

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A Hybrid Vehicle Hardware-in-the-Loop System with Integrated Connectivity for eHorizon Functions Validation

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Brunelli L., Capancioni A., Gonnella P., Casadio R., Brusa A., Cavina N., et al. (2021). A Hybrid Vehicle Hardware-in-the-Loop System with Integrated Connectivity for eHorizon Functions Validation. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, 70(5), 4340-4352 [10.1109/TVT.2021.3073807].

Availability:

This version is available at: <https://hdl.handle.net/11585/837848> since: 2021-11-10

Published:

DOI: <http://doi.org/10.1109/TVT.2021.3073807>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

A Hybrid Vehicle Hardware-in-the-Loop System with Integrated Connectivity for eHorizon Functions Validation

Lorenzo Brunelli, Alessandro Capancioni, Pierpaolo Gonnella, Rebecca Casadio, Alessandro Brusa, Nicolò Cavina, and Michele Caggiano

Abstract — Urbanization led to an increasing number of vehicles on the roads, resulting in more polluted air and more congested urban centers. This is being mitigated by the Hybrid Electric Vehicles equipped with telecommunication devices, which allow the implementation of predictive control strategies. This research is focused on the setup of an innovative and universal simulation environment for the development and the validation of predictive control strategies supported by Vehicle-to-Everything (V2x) connectivity. This helps the testing and validation of predictive control strategies, granting safety, reliability, and reproducibility. The simulation environment consists of a connected Hardware-in-the-Loop (HiL) system to test a supervisory controller (Hybrid Control Unit) where the predictive functions will be implemented. In addition to all the advantages of a conventional HiL layout, it can exchange real data from cloud service providers and nearby devices. The over-the-air interfaces between the powertrain controllers, the cellular network, and the Intelligent Transportation Systems (ITS-G5) are handled using a custom connectivity control unit with proprietary functionalities. Finally, this work presents the testing of the end-to-end communication for both the short- and long-range data exchange between real controllers.

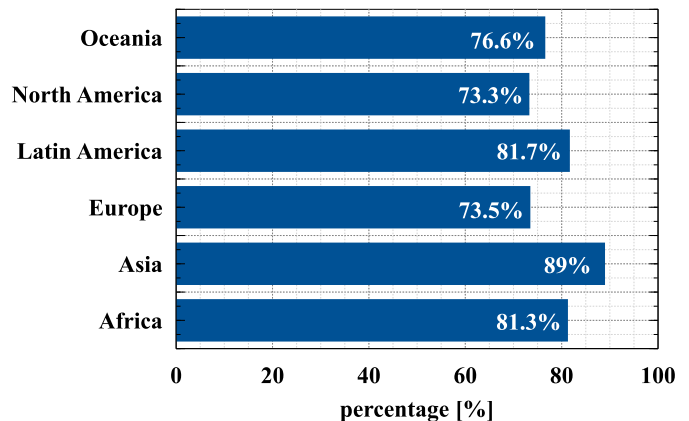
Index Terms — Connected vehicles, eHorizon, Hardware-in-the-Loop (HiL), Hybrid Electric Vehicles (HEVs), predictive strategies, Vehicle-to-everything (V2x)

I. INTRODUCTION

STARTING from the last decades of the twentieth century the industrialization has seen exponential growth that has led to a massive migration of people from the rural areas to the urban centers. In [1] and in Fig. 1 it is reported that, in 2016, 5 billion people lived in urban areas and this number is projected to increase to 7 billion by 2050 [2].

One of the most cited consequences is the worsening of urban mobility and air quality. In fact, the amount of CO₂ emitted per capita is now more than 4 tons per year and 21% comes from the transportation sector [3], which led to stricter regulations in terms of emitted pollutants per driven kilometers. As an example, the EU sets the limit of 95g/km of CO₂ for 2020 and 80g/km for 2025 [4]. Thus, manufacturers are responding with innovative solutions such as more efficient conventional engines and the introduction of Hybrid Electric Vehicles (HEVs).

In this framework, there is also the need to re-organize the



Source: EU commission, Atlas of Human Planet (2016)

Fig. 1 Urbanization percentage by continents

management of urban mobility to optimize traffic flows to decrease congestion. The most frequently applied measures are urban centers with increasing traffic limitations, regarding both the conventional vehicles (Low-Emission Zone – LEZ) and the non-conventional vehicles (Zero-Emission Zone – ZEZ) [5].

Moreover, innovative technologies are being implemented both on the vehicle and the infrastructure, such as wireless communication, cloud computing, innovative sensors, and artificial intelligence functions (as computer vision). These technologies result in vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N) communication.

Together with the latest Advanced Driver-Assistance Systems (ADAS), these technologies allow calculating an electronic horizon (shortly known as eHorizon) which represents a virtual reconstruction of the trip ahead for a planned route. It is conventionally divided into “short horizon”, which comprehends information about nearby vehicles and traffic lights, and the “long horizon”, which includes information about the selected route, the slope, and the speed limits. Such information can be used to manage the available energy more efficiently and sustainably. For example, regarding single HEV strategies, if the electronic control unit (ECU) would be

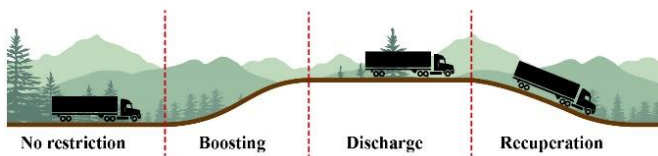


Fig. 2 Example of a predictive control strategies in front of a hill

informed about the presence of a hill in the route ahead, it could decide to use the power from the battery during the uphill phase and to recharge it during the downhill one, as shown in Fig. 2 and described in [6]. Other remarkable single-vehicle strategies involve the eco-driving based on information coming from the V2x or the determination of the optimal (or sub-optimal) power split. These kinds of functions can be also applied to double or multiple vehicles such as the Adaptive Cruise Control (ACC) based and the global connected energy management strategies. The information coming from the V2x connectivity is also used in [7] along with neural networks to forecast the velocity to use it as the basis for an equivalent consumption minimization strategy, or to feed a real-time HEV optimization method with traffic prediction as in [8] through Pontryagin's Minimum Principles and in [9] through a Nonlinear Model Predictive Controller. A deeper overview of the predictive energy management state-of-art is given in [10].

However, the development and validation of such complex functions collide with the shortening of the vehicle time-to-market. A recent survey shows that 68% of the automotive companies have now a product development and launch cycle under two years [11]. The old consolidated methodologies (reliable road tests above all) do not fit anymore because the safety validation of a current driver assistance system alone requires up to 2 million test kilometers [12]. Besides, most of the functions under test regard dangerous and highly variable conditions that are difficult and expensive to replicate with conventional tests.

Thus, the tendency is to move the driven kilometers from the road to advanced simulation environments in a process called Road-to-Rig-to-Desktop. Among all the advantages of a virtual

test, there is the possibility to control every variable, granting repeatable tests especially for conditions hard to recreate in the reality.

To respond to this problem, an innovative Connected Hardware-in-the-Loop (C-HiL) universal simulation environment has been developed and it is presented in the paper.

The architecture of the C-HiL is briefly described in the following paragraphs and its schematic representation is provided in Fig. 3. In particular, a detailed and already validated vehicle model is implemented on a Real-Time PC. It includes sub-models for the vehicle longitudinal dynamics, the hybrid powertrain components, and their controllers. The Real-Time PC is connected to a rapid-prototyping Hybrid Control Unit (HCU) running an advanced energy management supervisory controller modeled in MATLAB & Simulink. This controller derives from a fully working high-performance hybrid technology demonstrator developed by FEV Europe and it runs the same software, thus allowing the validation of the control functions as if it were installed in the real vehicle. As it is the hardware within the testing loop and so where the functions will be implemented, it is marked with the red block in Fig. 3.

Then, the HCU is connected to another rapid prototyping control unit that works as a Central Gateway (CG). Principally, it prepares and forwards all the eHorizon-related data to the HCU (from the network and the external On-Board Units (OBU)), while sending back to the server the GPS coordinates generated by the GPS and Traffic Simulator. The latter runs on another PC, named Sim PC, in charge of all the environment and radio link simulations.

The V2x communication is handled by a Telecommunication Control Unit (TeCU) using the Multi Radio Access Technology (MultiRAT). The latter enables the data exchange with external real servers via Long Term Evolution (LTE) and with RoadSide Unit (RSU) using a commercial OBU (implementing ITS-G5 protocol stack). All the hardware within the light-grey box in Fig. 3, namely the HCU, the CG, the TeCU, and the OBU, will be installed in the vehicle.

Thus, it is possible to feed the HCU control functions with

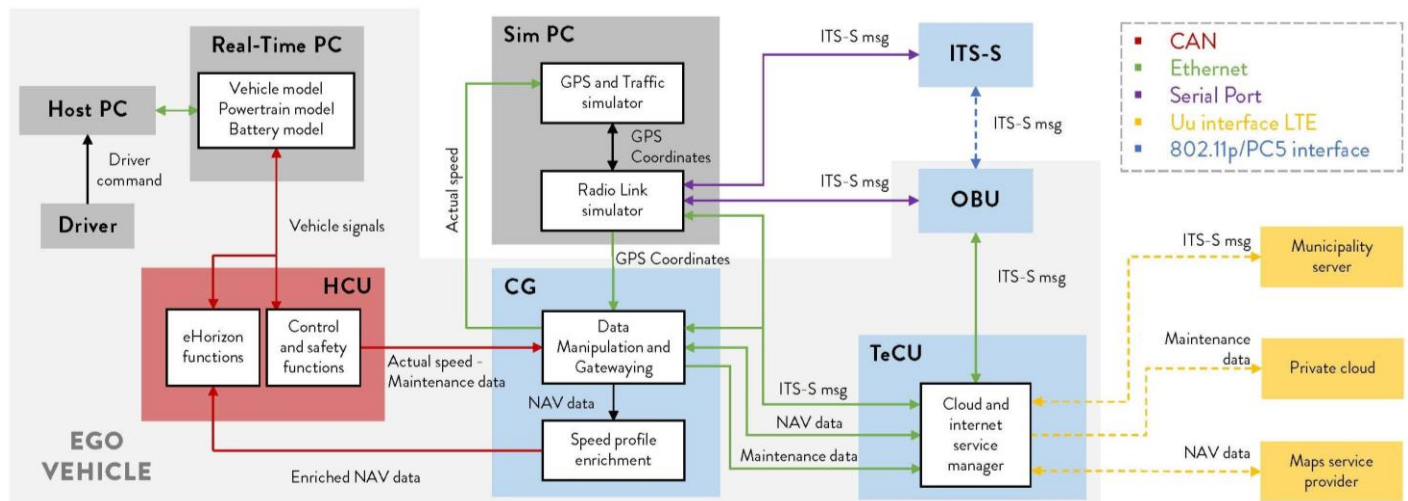


Fig. 3 Connected HiL system representation

realistic data sent through a real communication protocol laying the ground for predictive strategies test and validation, with a focus on:

- *eHorizon predictive control strategies*, which will need an accurate representation of the surrounding environment (short and long horizon). The functions that will initially be tested are long horizon predictive functions such as thermal and energy management. Short-range predictive energy management functions tests will follow in future works;
- *cloud maintenance functions*, which will carry out fleet management using the maintenance data calculated and stored in the vehicle control units. As part of this project, a cloud-based function is being developed to predict the **remaining** State of Health (SoH) and **Remaining Useful Life (RUL)** of the vehicle battery, based on measurements of the **last all** driving cycles.

The other sections of this work are divided as follows. In Section II a detailed analysis of the state of the art of connectivity and controllers testing simulation has been performed. Then, the complete architecture of the proposed system is described in detail in Section III focusing on each element, and on how they operate and communicate. Finally, Section IV focuses on the tests of the different communication protocols for each case study, and the main conclusions are drawn in Section V.

II. RELATED WORKS

The validation and verification of connectivity-related functionalities are becoming more demanding as they must be tested in a huge number of scenarios, regarding dangerous (e.g., Emergency Brake) and highly unpredictable situations (e.g., Collaborative Adaptive Cruise Control and connectivity-related functions) to be declared reliable. The challenge is therefore to develop a testing and validation framework that can replicate the effectiveness of road conditions and traffic scenarios. In this way, it is possible to transfer the tests from the road to the virtual simulation, saving costs and time. Following this tendency, specific commercial software were developed and made available on the market, such as PreScan and ITS Modeller presented by Tideman and van Noort in [13], whose aim is to provide the automotive industry with a tool for developing connected vehicle systems from concept to production. Always in a virtual environment, Aramrattana et al. [14] presented a simulation framework (consisting of driving, traffic, and network-simulators) for testing and evaluating Co-operative Intelligent Transport Systems (C-ITS) applications.

Once the simulation of vehicles and networks in a virtual environment is established, Xu and Shen [15] improve it with optimal energy management, which is tested using the short horizon information coming from the leading vehicle in the collaborative environment.

Grahle et al. [16], from Bosch, used this simulation framework to evaluate the advantages of a route preview (long horizon) in order to determine in which part of the trip it would

TABLE I
REMARKABLE STUDIES ON SIMULATION SYSTEMS

Reference	Vehicle model	Long horizon	Short horizon	Cloud computing	Use cases	Major innovations
Tideman and Van Noort [8]	No	No	Simulated	No	C-ITS	Software for designing and evaluating connected vehicle systems
Aramrattana et al. [9]	Yes	No	Simulated	No	C-ITS (CACC) ^a	Simulator to involve human driver
Xu and Shen [10]	Yes	No	Simulated	No	C-ITS (CACC)	Optimal energy management with V2V data
Grahle et al. [11]	Yes	Simulated	No	No	eHorizon functions	Demonstrates the advantages of route information in DPF regeneration
Hopka et al. [12]	Real vehicle	Yes	No	No	eHorizon functions	DPF regeneration strategy with prediction of traffic and route information on-board
Menarini et al. [13]	No	No	Yes	No	C-ITS (ICW)	HiL system for developing connectivity controllers ready for the road
Szendrei et al. [14]	No	No	Yes	No	C-ITS (dynamic rerouting)	HiL system with V2X OBU/RSU support and real C-ITS protocol stacks and APIs
Gelbal et al. [15]	Yes	No	Yes	No	CAVs algorithms	Highly realistic HiL system for connected and autonomous driving functions
Shao et al. [16]	Real vehicle	No	Yes	No	C-ITS (CACC and emissions)	Engine-in-the-Loop that replicates a real vehicle on the road
Kim et al. [17]	Yes	No	Yes	Yes	CAVs algorithms	Connected HiL for developing and testing of on-board and in-cloud automated functions
Our work	Yes	Yes	Yes	Yes	eHorizon functions, CAVs algorithms	Modular Connected HiL for developing and testing of on-board and in-cloud predictive eHorizon functions

^a Between parenthesis there is the specific Intelligent Traffic System analyzed in the article

be more convenient to regenerate the Diesel Particulate Filter (DPF). Likewise, in Ford, Hopka et al. [17] translate that example on a real prototype controller mounted on a prototype vehicle. On the other hand, the short horizon communication is not implemented in this contribution.

A similar on-road test is proposed by Menarini et al. [18] where a short-range wireless communication is tested with an Intersection Collision Warning function and then verified on a real vehicle. Symmetrically, the long horizon is left apart as well as the vehicle dynamics. An evolution of that work is presented by Szendrei et al. [19] where several OBU/RSU hardware have been connected to a microscopic traffic simulator (Simulator of Urban Mobility - SUMO) to integrate real vehicular communication devices. Anyway, it focuses on the C-ITS simulation for automated vehicles while it does not simulate the vehicles' dynamics nor the long horizon connectivity. Simulation frameworks like these are used also for Automated Vehicles functions development as shown by Gelbal et al. [20], who set up a Hardware-in-the-Loop simulator for developing automated driving algorithms. In this case, the Real-Time PC carries out the simulations of other moving vehicles while also generating traffic scenarios while the prototype control unit runs the algorithm. However, the long horizon is not simulated even in this work. A step further was made by Shao et al. [21] who developed an Engine-in-the-Loop system integrated with a real-time traffic simulator (named VISSIM), to evaluate the performance of emerging connected vehicle applications. This allows a systematic evaluation of connected vehicle mobility and energy savings, as emissions and fuel consumption can be measured precisely. Then, a real vehicle equipped with an OBU is driving along with other connected vehicles. That vehicle data is transmitted to the HiL, which reacts consequently. There is nothing concerning the long horizon, and, besides, it has been assumed that perfect communication is available between vehicles. Finally, Kim et al. [22] focused on the development of a sustainable framework for testing control strategies for Connected Automated Vehicles (CAVs). They presented an HiL where vehicle dynamics are up to ETAS DESK-LabCar, controlled by on-board ECUs i.e., MicroAutoBox and Matrix embedded PC-Adlink. The latter oversees the communication with the OBU and the cloud, respectively through Ethernet and LTE. The environment and the perception sensors are simulated with PreScan while the micro-traffic with PTV VISSIM. Such an advanced simulation framework is very interesting, but it has been presented with a short driving routine. The overmentioned state of the art is summarized in Table I.

With respect to the aforementioned published works, this one counts on a highly detailed vehicle model (validated over experimental data) supervised by the production level HCU software unlike those presented in [13], [18], [19], with complete access to the components and controllers models. The HCU is then equipped with the typical V2x communication technologies, both for the long horizon such as in [16], [17] and the short horizon such as in [18]–[22]. Moreover, the connection to the private server enables the testing of cloud computing and predictive maintenance functions, as in [22]. So,

the proposed HiL-based validation platform results in a more modular and universal tool for testing and validating predictive eHorizon functions. On one hand, it provides higher flexibility due to the possibility to test a different kind of predictive functions (long and short horizon, predictive maintenance) and the capability of acting on each component and controller model. On the other hand, it grants higher reliability, as both the hardware and the software are the same as those implemented on the vehicle, allowing seamless functions implementation on-board once validated at the HiL. Thus, it will shorten the validation process and further reduce the gap between laboratory and on-vehicle tests.

III. SYSTEM DESCRIPTION

The system components, shown in Fig. 3, are described in the following paragraphs, starting from the conventional HiL framework and the CG description, up to the TeCU platform and the server layout.

A. Vehicle model

The prototype vehicle under test is a Plug-in Hybrid Electric Vehicle (PHEV) equipped with a V10 5.2 liters FSI engine able to provide 533 Nm at 6500 rpm that can rely on a pure electric front axle (P4x2), able to guarantee the four-wheel drive (4WD) to be independent by the state of charge (SoC) of the battery. This front axle is equipped with two high-torque electric machines powered by the Integrated Starter Generator (ISG, P1). With respect to the original prototype depicted in detail in [23], in the actual one the electric axle is powered by a high-voltage battery with a 2p93s layout, able to provide 19.4 Ah of

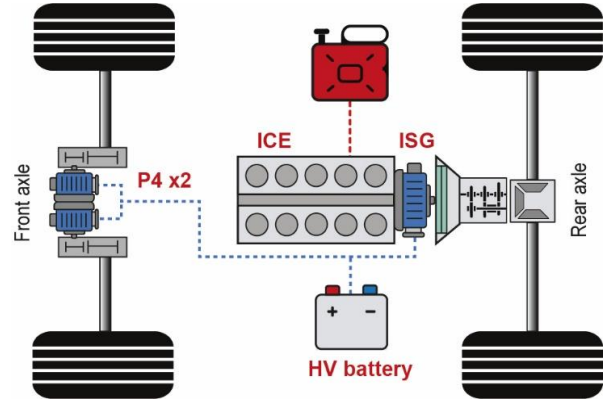


Fig. 4 The P1-P4 hybrid architecture

TABLE II
VEHICLE POWERTRAIN DATA

Component	Characteristic
Battery nominal capacity (1C @25°C)	19.4 Ah
Battery nominal / max. voltage	384 / 391 V
Motor continuous / peak torque	145 / 350 Nm
Motor continuous / peak power	64 / 140 kW
Engine max. torque	533 Nm
Engine max. power	449 kW
Overall max. power (cont. / peak)	577 / 729 kW
Transmission	6-Gears

total capacity (at 1C rate and $T=25^{\circ}\text{C}$) and 384 V nominal voltage (at $\text{SoC}=50\%$ SoC and $T=25^{\circ}\text{C}$). The conventional powertrain is completed with a 6-speed gearbox. The propulsion system is depicted in Fig.4 and the vehicle's main characteristics are listed in Table II.

Even if it is just a technology demonstrator, the vehicle had to run several validation processes, both at a component level and as a whole vehicle.

From the simulation side, the prototype is entirely modeled in MATLAB & Simulink. The components' models are described in Paragraph 1) while the controllers' models in Paragraph 2). All of them have been validated over experimental data and the results of the validation are shown in Paragraph 3). All the vehicle's models are running into a dSPACE SCALEXIO while the HCU Controller into a dSPACE MicroAutoBox.

1) Powertrain

In order to represent the vehicle dynamics of the given 4WD powertrain, its main components, namely the Internal Combustion Engine (ICE), the Electric Motors (EMs), the Automatic Manual Transmission (AMT), and the wheels, have been modeled with Simscape mechanical libraries. Simscape is a tool for modeling and simulating multidomain physical systems within the Simulink environment by means of physical connections.

Then, the electrical and thermal control-oriented models of the high-voltage battery have been developed. The electrical model is represented by a single-polarization equivalent circuit model while the 0-D thermal one relies on the use of a single thermal mass to reproduce the thermal behavior of the whole battery pack. In this case, contributions to the net heat flow rate are given by power losses (due to Joule effect), air-battery convective heat exchange, and coolant-battery convective heat exchange. The latter is calculated using a simplified cooling circuit model in which the main actuators, namely the high-voltage compressor and the electric pump, have been inserted via experimentally derived characteristics respectively in the refrigerant and coolant loop of the system. In order to reproduce the signals of Current-Voltage Sensors (CVS) and temperature sensors used by both the onboard and the virtual Battery Management System (BMS), the voltage of the cell and the temperature of the battery evaluated by the model have been replicated for each sensor.

The electric propulsion system is completed with the DCDC converter and the Integrated Power Unit (IPU), which are modeled as energy conversion efficiencies.

The vehicle is able to follow a given driving path thanks to a driver modeled as a calibrated PID controller that produces the accelerator and brake pedals position as outputs. All the driver's decisions (tip up, tip down, accelerate, brake, driving mode) can be overwritten by an external source as seen in Fig. 3. This allows introducing unpredictability in driver behavior, which is a useful feature for control functions' validation.

All the Simulink models are proprietary of FEV Italy and so fully accessible, thus simplifying the development of the

functions.

2) Controllers

The controllers whose simplified models have been implemented in the Real-Time PC are:

- ECU: it receives from the HCU the torque request and the start and stop command. Then, it is in charge of controlling the Internal Combustion Engine (idle control and smoothing the output torque);
- TCU: it controls the shifting phases (clutch opening, torque reduction, engagement, clutch closing);
- The IPUs, the DCDC, and the ISG controllers introduce torque and power limitations, along with the respective efficiencies.

Then, the two software-level control unit models are described below: the vehicle supervisor (HCU) and the battery control unit (BMS).

a) Hybrid Control Unit

The prototype supervisor, named Hybrid Control Unit, manages all other ECUs. The HCU coordinates the requests to the powertrain's subsystems, in order to guarantee the vehicle's performance and all the safety requirements as:

- electric traction (without the need to add gearbox or clutch controls);
- front axle torque vectoring (without any electronic differential system);
- 4WD control;
- torque fill during shifting (shift assist system);
- boosting function.

In particular, regarding the 4WD control, the HCU calculates the driver torque request which is mapped as a function of the vehicle speed and the throttle pedal. When the latter is less than 5% and the vehicle is moving, the supervisor control interprets this condition as a negative torque request, which is actuated by electric motors operating as generators (and thus regenerative braking is performed). If the whole negative torque can not be supplied by the only motors, then even the mechanical braking takes place (and thus blended braking is performed). In all the other conditions, positive torque is demanded. In this case, a rule-based energy management strategy evaluates the torque split between the ICE and the EMs. This strategy is based on rules depending on fixed thresholds (which have been previously calibrated). In addition to the driver torque request, parameters such as the vehicle speed and the state of charge of the HV battery are involved in the torque split factor evaluation. Thus, the controller firstly discharges the battery and then keeps the state of charge around the chosen threshold. Such an approach is commonly referred to as the charge-depleting/charge-sustaining approach and is typically used for PHEVs, like the one used for this activity. Since this kind of control logic is characterized by reliability and easily real-time implementation due to the low computational burden required [24], it is currently the on-board standard strategy.

The software of the supervisor controller has been written in MATLAB & Simulink and downloaded into a dedicated rapid prototyping control unit (MicroAutoBox by dSPACE GmbH).

Even in this case, the software is proprietary of FEV Italy so fully open to modifications and functions implementation. The software and the hardware are the exact replica of those mounted on the prototype, so the developed predictive functions can be seamlessly implemented on-board once validated at the HiL.

b) Battery Management System

The application software of the on-board Battery Management System (BMS) has been implemented in the Real-Time PC in the present configuration, but the Connected-HiL system is designed to integrate the corresponding hardware component seamlessly.

This highly detailed model has most of the functionalities of the real software, such as

- contactors control (for battery pre-charge);
- isolation monitoring;
- power limitation;
- system diagnostics.

Particular attention has been paid to the latter because error monitoring plays a vital role in preserving the safety of the vehicle's components and the passengers. Moreover, regarding the present work, the implementation of the real HCU in the HiL facility has enabled the testing of this functionality before implementing on-board any control function. Therefore, the BMS is able to recognize if threshold values for monitoring are exceeded for several important parameters, such as battery voltage, current, power, and temperature. High-voltage interlock lines and battery isolation status are monitored, as well. In case of error, the BMS sends this information to the HCU which takes remedial action depending on the type and severity of the error. If such an action is not applied within a certain time, the BMS itself takes over control and opens the high-voltage relays. At this aim, the battery pre-charge circuit comprehending the contactors has been implemented in the battery model.

3) Validation

The whole vehicle model has been validated with experimental data acquired:

- during an All-Electric Range test, that consists of consecutive NEDC cycles, repeated until the battery is fully discharged and the ICE starts up. So, this test is focused on the validation of the electric propulsion only;
- on a rural road drive that consists of driving the prototype vehicle in conventional mode keeping the battery switched off for the validation of the conventional powertrain.

The results of the validations are shown in Fig. 5 and in Fig. 6, respectively. The ability to correctly predict experimental data is demonstrated by the graphs, and it proves to be accurate enough for a control-oriented modeling approach.

B. Central Gateway

The Central Gateway (CG) collects, elaborates, and exchanges the data between each control unit, ensuring proper

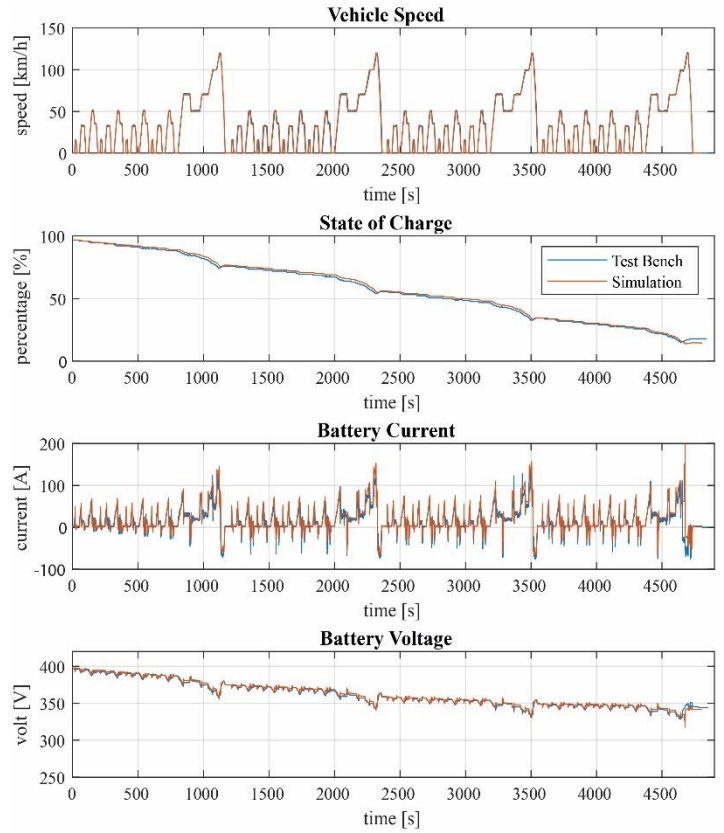


Fig. 5 Electric powertrain validation (blue line: experimental data; red line: simulation data)

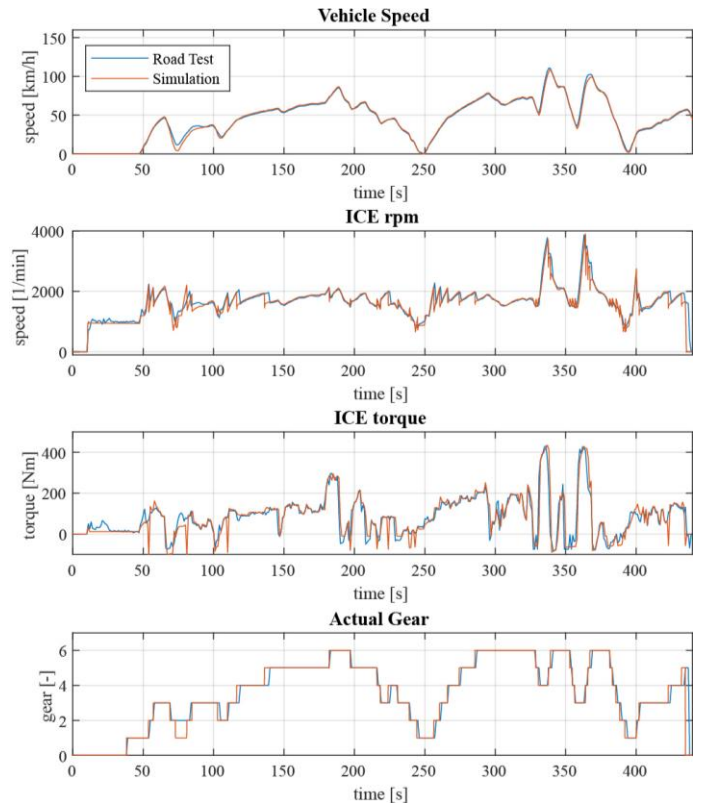


Fig. 6 Conventional powertrain validation (blue line: experimental data; red line: simulation data)

and effective communication. The selected hardware is Miracle2 which is a compact and fully programmable control unit suitable for rapid control prototyping and smart data acquisition [25].

Regarding the eHorizon functions validation, the CG has to provide the HCU with the prediction of the navigation data. At this aim, firstly the CG has to simulate the onboard navigator by forwarding the vehicle's actual position and the desired destination to the TeCU. The latter communicates with the map provider and sends back the navigation data to the CG as an XML file (the communication is described in Paragraph C).

Then, the navigation data has to be enriched and forwarded to the HCU. To do so, the CG receives from the OBUs, through the TeCU:

- 1) Cooperative Awareness Messages (CAM)
- 2) Signal Phase and Timing (SPaT) messages
- 3) Decentralized Environmental Notification Messages (DENM)

that are standardized messages for the cooperative vehicular communication systems, as described in [26]. In this work, the CAM, SPaT, and DENM messages will include short horizon data such as the next traffic lights phases and timing or the presence of a moving vehicle approaching the same intersection of the ego vehicle.

Moreover, the CG communicates with GPS and traffic simulator that is SUMO (Simulation of Urban Mobility), an open-source traffic simulator [27]. SUMO simulates the ego vehicle moving around the city, the position in GPS coordinates of other moving vehicles (some of them equipped with other simulated OBUs), and the position of RSUs. In particular, SUMO sends the GPS coordinates and the heading of the ego vehicle to the CG for the request to the map provider which requires ego vehicle updated information. Moreover, SUMO simulates and also sends the coordinates, the actual status, and

the duration of the predicted stop events such as traffic lights and right of ways.

Then, the CG processes this information to enrich the navigation data. The enrichment of the speed profile is composed mainly of two parts. In the first part, the static speed limits coming from the map provider is integrated with the stop events. In the second part, the updated speed profile is modified with the introduction of acceleration and deceleration phases. In parallel, accelerations that are not feasible for the ego vehicle are removed. Finally, the CG sends such information to the HCU via CAN at the beginning of each trip and every time there is a request for a new or updated speed profile, using the existing communication layout of the prototype vehicle.

For what concerns the in-cloud maintenance functions validation, the CG sends the vehicle maintenance data coming from the HCU to the TeCU via TCP. Then, the data are sent to the private server and elaborated by the proprietary cloud-based functions.

C. Telecommunication Control Unit

In the considered architecture, the role of handling the diverse Radio Access Technologies is taken over by a single board computer. The latter is capable of exchanging data using the different protocols enabling the typical V2x connections. The messages typically sent in eHorizon scenarios are summarized in Table III and, for each message, the requirements to be fulfilled by the system are listed accordingly. The message types have been classified based on specific data required for the use cases mentioned in I.

In Table III, the upload and download flow needed for each message is also specified. The reference device is the ego vehicle, so the DL (e.g., I2V) refers to incoming flow and UL (e.g., V2N) for outgoing one.

The update frequency depends on the type of application, so it is only indicated in a qualitative way (high and low). In the considered case, the shorter the eHorizon the smaller the update

TABLE III
SYSTEM REQUIREMENTS AND COMMUNICATION PROTOCOLS

Use Case Category	Message type	UL/DL	Update frequency	Payload size	Required Throughput	Remote server	Protocol stacks
eHorizon	Navigation data	UL/DL	event driven*	Mbytes	HIGH	MSP	LTE
	Traffic data, traffic lights phases	DL	event driven*	Mbytes	HIGH	MSP, M, R-ITS	LTE, ITS-G5***, WAVE***
	ZEZ topological data	DL	event driven*	Kbytes	HIGH	MSP	LTE
	Distance from traffic light, intersections, vehicles, moving objects, road hazard	DL	~100 ms	>400 bytes	HIGH	R-ITS / V-ITS	ITS-G5 ***, WAVE***
Predictive Maintenance	Vehicle maintenance data	UL	event driven**	Kbytes	LOW	PS	LTE

UL = Uplink; DL = Downlink; MSP = Map Service Provider; M = Municipality; R-ITS = Roadside ITS; V-ITS = Vehicle ITS; PS = Private Server

* wrong way

** every key on

***802.11p, PC5 (LTE-V2x) at PHY layer

frequency in terms of time intervals. In some cases, it is not possible to determine this parameter because it can be very high or very low depending on unpredictable situations like in the case the driver takes the wrong direction and the navigation data must be updated.

The estimation of the payload size for the long horizon has been obtained considering the size of files typically used for navigation data, such as KML, XML, JSON files. The short horizon data have a payload size in the same range of CAM, DENM, SPaT messages which can be about 200-400 bytes. The required throughput has been estimated exploiting update frequency and payload size reported in each entry. Thanks to such estimation, each message, and therefore each link, is associated with a V2x connection. Each server is interrogated to retrieve the information needed for enabling the eHorizon functionalities. In particular, communication is established with:

- Map Service Provider: it provides the route to follow, the speed limits, and the slope profile along that route. Other information given by the map provider is the ZEZ topological limits.
- Private Server: it is specific for OEM functions implementation. Its exploitation in the project is explained in section IV paragraph *C. Cloud-based control functions*.
- Municipality Server (or platform): it is strictly related to the concept of the smart city where connected vehicles collect and share real-time data about traffic, pedestrians, surrounding vehicles, and cyclists.

Then, the relative protocol stack has been defined for each message type. For this analysis, only technologies available on the market have been considered (e.g., 5G-NR is not considered yet). The LTE is a long-range radio access technology with relatively low latency, useful, for in-vehicle implementation, to communicate with diverse internet services. LTE-V2x refers to the device-to-device version of the LTE (Mode 4) used for vehicular communication. It uses the PC5 air interface, while LTE indicates the use of the Uu air interface (Mode 3), as explained by the standard definition in [28].

The system known as ETSI ITS-G5 was developed by the ITS technical committee referring to the previous US project, the Wireless Access in Vehicular Environments (WAVE). The WAVE project defined the changes to the IEEE 802.11 standard (the one behind the Wi-Fi products) to support the requirements of vehicle transport systems, producing the so-called IEEE 802.11p version.

Therefore, a single-board computer has been selected to implement the TeCU. This control unit constitutes a multi-RAT technology: it works as a router for the In-Vehicle Network (IVN) and it also provides some grade of automation at the application layer. The hardware used is APU3C4 from PC Engines [29] which mounts a Linux-based operating system and its functionalities have been implemented with Python. At the IVN side, it is connected via ethernet to the CG, it uses TCP/IP and UDP/IP at the transport layer, and SFTP for file exchange. On the external network side, it exploits an LTE UE for internet access, it uses Message Queue Telemetry Transport (MQTT), and a custom TCP/IP based protocols for data exchange with internet services that will be described in

Paragraph IV.B. For short-range communications, it is connected via serial port to an OBU equipped with ITS-G5 stack, so that all the ITS information is passed to the IVN through the TeCU.

IV. DATA FLOW TESTING AND USE CASES

The main purpose of this part of the project is to test the end-to-end communication for each protocol. The first test that will be presented is the communication to the HCU of the enriched navigation data that are necessary for the validation of predictive control strategies. The flowchart of the test is depicted in Fig. 7 and described in the following paragraph. Then, the communication with the private server is tested, where proprietary predictive maintenance functions run, and, in this case, the communication follows the flowchart of Fig. 14. Finally, a possible use case is presented in Fig. 16.

A. Enriched navigation data communication

The eHorizon control functions require the prediction of the navigation data to be as accurate as possible. So, as shown in Fig. 7, the CG performs the enrichment of the static navigation data (static speed limits, slope, and ZEZ topological information) with the stop events identified by SUMO from the map. Then, the CG receives short-range eHorizon data, as well. Finally, the CG forwards the prediction to the HCU where the eHorizon control strategies will receive it as if the vehicle were on the real road.

1) Long horizon – Testing navigation data

Following the light green boxes in Fig. 7, the driver sets the desired destination (point B in Fig. 8) into the CG that has an integrated Human-Machine Interface (HMI). Then, the CG combines it with the actual vehicle position, simulated by SUMO, shown as point A in Fig. 8, and it sends them via the internal vehicle network to the TeCU along with the request of

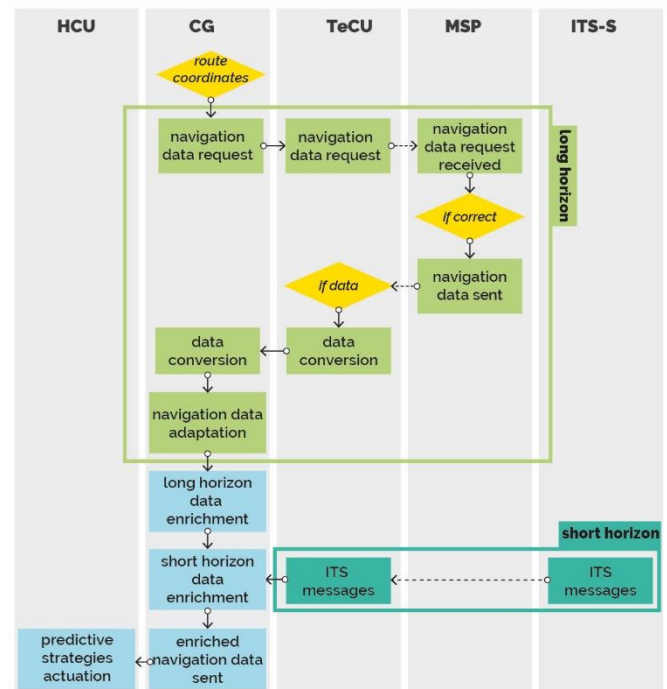


Fig. 7 Short and long eHorizon data flowchart

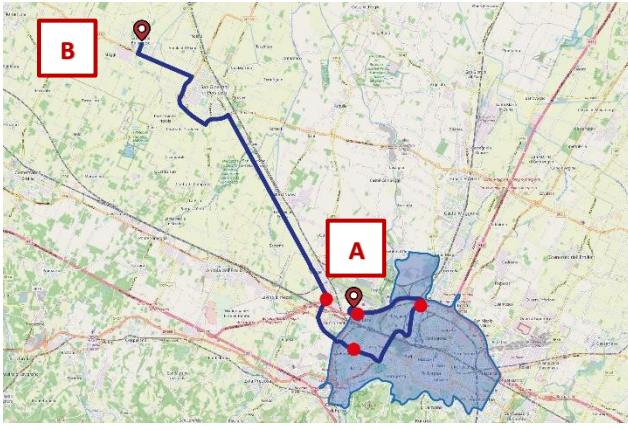


Fig. 8 Route representation with stop events (red dots)

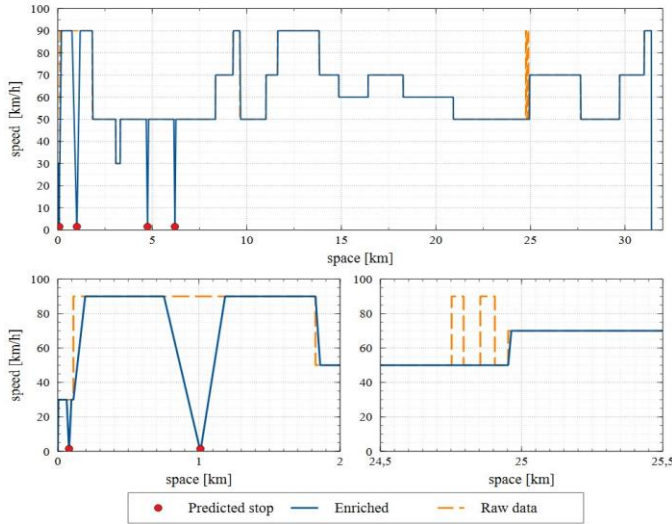


Fig. 10 Speed profile received and enriched

the navigation data prediction. The TeCU elaborates the request and publishes it on the remote server (Map Service Provider) using the MQTT protocol. The MSP checks if the received coordinates represent a plausible vehicle's position, e.g., if the vehicle position is within a building or outside the road. Again via MQTT, the MSP sends the prediction of the navigation data back to the TeCU (the orange line in Fig. 9, named "raw data"). As already stated, the MSP provides:

- Static speed limits;
- Slope profile;
- ZEZ topological information (identified with the blue area in Fig. 8).

Finally, the prediction is forwarded to the CG that performs the enrichment, using the information about stop events, decelerations, and accelerations. In particular, the CG:

- searches for and erases unplausible discontinuities in the speed limits, (the orange spikes depicted in the bottom right chart in Fig. 9) from the prediction. These discontinuities are due to driving through roads with different speed limits, but where it would be either impossible or illogical to accelerate the vehicle to reach the limit and then immediately decelerate;

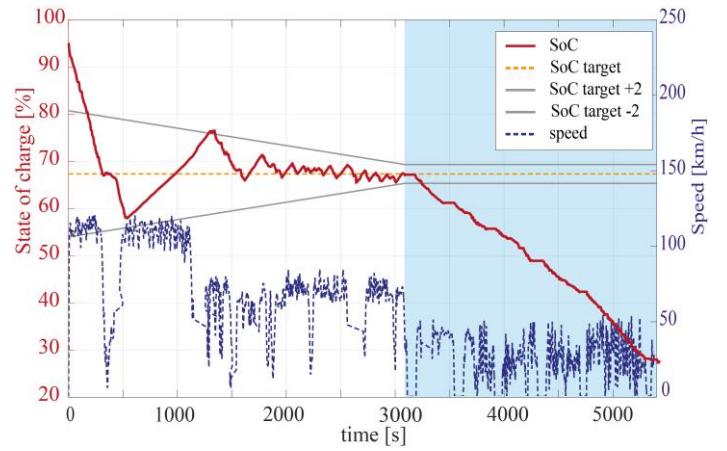


Fig. 9 Application of long-horizon predictive function

- enriches the speed profile with the information about the static stop events (red points in Fig. 8 and Fig. 9) foreseen by SUMO along the route. The stop events include traffic lights, rights of way, and roundabouts;
- modifies the speed profile with realistic accelerations and decelerations, as shown in the bottom left chart of Fig. 9.

Then, following the flowchart of Fig. 7, the navigation data prediction should now be enriched with short horizon data.

In Fig. 10, an HCU's Predictive Energy Management (PEM) function that uses the long-horizon data is presented. Currently, the chosen commercial MSP is not able to provide a sufficiently detailed prediction of the vehicle's speed profile, so the TeCU overwrites it and sends a speed profile measured on the road. The measured speed profile used as prediction comprehends an initial highway road segment, followed by an extra-urban and urban one (dashed blue line in Fig. 10). Then, the PEM strategy calculates the amount of energy needed to complete the urban segment in pure electric, to simulate a Zero-Emissions Zone (blue area). In this way, the PEM strategy sets an SoC target (orange dashed line) to be reached at the beginning of the urban event. At this aim, the PEM controller makes the SoC upper and lower bounds stricter as the urban event is approaching, forcing the actual SoC to lie within the limits reported in grey in Fig. 10. Consequently, the vehicle has sufficient stored energy to perform the urban event without switching on the ICE.

2) Short horizon – Simulation of V2I and V2V

In this paragraph, the V2I and V2V communication is simulated to test the short horizon data exchange, following the dark green boxes in the flowchart shown in Fig. 7.

In Fig. 11, a focus on the components of Fig. 3 involved in V2I and V2V communication is presented, with an insight of the radio link simulator. The real communication system (the light blue boxes) includes the TeCU, the OBU, and the external ITS-S. Those included in the light grey box are the same as those on the vehicle. In this configuration, the TeCU receives the short horizon messages, defined in Table III, that are on the top of the facility layer defined by CAM, DENM, and SPaT. Then, the TeCU forwards that data to the CG.

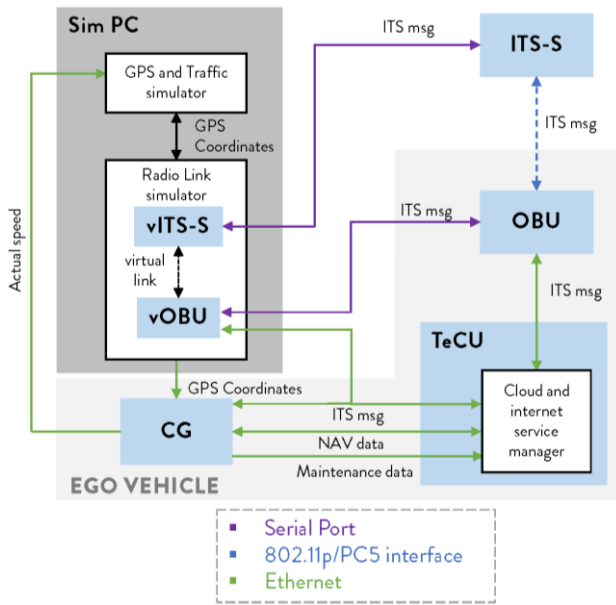


Fig. 11 V2I and V2V communication simulator

The real communication system involves the ego vehicle OBU that exchanges messages with the ITS-S using over-the-air communication. The ITS-S can work as an OBU or as an RSU depending on the situation encountered during the simulation. In this case, the ITS-S simulates an OBU mounted on another moving vehicle. The communication takes place between the real ITS-Ss, the TeCU, and the CG but in order to overcome the fact that both the OBU and the ITS-S are close to each other in the laboratory, and the communication would always be successful, a virtual link simulator is implemented.

The ITS-Ss need the vehicles' GPS positions as input. Thus, the GPS and Traffic Simulator sends the simulated WGS84 coordinates to the Radio Link Simulator (RLS), to calculate the distances between each station. While the communication delay is actually the real one already introduced by the devices, the RLS works on the introduction of realistic errors. The distances are used to calculate the receiving power of the ITS-Ss involved in the simulation, indicated as virtual OBU (vOBU) and virtual ITS-S (vITS-S). Then, the RLS simulates a virtual link for each of the vITS-S. This means that, instead of controlling the transmitted power (and so introducing attenuation) of the ITS-S, obtaining errors in signal demodulation, the radio link simulator automatically decides either if a packet can be sent by the ITS-S (and thus received from the OBU) or not, depending on the condition imposed by the simulation (distances, path loss, shadowing). In the model, path loss and shadowing are computed for each link with respect to the model described by 3GPP in [30], which has been implemented in our system by means of a Python program. If the receiving power exceeds the receiver sensitivity (set at -85 dBm, the carrier frequency at 5.9 GHz, and the bandwidth at 10 MHz as it would be in the real OBU), then the RLS triggers through serial port the real ITS-S, which sends the facility layers message (e.g., CAM). Concurrently, the RLS adds the messages received by the vOBU to a database. When the real OBU receives the message from the ITS-S via 802.11p, it triggers the TeCU to read all the messages stored into the database and forwards them to the CG.

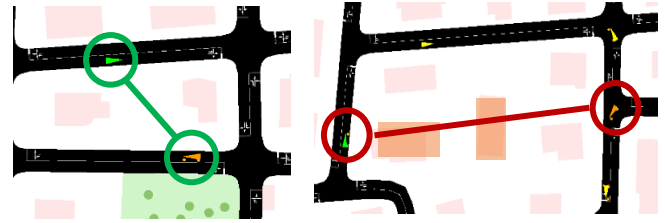


Fig. 12 LOS (left), NLOS (right)

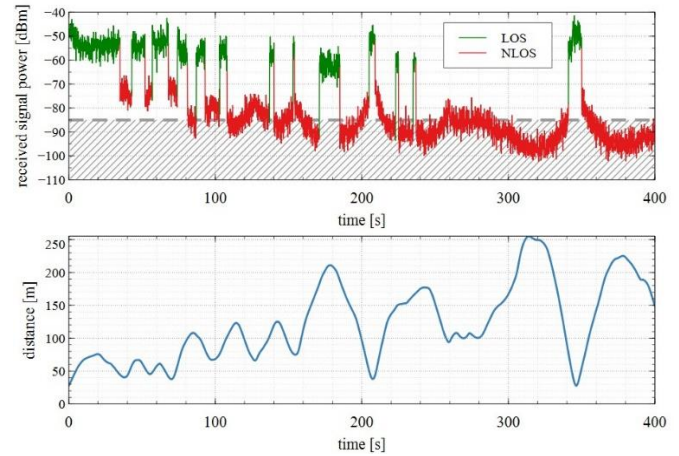


Fig. 13 Message transmission simulation between two moving vehicles

To test the V2I and V2V communication, two vehicles have been simulated in SUMO while driving and exchanging messages in an urban area of the city of Bologna. Two screenshots of the simulation are presented in Fig. 12. There, the ego vehicle is represented by the orange triangle, the other connected vehicle by the green triangle while the yellow ones represent the traffic vehicles not involved in the communication. The pink/orange shapes represent the buildings, imported from OpenStreetMap.

In Fig. 13, the behavior of the system during the simulation is presented. The upper plot shows the variation of the received signal power at vOBU. Moreover, to help the visualization, the line becomes green when there is Line of Sight (LOS) and red when there is No Line of Sight (NLOS), highlighting how power is affected. Then, the lower plot shows the distance between the two vehicles. More in detail, the vOBU (orange arrow in Fig. 12) receives CAMs every 100ms from the vITS-S mounted on the simulated vehicle (green arrow in Fig. 12) which follows the same route as the virtual ego vehicle at the beginning of the simulation. Then, the two vehicles start to follow different routes and encounter each other at some intersections. These situations are shown in Fig. 13, where the more the vehicles are distant and there is NLOS condition, the more the received power decreases. On the other hand, if the vehicles are distant from each other but there is LOS, the receiving power is less conditioned.

Therefore, the C-HiL is able to simulate a traffic flow in SUMO that could comprehend hundreds of vehicles in order to simulate a realistic ego vehicle speed trajectory. Anyway, only a few of them will be equipped with OBUs, to focus on a realistic case for the near future, and up to around ten connected vehicles will be simultaneously simulated. In fact, since a

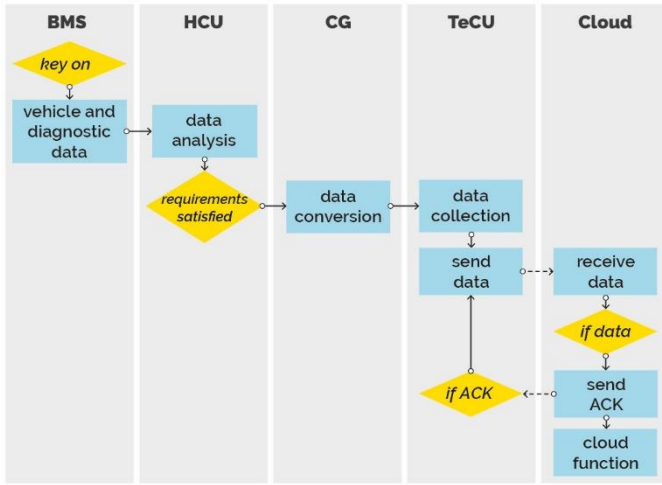


Fig. 14 In-cloud functions communication flowchart

message (in this case CAM) of 350 bytes in ITS-G5 can be transmitted in 500 μ s and the maximum period of CAM transmission is 100ms, if the use case involves the simulation of 10 OBUs transmitting without any overlap, each message is occupying about 5ms. Moreover, since the OBUs in ITS-G5 use CSMA/CA, it results highly improbable that two of them transmit in the same instant even if both are willing to send at the same time. Of course, this holds for all the OBUs which are sufficiently close to each other and so the received power is above the threshold. Given the statement above, this kind of delay can be considered negligible for the specified use cases, so only the delay of the real communication system (one OBU for remote vehicles and one OBU for the ego vehicle) is considered. In fact, the use cases are not related to safety functions and so they are not sensitive to latencies in the order of milliseconds. Even if only the delay of the real system is considered, the simulated OBUs send messages in an asynchronous way as in real systems.

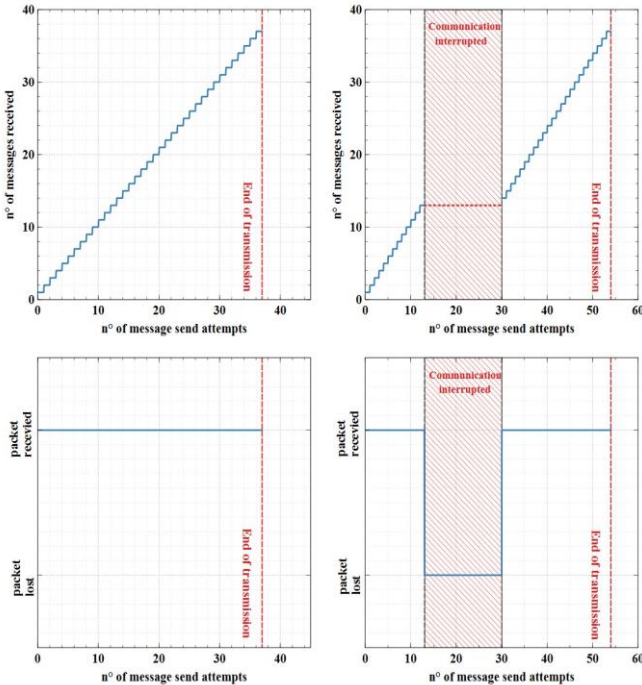


Fig. 15 Simulation of communication interruption

In the end, because the short-range predictive energy management functions are still in a development phase, and the enrichment algorithm depends strictly on the data required by the functions themselves, only the communication from OBU/RSU to Central Gateway is described. As for now, the short horizon prediction is not vital for the eHorizon functions that will initially be tested.

B. Predictive Maintenance – Testing in-cloud function

The communication with private servers enables the OEMs to implement new functionalities. With respect to our use case, it is used to receive diagnostic data from the ego vehicle and to process the data in order to predict faults. Another benefit will be the possibility to implement control functions with high computational load, otherwise not suitable for the on-board application.

To implement the V2N, an application for High-Performance Computers (HPC) has been developed as a cloud instance and implemented for being tested in the proposed platform. More in detail, the TeCU uses a custom communication protocol based on TCP/IP for communicating with the cloud instance, and such protocol has been implemented with Python language.

To test the communication with a realistic case study, a predictive battery maintenance function is running in the private server. In Fig. 14, the simplified example of the communication flowchart between the vehicle and the cloud is presented. Such communication is performed at every key on event, and in this example, the diagnostic data come from the BMS that measures physical parameters of the battery (such as current, voltage, and temperature) and sends them to the HCU. The latter uses these data to perform an estimation of the remaining capacity of the battery. Then, they are converted into strings by the CG and finally sent to the TeCU where these data are collected and transmitted. The communication protocol developed is based on acknowledgments (ACKs). The first packet is sent to the cloud function. From the second onwards, the packet is sent if and only if the cloud function sends back

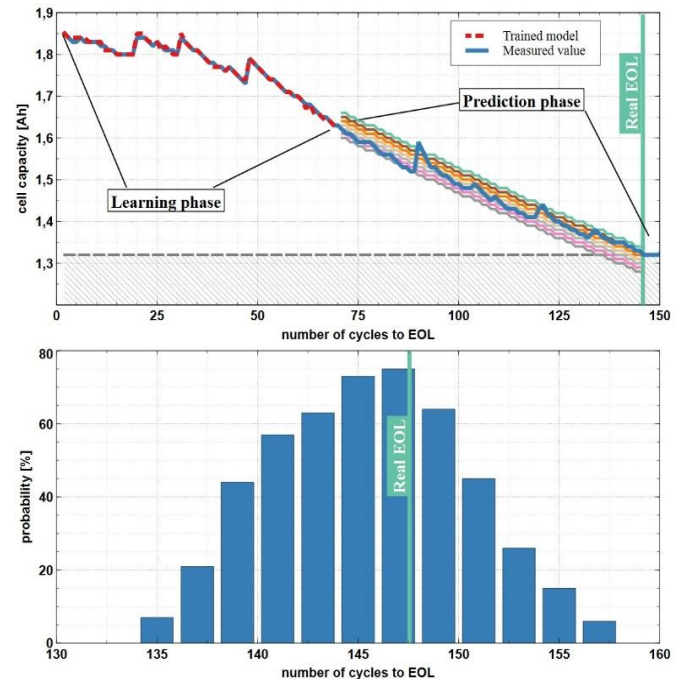


Fig. 16 Predictive maintenance use case

an ACK containing the sequence number of the previous message, as shown in Fig. 14. In this way, when the cellular connection is down (e.g., the vehicle parked in an underground), the TeCU can store the packets coming from the CG into its internal memory, and then it will try to send them when the connection is restored. To test it, a communication interruption has been forced externally during the data transmission and, as this happens, the TeCU immediately starts storing the data while it continues to try sending them. When the communication is restored, the data transmission ends without losses, as shown in Fig. 15. There, if the communication is up, every time there is a data send attempts there is the packet receiving. When the interruption occurs, the attempts rise in numbers while the number of packets received remains constant.

As already stated, a predictive battery maintenance function has been used to validate the communication system. At this aim, aging data of a Li-Ion cell have been retrieved from the NASA Ames Prognostics Data Repository [28]. More in detail, supposing the connection is not interrupted, every time a driving cycle is over and the BMS sends the battery data, the in-cloud function stores the data in the server. Then, it starts to train a predictive battery SoH model. In the top chart of Fig. 16, the blue line represents the measured cell capacity data that have been sent to the cloud at the end of every driving cycle.

Once the learning phase is completed, the function provides several predictions (colored lines), each of them characterized by a specific probability as a function of the number of cycles to end of life (EOL), as shown in the bottom chart of Fig. 16. Finally, the prediction with the highest probability is chosen (light blue line). In this case, the advantage of the cloud computing results clear when the predictive SoH function will be applied to an entire vehicles' fleet, each of them sending maintenance data to the OEM server. In fact, the more the data available and stored, the more accurate is the prediction.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, a hybrid vehicle Hardware-in-the-Loop system with integrated connectivity has been presented as an answer to the market request for cleaner, more efficient, and safer vehicles while reducing their time-to-market.

It results in a more modular and universal tool for testing and validation of predictive electronic horizon functions. On one hand, it provides higher flexibility due to the possibility to test different types of predictive functions (long and short horizon, and predictive maintenance) and the capability of acting on each component and controller model. On the other hand, it grants higher reliability, as both the hardware and the software are the same as those implemented on the vehicle, allowing seamless functions implementation on-board once validated at the HiL. Thus, it will shorten the validation process and further reduce the gap between laboratory and on-vehicle tests.

Thanks to the C-HiL, eHorizon functions will be tested and validated with the same information they would have on the real vehicle. The following step will be to move them into the private servers relieving the HCU from the excessive computational load and enhancing the prediction range. Moreover, the radio link simulator will be enhanced for the long-range communication technologies by setting up a V2N

simulator in order to simulate and control cellular network issues such as traffic load, handover, and typical over-the-air interface disturbances. Moreover, for those cases in which an increasing number of simulated OBUs is needed, a model for collisions will be implemented, since in a dense traffic scenario it is probable to have collisions, and therefore loss of packets.

Finally, future work comprehends the 5G implementation, because thanks to its low latency it will be particularly suitable for V2x communication.

REFERENCES

- [1] P. Martino, M. Melchiorri, A. Siragusa, and T. Kemper, "Atlas of the Human Planet 2016. Mapping Human Presence on Earth with the Global Human Settlement Layer," *EUR 28116 EN*, 2016, DOI: [10.2788/889483](https://doi.org/10.2788/889483).
- [2] H. Ritchie and M. Roser, "Urbanization," *Our World in Data*, Jun. 2018. <https://ourworldindata.org/urbanization> (accessed Apr. 27, 2020).
- [3] H. Ritchie and M. Roser, "CO₂ and Greenhouse Gas Emissions," *Our World in Data*, May 2017. <https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions> (accessed Apr. 27, 2020).
- [4] "REGULATION (EU) 2019/631 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL," *Official Journal of the European Union*, Apr. 2019.
- [5] J. Mueller and Y. Le Petit, "Low-Emission Zones are a success - but they must now move to zero-emission mobility," *European Federation for Transport and Environment AISBL*, 09/19. https://www.transportenvironment.org/sites/te/files/publications/2019_09_Briefing_LEZ-ZEZ_final.pdf.
- [6] J. T. B. A. Kessels and P. P. J. van den Bosch, "Electronic Horizon: Energy Management using Telematics Information," in *2007 IEEE Vehicle Power and Propulsion Conference*, Arlington, TX, USA, Sep. 2007, pp. 581–586, DOI: [10.1109/VPPC.2007.4544190](https://doi.org/10.1109/VPPC.2007.4544190).
- [7] F. Zhang, J. Xi, and R. Langari, "Real-Time Energy Management Strategy Based on Velocity Forecasts Using V2V and V2I Communications," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 416–430, Feb. 2017, DOI: [10.1109/TITS.2016.2580318](https://doi.org/10.1109/TITS.2016.2580318).
- [8] M. A. Mohd Zulkefli, J. Zheng, Z. Sun, and H. X. Liu, "Hybrid powertrain optimization with trajectory prediction based on inter-vehicle-communication and vehicle-infrastructure-integration," *Transportation Research Part C: Emerging Technologies*, vol. 45, pp. 41–63, Aug. 2014, DOI: [10.1016/j.trc.2014.04.011](https://doi.org/10.1016/j.trc.2014.04.011).
- [9] J. Oncken and B. Chen, "Real-Time Model Predictive Powertrain Control for a Connected Plug-In Hybrid Electric Vehicle," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8420–8432, Aug. 2020, DOI: [10.1109/TVT.2020.3000471](https://doi.org/10.1109/TVT.2020.3000471).
- [10] F. Zhang, X. Hu, R. Langari, and D. Cao, "Energy management strategies of connected HEVs and PHEVs: Recent progress and outlook," *Progress in Energy and Combustion Science*, vol. 73, pp. 235–256, Jul. 2019, DOI: [10.1016/j.pecs.2019.04.002](https://doi.org/10.1016/j.pecs.2019.04.002).
- [11] C. Morley, "Automotive Industry Life Cycle - Shorter Timelines | Jabil," *Jabil.com*, Dec. 2018. <https://www.jabil.com/blog/automotive-industry-trends-point-to-shorter-product-development-cycles.html> (accessed Jul. 30, 2020).
- [12] W. Wachenfeld and H. Winner, "The Release of Autonomous Vehicles," in *Autonomous Driving*, 1st ed., M. Maurer, C. Gerdes, B. Lenz, and H. Winner, Eds. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2016, DOI: [10.1007/978-3-662-48847-8_21](https://doi.org/10.1007/978-3-662-48847-8_21).
- [13] M. Tideman and M. van Noort, "A simulation tool suite for developing connected vehicle systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, QLD, 2013, pp. 713–718, DOI: [10.1109/IVS.2013.6629551](https://doi.org/10.1109/IVS.2013.6629551).
- [14] M. Aramrattana, T. Larsson, J. Jansson, and A. Năbo, "A simulation framework for cooperative intelligent transport systems testing and evaluation," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 61, pp. 268–280, 2019, DOI: [10.1016/j.trf.2017.08.004](https://doi.org/10.1016/j.trf.2017.08.004).
- [15] F. Xu and T. Shen, "Look-Ahead Prediction-Based Real-Time Optimal Energy Management for Connected HEVs," in *IEEE Trans.*

- Veh. Technol.*, vol. 69, no. 3, pp. 2537–2551, March 2020, DOI: [10.1109/TVT.2020.2965163](https://doi.org/10.1109/TVT.2020.2965163).
- [16] T. Grahle and M. Tonne, “Regeneration of the Particulate Filter by Using Navigation Data,” *MTZ Worldwide*, vol. 77, no. 1, pp. 16–21, Jan. 2016, DOI: [10.1007/s38313-015-0078-7](https://doi.org/10.1007/s38313-015-0078-7).
 - [17] M. Hopka, D. Upadhyay, and M. V. Nieuwstadt, “Smart DPF Regenerations - A Case Study of a Connected Powertrain Function,” *SAE Int. J. Adv. & Curr. Prac. in Mobility* 1 (2), pp. 762–770, 2019, DOI: [10.4271/2019-01-0316](https://doi.org/10.4271/2019-01-0316).
 - [18] M. Menarini, P. Marrancone, G. Cecchini, A. Bazzi, Barbara M. Masini, and A. Zanella, “TRUDI: Testing Environment for Vehicular Applications Running with Devices in the Loop,” in *2019 International Conference on Connected Vehicles and Expo*, Graz, Austria, Nov. 2019, pp. 1–6, DOI: [10.1109/ICCVE45908.2019.8965152](https://doi.org/10.1109/ICCVE45908.2019.8965152).
 - [19] Z. Szendrei, N. Varga, and L. Bokor, “A SUMO-Based Hardware-in-the-Loop V2X Simulation Framework for Testing and Rapid Prototyping of Cooperative Vehicular Applications,” in *Vehicle and Automotive Engineering 2*, K. Jármai and B. Bollò, Eds. VAE 2018: Springer, Cham, 2018, pp. 426–440, DOI: [10.1007/978-3-319-75677-6_37](https://doi.org/10.1007/978-3-319-75677-6_37).
 - [20] S. Y. Gelbal, S. Tamilarasan, M. R. Cantas, L. Guvenc, and B. Aksun-Guvenc, “A connected and autonomous vehicle hardware-in-the-loop simulator for developing automated driving algorithms,” in *2017 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Banff, Canada, 2017, pp. 3397–3402, DOI: [10.1109/SMC.2017.8123155](https://doi.org/10.1109/SMC.2017.8123155).
 - [21] Y. Shao, M. A. M. Zulkefli, Z. Sun, and P. Huang, “Evaluating connected and autonomous vehicles using a hardware-in-the-loop testbed and a living lab,” *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 121–135, May 2019, DOI: [10.1016/j.trc.2019.03.010](https://doi.org/10.1016/j.trc.2019.03.010).
 - [22] Y. Kim, S. Tay, J. Guanetti, F. Borrelli, and R. Miller, “Hardware-In-the-Loop for Connected Automated Vehicles Testing in Real Traffic,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, 2018, pp. 2878–2883, DOI: [10.1109/ITSC.2018.8569753](https://doi.org/10.1109/ITSC.2018.8569753).
 - [23] R. Parenti, S. Mazzetti, F. Belletti, F.-W. Speckens, and M. Caggiano, “Will the 95g/km Limit Affect Fun and Emotion? The Lamborghini Answer,” presented at the 24th Aachener Kolloquium Automobile and Engine Technology, Aachen, Germany, 2015.
 - [24] F. R. Salmasi, “Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 2393–2404, Sep. 2007, DOI: [10.1109/TVT.2007.899933](https://doi.org/10.1109/TVT.2007.899933).
 - [25] “Alma Automotive - Custom solutions for engine development and testing,” <https://www.alma-automotive.it/> (accessed Apr. 30, 2020).
 - [26] J. Santa, F. Pereñíguez, A. Moragón, and A. F. Skarmeta, “Experimental evaluation of CAM and DENM messaging services in vehicular communications,” *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 98–120, Sep. 2014, DOI: [10.1016/j.trc.2014.05.006](https://doi.org/10.1016/j.trc.2014.05.006).
 - [27] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO – Simulation of Urban MObility,” *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. SIMUL 2011, 23.-28. Oct. 2011, Barcelona. ISBN 978-1-61208-169-4.
 - [28] 3GPP, “Initial Cellular V2X standard completed,” *Initial Cellular V2X standard completed*. https://www.3gpp.org/news-events/3gpp-news/1798-v2x_r14 (accessed May 25, 2020)
 - [29] “PC Engines apu3c4,” *PC Engines*. <https://www.pcengines.ch/apu3c4.htm> (accessed May 25, 2020).
 - [30] 3GPP, “Study on evaluation methodology of new Vehicle-to-Everything (V2X) use cases for LTE and NR,” TR 37.885 V15.3.0, Jun. 2019.
 - [31] B. Saha and K. Goebel, “Battery Data Set,” *NASA Ames Prognostics Data Repository*, 2007. <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/> (accessed Jun. 25, 2020), NASA Ames Research Center, Moffett Field, CA.