the sidelink

LTE-V2X wireless link. At the same time, the information about the VUT is sent to the OBU, again through the GPS parser. This configuration represents the real environment where a number of vehicles around the VUT are transmitting awareness messages through the C-V2X interface and the VUT receiver attempts to decode these messages. It is worth noting that the RSU in our configuration plays the role of multiple OBUs, thus avoiding the deployment of a huge number of hardware devices.

A human-machine interface (HMI) is then connected to the OBU of the VUT through a WiFi link. In the experiment shown, a forward collision warning (FCW) application is running as an example. As soon as a vehicle in front of the VUT performs a hard brake, the alert appears on the screen of the HMI.

Fig. 5: Implementation of the platform with FCW.

B. Long-Range Communication

When considering long-range communications, the VUT connects to the Internet using the legacy cellular network, and the proxy system described in Section V is used to introduce realistic impairments in the communication. The long-range link simulator, shown in Fig. 4, simulates a radio link for both uplink and downlink communications for each eNB in a range determined by the minimum received power. The received power for long-range communication is calculated on the basis of the path loss and shadowing models defined in Third Generation Partnership Project (3GPP) TR 38.901 and 3GPP TR 37.885, respectively. Initially, the building coordinates are collected from Traci, which is a Sumo library [5], and are then used to evaluate the V2I link

conditions. The base station positions are extracted from a separate data-set (<https://opencellid.org/>) and pre-loaded in the platform for path loss computations. The GNSS simulator sends the latitude and longitude information, in the World Geodetic System 1984 (WGS84) format, to the different components of the simulator, including the VUT. Latitude and longitude are used in the channel model functions to compute distances and model the V2I path loss. The proxy acts as a logic switch: depending on the proxy controls computed by the channel model functions, it allows or forbids communication between the VUT and the internet, both in uplink and downlink. If the channel model functions compute an invalid packet at the VUT (e.g., a received power value below the threshold), the communication is interrupted, otherwise the link remains valid.

Fig. 6 shows an example of the long-range simulation in downlink. The vehicle position is represented with a green circle when the received power at the VUT is above the threshold. On the contrary, the circle is red when the received power is below the threshold. It can be noted that when the vehicle is outside the coverage area, the circles are mostly red, while when it is inside, the majority of the circles are green. As a result of the shadowing effect, a transitory region can be observed in which some losses and some receptions occur. The simple logic described above is defined in order to clarify a solution to establish a control for the correctness of the transmission. This is important because it allows us to recreate, in a laboratory environment, the realistic errors that can be encountered in real traffic scenarios. The errors considered are the ones related to packet losses caused by path loss, shadowing and LOS or NLOS conditions.

Fig. 6: Implementation of long-range link simulation.

VII. CONCLUSION AND FUTURE WORKS

In this work, we have presented the design and implementation of a novel simulation platform with HiL to test CAVs in a controlled laboratory environment, with the aim of solving potential problems before undertaking expensive and time-consuming field trials. The CarLink platform, which integrates short-range (i.e., based on IEEE 802.11p and side-link C-V2X) and long-range (i.e., legacy cellular) wireless communications as well as ADAS components, is based on a simulator of the scenario and vehicular traffic on the premises of the vehicle under test, called TRS, and a simulator of the wireless communication reliability, referred to as EMU. Examples of the tests enabled by the platform have been presented, with a focus on both short-range and long-range communications.

As future work we intend to test advanced applications requiring the integration of wireless communications with Advanced Driver Assistance Systems (ADAS), such as radar, lidar and cameras.

REFERENCES

- [1] J. Wang, Y. Shao, Y. Ge, and R. Yu, "A Survey of Vehicle to Everything (V2X) Testing," *Sensors*, vol. 19, no. 2, p. 334, 2019.
- [2] F. A. Schiegg, J. Krost, S. Jesenski, and J. Frye, "A Novel Simulation Framework for the Design and Testing of Advanced Driver Assistance Systems," in *Proc. of VTC-Fall*. IEEE, 2019.
- [3] I. Automotive, "CarMaker: Virtual Testing of Automobiles and Light-Duty Vehicles," Available at: <https://ipg-automotive.com/products-services/simulation-software/carmaker/>, 2018, Access on July, 2021.
- [4] M. R. Zofka, S. Klemm, F. Kuhnt, T. Schamm, and J. M. Zöllner, "Testing and Validating High Level Components for Automated Driving: Simulation Framework for Traffic Scenarios," in *Proc. of IV*. IEEE, 2016.
- [5] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic Traffic Simulation Using SUMO," in *proc. of ITSC*. IEEE, 2018.
- [6] S. Chen, Y. Chen, S. Zhang, and N. Zheng, "A Novel Integrated Simulation and Testing Platform for Self-Driving Cars with Hardware in the Loop," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 425–436, 2019.
- [7] W. M. Griggs, R. H. Ordóñez-Hurtado, E. Crisostomi, F. Häusler, K. Massow, and R. N. Shorten, "A Large-Scale SUMO-Based Emulation Platform," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3050–3059, 2015.
- [8] E. Zhang and N. Masoud, "V2XSim: A V2X Simulator for Connected and Automated Vehicle Environment Simulation," in *Proc. of ITSC*. IEEE, 2020.
- [9] A. Bazzi, T. Blazek, M. Menarini, B. M. Masini, A. Zanella, C. Mecklenbräuker, and G. Ghiaasi, "A Hardware-in-the-Loop Evaluation of the Impact of the V2X Channel on the Traffic-Safety Versus Efficiency Trade-offs," in *Proc. of EuCAP*. IEEE, 2020.
- [10] Seokil Song, Taein Jeon, Eungjin Kim, Jinsu Kim, Dojin Choi, Yeona Kim, Hyojin Choi, Kyunbyoung Ko, and Cheol Mun, "Demo: Human-Interactive Hardware-in-the-Loop Simulator for Cooperative Intelligent Transportation Systems and Services," in *Proc. of VNC*. IEEE, 2015.
- [11] C. Obermaier, R. Riebl, and C. Facchi, "Fully Reactive Hardware-in-the-Loop Simulation for VANET Devices," in *Proc. of ITSC*. IEEE, 2018.
- [12] Ş. Y. Gelbal, S. Tamilarasan, M. R. Cantas, L. Güvenç, and B. Aksun-Güvenç, "A Connected and Autonomous Vehicle Hardware-in-the-Loop Simulator for Developing Automated Driving Algorithms," in *Proc. of SMC*. IEEE, 2017.
- [13] M. A. Hoque, X. Hong, and M. S. Ahmed, "Parallel Closed-Loop Connected Vehicle Simulator for Large-Scale Transportation Network Management: Challenges, Issues, and Solution Approaches," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 62–77, 2018.
- [14] B. Mafakheri, P. Gonnella, A. Bazzi, B. M. Masini, M. Caggiano, and R. Verdone, "Optimizations for Hardware-in-the-Loop-Based V2X Validation Platforms," in *proc. of VTC-spring*. IEEE, 2021.

Babak Mafakheri is a senior research engineer at Safran Passenger Innovations, Germany. He received his Ph.D. degree from the University of Bologna in collaboration with the Fondazione Bruno Kessler (FBK) Research Center in Trento, Italy. He also worked as a Postdoc at the University of Bologna in collaboration with FEV Italia Srl for almost 1.5 years. His research interests include connected vehicles, the coexistence of wireless technologies in heterogeneous networks, and blockchain technologies. He has served as a reviewer for several journals and conferences.

Pierpaolo Gonnella received the M.Eng. degree in telecommunications engineering from the University of Bologna, in 2020, with a thesis entitled "Development of a V2X platform for predictive driving functions validation" executed at FEV Italy in collaboration with Green Mobility Research Lab, Bologna, Italy. He became project engineer responsible of connectivity research and development for lab building at FEV Italy, in 2019 and also joined FEV cyber security team in 2021.

Barbara Mavi Masini (S'02-M'05-SM'19) is a Senior Researcher at CNR-IEIIT, adjunct Professor at the University of Bologna and member of CNIT/Wilab. The research is mainly focused on connected and automated vehicles (CAVs) where she acts as keynote speaker, tutorial instructor and coordinator of national and international research projects. She is editor of IEEE Access and Computer Networks.

Alessandro Bazzi (S'03-M'06-SM'18) is a Senior Researcher at the University of Bologna and member of CNIT/WILAB, after being with CNR for more than 15 years. His work mainly focuses on connected and autonomous vehicles (CAVs). On CAVs, he won a best paper award, he acted as keynote speaker, tutorial instructor, panelist, or workshop organizer at major IEEE conferences, and he contributes to ETSI activities. He is Chief Editor of Hindawi Mobile Information Systems.