IET Intelligent Transport Systems

Special Section: Scientific Seminar of the Italian Association of Transport Academicians (SIDT) 2019

IET Journals
The Institution of Engineering and Technology

Integrating tools for an effective testing of connected and automated vehicles technologies

ISSN 1751-956X
Received on 16th October 2019
Revised 1st June 2020
Accepted on 24th June 2020
E-First on 28th July 2020
doi: 10.1049/iet-its.2019.0678
www.ietdl.org

Luigi Pariota[1,2] ✉, Angelo Coppola[3], Luca Di Costanzo[1], Antonio Di Vico[1], Arcangelo Andolfi[1], Claudio D'Aniello[2], Gennaro Nicola Bifulco[2]

[1]*LAERTE, Laboratory for Advanced Experiments on Roads and Traffic Environments – Centre for Advanced Metrology Services, University of Naples 'Federico II', Napoli, Italy*
[2]*Department of Civil, Architectural and Environmental Engineering, University of Naples 'Federico II', Napoli, Italy*
[3]*Department of Electrical Engineering and Information Technology, University of Naples 'Federico II', Napoli, Italy*
✉ *E-mail: luigi.pariota@unina.it*

Abstract: The development of connected and automated driving functions involves that the interaction of autonomous/automated vehicles with the surrounding environment will increase. Accordingly, there is a necessity for an improvement in the usage of traditional tools of the automotive development process. This is a critical problem since the classic development process used in the automotive field uses a very simplified driver model and the traffic environment, while nowadays it should contemplate a realistic representation of these elements. To overcome this issue, the authors proposed an integrated simulation environment, based on the co-simulation of Matlab/Simulink environment with simulation of urban mobility, which allows for a realistic model of vehicle dynamic, control logics, driver behaviour and traffic conditions. Simulation tests have been performed to prove the reasoning for such a tool, and to show the capabilities of the instrument. By using the proposed platform, vehicles may be modelled with a higher level of details (with respect to microscopic simulators), while the autonomous/automated driving functions can be tested in realistic traffic scenarios where the features of the road traffic environment can be varied to verify in a realistic way the level of robustness of the on-board implemented functions.

1 Introduction

The increasing demand for mobility poses great challenges for road transport systems in the field of safety, energy-saving and mobility efficiency. Road congestion increases travel time, fuel consumption and reduces safety, triggering aggressive and unsafe driver behaviour [1]. New technologies are nowadays seen as a solution to increase vehicles' performance and capabilities and are going to interact with road driving environment helping or replacing the driver to manage unexpected and/or unsafe traffic situations [2]. This idea, carried out starting from Prometheus initiative in 1986 [3], led to the development of Advanced Driving Assistance Systems (ADAS), which support the driver in different ways: avoiding or reducing the probability of collisions; alerting the driver in case of emergency; assisting the driver while driving. The main goal of these systems is to reduce road accidents in Europe by 2020, from 31,000 in 2010 to 15,000.

However, automated driving (AD) SAE-3 level or higher, could not exploit all their benefits if cooperative and communication components are not implemented [4]. Intelligent Transportation Systems (ITS) exploit V2X (Vehicle-to-Everything) communication for several applications that aim to enhance road safety, traffic efficiency and infotainment. More specifically, one of the fundamental aims in ITS is to cooperatively drive along the road, as highlighted in the final reports of C-ITS (Cooperative Intelligent Transportation Systems) platform by EU. Connected autonomous vehicles, leveraging Vehicle-to-Vehicle (V2V) [5] and/or Vehicle-to-Infrastructure [6] communication, can share information with surrounding road users and/or receive a reference signal coming from a leading vehicle or a road infrastructure. They significantly improve road safety, road efficiency and reduce at the same time the fuel consumption [7, 8].

The increasing complexity of these systems requires continuous reliability and verification tests, to guarantee safety, quantify possible advantages and evaluate arising issues [9]. However, an on-road test of the vehicle behaviour requires a considerable effort (e.g. in terms of time and money) and does not allow to repeat a specific test in the same identical condition (no deterministic test conditions). To overcome all these issues, automotive manufacturers make extensive use of simulation solutions (also called 'Virtual Validation' solutions), which nowadays have a key role in the development process of autonomous systems because of lower time and higher simplicity in carrying out test [10].

In this work, we developed an integrated simulation environment that responds to this need. It embeds traffic micro-simulators and vehicle dynamics to create a holistic virtual simulation environment and exploits the strengths of each component. The proposed simulation environment, combining Matlab/Simulink and SUMO (Simulation of Urban Mobility) [11] environments, is a modular platform with different (and replaceable/modifiable) levels of detail for each component. More specifically, the holistic vision of the ntegrated simulation platform allows to obtain an explicit and realistic representation of vehicle dynamics, driver behaviour, road traffic environment (e.g. infrastructure and traffic), autonomous/AD systems and communication systems for cooperative applications. The use of the Matlab/Simulink environment allows to emulate each component both by in-house and specific third-part hardware/software solutions, so they are replaced and/or modified as needed. Each of them is connected to the rest of the system and, hence, its testing and validation is performed by simulating all the components simultaneously in a unified common environment.

2 Related work

Traffic micro-simulators, such as SUMO, Vissim [12], and MATSim [13], are conventionally used to estimate the level of service of road infrastructure, design and calibrate traffic control logics [14], and compare the performances of different road design solutions. Moreover, the increasing attention to environmental problems, such as air pollution, fuel consumption and noise production, has suggested to include the above aspects as decision criteria for road or traffic control designs and, more generally, for

urban mobility [15]. Some studies, such as in [16], focused on the applicability of traffic micro-simulators to estimate vehicle emission based on real-world data from an expressway, but, due to systematic error, they not provide a realistic driving behaviour to estimate both consumptions and pollutant emissions of the vehicle. By combining the use of these tools and instantaneous emission and consumption models [17, 18], exploiting micro simulation data as inputs, is possible to overcome these issues.

From the early 2000s, traffic micro-simulators have also been used to study the impact and the effect of ADAS on traffic quality and efficiency [19]. A wide literature focused on the analyses of different ADAS such as Adaptive Cruise Control (ACC) [20], Intelligent Speed Adaptation [21] and overtaking assistants [22]. Moreover, some authors investigate on variables that influenced the effect of such systems on traffic, by varying penetration rate [23, 24], vehicle fleet [24], road infrastructure and driver behaviour. Consequently, the availability of high-detailed digital maps is rapidly becoming an absolute requirement, to categorise the points of the drivable region and determine the location of the road signs also in complex urban environments, e.g. when the road signs are located on the ground [25].

Similarly, following the development of communication technologies, traffic micro-simulators started to be used as simulation testing environment for technologies ranging from information systems [26], up to modern systems based on V2X (Vehicle-to-Everything) communication, and traffic control logics. Plexe [27], a SUMO extension that allows vehicular network simulation by integrating Veins, was widely used to simulate platooning applications, in presence [28] or not of communication delay [29, 30], while in [31] emission effects of Cooperative Adaptive Cruise Control (C-ACC) are estimated.

It is worth noting that several authors criticised this kind of approach, since it considered only the functioning of the system, and not the changes induced in driver behaviour [20]. Hence, conventional microscopic traffic simulation models need to be enhanced to manage both the driver assistance systems and the variation in driver behaviour that these systems may induce. Some micro-simulation tools allow on-board functionalities to be embedded via an application programming interface, however, the implemented systems are very simplified models of the on-board real ones. In some cases, such simplifications can be justified; In some cases, e.g. for systems that take partial control of the driving, using a simplified model may not produce a valid representation of the behaviour of the vehicle in the traffic simulation scenarios.

On the other hand, the classical automotive development process focuses on a highly detailed model of the vehicle and its components (e.g. body, powertrain and tyre), on-board sensors and control logics. The spread of Autonomous/Automated functions, as well as V2X and ADAS [32, 33], lead cars to become part of a complex interacting system, which need to be well represented in the used tools to accurately design and test the vehicle and its technologies.

Virtual Testing, allowing for short-time simulations, safe test conditions, test reproducibility and a high number of simulations runs, represents the most effective solution to verify and validate such complex systems prior to on-road tests [10]. In one particular setting, hardware-in-the-loop (HIL), simulations are performed to test physical components of the vehicle, especially electronic control units [34, 35].

Existing commercial solutions [36], e.g. IPG CarMaker, TASS PreScan, VIRES VirtualTestDrive and dSPACE, typically focus on the control of a high-detailed single vehicle, which can eventually be equipped with on-board sensors (e.g. camera, radar and lidar) and devices for V2X communication. For instance, dSPACE has embedded a model for automotive simulations into its testing environment with the aim to develop and test driver assistance systems that react to surrounding vehicles. Specifically, it allows to test the behaviour of a single controlled vehicle, equipped with on-board sensors, that interact with surrounding vehicles (for a maximum of 15 and following a pre-defined behaviour) [37]. This latter tool, like many others, presents simplified road description and surrounding vehicles' behaviour, which consists of pre-loaded and predefined driving trajectories, given as a simulation input.

Open-Source solutions [38–42], which allow to reduce costs, also present some issues. TORCS, a multi-vehicle simulation platform, do not contain the mathematical representation of vehicle behaviour, while CoInCar-Sim does not integrate vehicle dynamics and realistic traffic representation. CARLA, AirSim and Udacity present a detailed physic environment but focus on generating behaviours for learning-based approaches.

The lack of appropriate simulation tools involves both a deficiency of information in the development process of autonomous/AD solutions, and an inability to properly consider the real impact of the considered system on the surrounding traffic actors.

To overcome the above issues, current research focuses on the creation of an integrated simulation environments. Some commercial solutions have been integrated traffic simulator to simulate realistic traffic representation, e.g. IPG Car Maker can be integrated with SUMO [43] and Vissim [44]. Cottignies *et al*. [45] proposed a co-simulation between rFpro and SUMO allowing a complete simulation of urban environments for ADAS and Autonomous Testing. Currently, the last version of DYNA4, integrated with SUMO, allows also multi-vehicles control with detailed dynamics.

Differently, So *et al*. [46] proposed an integrated use of AIMSUM, i.e. the traffic micro simulator, CARSIM, i.e. vehicle dynamics simulator, and Passenger Car and Heavy Duty Emission Model to estimate pollutant emissions. The vehicle dynamics and emission model run in post-process by using outputs from AIMSUM simulation as inputs. This type of solution, even if it offers high fidelity vehicle and sensors' models, is very expensive and not light-weighted.

Less work exists on coupling automated vehicles with traffic simulations. Few in-house solutions exist on coupling vehicle dynamics and traffic simulators [47, 48], which allow to reduce cost and unnecessary dependencies; they are often based on Matlab/Simulink environment to emulate vehicle dynamic and sensors. Griggs *et al*. [49] have proposed an integrated framework for HIL solutions, combined with SUMO, while authors in [47] combined the traffic micro-simulator with a high-detailed powertrain modelled in Simulink. A different approach is proposed in [50, 51], in which driving simulators are combined with microscopic traffic simulator to provide a consistent solution to deal with the challenging problems described above.

This work falls within the latter path and its aim is to motivate and show the development of an integrated simulation environment, which embeds different interacting simulation tools within a holistic view of the co-simulation approach. More specifically, we integrate SUMO, as traffic micro-simulation environment, and Matlab/Simulink as integration environment in which model and emulate complementary parts of the simulation, such as vehicle dynamics, ADAS and V2X communication technologies.

3 Materials and methods

The platform had been developed unifying, in an interactive way, a set of tools, each of which performs specific tasks to use the strengths of each component.

The classic approaches in the automotive field are characterised by a highly detailed characterisation of the vehicle model and the on-board components. Instead, the traffic environment and the driver model used in the simulation are a really weak aspect. Traffic environment is characterised by a limited or no presence of pedestrians and other vehicles and a limited characterisation of the network, while driver model, typically, presents an extremely simplified behaviour, generally pre-set in each run. In the proposed approach, also critical components are modelled in detail, using field-specific tools, integrated with traditional ones. Matlab/Simulink environment is used for vehicle model (i.e. vehicle dynamics, tyre and powertrain models), driver model and control logics, while SUMO environment provides the traffic environment. The Traffic Control Interface (TraCI) 4Matlab is used to establish communication and to exchange information between SUMO and Matlab/Simulink.

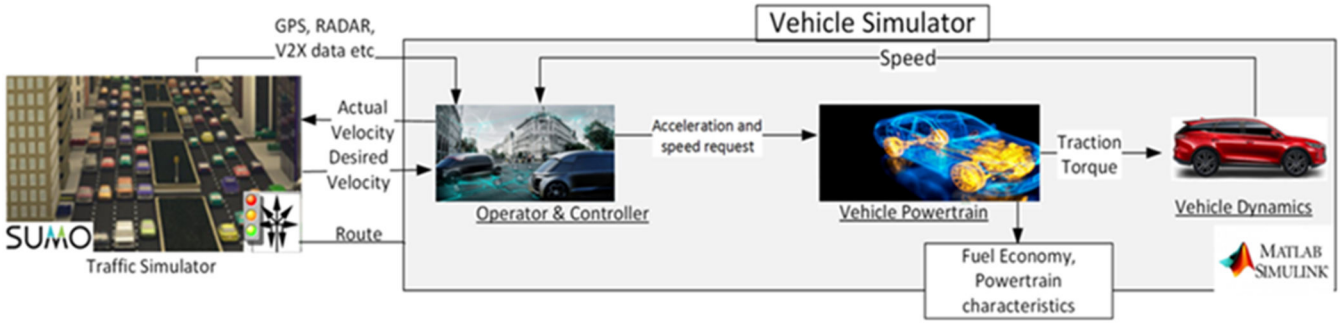

Fig. 1 *Integrated simulation environment architecture*

Table 1 Configuration of vehicle models

	Bicycle	Quadricycle
ICE	✓	✓
EM	✓	✓

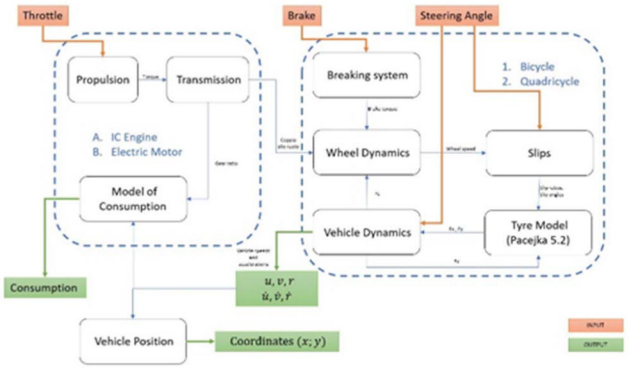

Fig. 2 *Vehicle model architecture*

The operating scheme of the proposed integrated simulation platform is shown in Fig. 1.

Note that the proposed simulation environment allows to control one vehicle at a time, called Ego-Vehicle.

3.1 Road traffic environment

The road traffic environment is emulated by SUMO, an open-source, microscopic, multi-modal and time-discrete traffic simulator. Furthermore, it contains a set of tools that help to prepare and perform the simulation of a traffic scenario. The traffic simulation performed by SUMO is time-discrete, with a default step length of 1.0 s, which can be chosen down to 1.0 ms, and the maximum duration of a scenario is 49 days. A simulation, to be performed, needs for a road network and a traffic demand. Each of them is defined into a specific XML file, which is the SUMO file format.

Road Network can be generated both by using the application *netgenerate* and by importing a digital map using *netconvert*, which allows to read networks from other traffic simulators (e.g. VISUM, Vissim or MATSim) or other digital road networks formats (e.g. OpenStreetMap [52], OpenDRIVE [53] and RoboCup). Another tool, named *netedit*, allows editing road network using a GUI (Graphical User Interface). The SUMO-network file contains information about nodes, edges and traffic light plans. Nodes describe the intersections, which consist of a position, shape and right-of-way rules. Edges describe the road segments, which have a fixed number of lanes characterised by a defined geometry, maximum allowed speed and information about vehicle classes allowed on them.

The traffic demand can be defined both manually and by using *origin/destination matrices* (O/D matrices), which describe the movement between different traffic analysis zones. However, to use these matrices in SUMO, they must be disaggregated into individual trips. To reach this goal is possible to use *od2trips* tool, which converts O/D matrices into single-vehicle trips. Each vehicle is modelled explicitly, defined by a unique ID, route, departure time and speed. It can be assigned to a defined Vehicle Type, which describes the physical properties of the vehicle and the variables of the movement model, and to a pollutant or noise emission classes. Vehicle motion is controlled through the car-following model (compute the desired speed of Ego-Vehicle on the base of its own state and the surrounding environment), lane change model and right-of-way at the intersection.

3.2 Vehicle model

The motion of the Ego-Vehicle is described through dynamics equations which refer to a 3-Degree of Freedom (DOF), rear-drive and front steering vehicle model. The Ego-Vehicle model is built in a modular framework allowing to choose between two types of propulsion and relative model of consumption, Internal Combustion Engine (ICE) or Electric Motor (EM), and different configurations for wheels (*bicycle* or *quadricycle* model). In conclusion, as schematically represented in Table 1, four different versions of the vehicle model are available.

The torque engine/motor ($T_{\text{motor/engine}}$), provided to the driveshaft, is evaluated as the ratio between its power and rpm, multiplied for the throttle percentage and then it flows to the drive wheels through the gearbox and differential. Each wheel is outlined as a 1-DOF dynamic model whose rotational acceleration is the result of dynamic balance among drive, brake and resisting torque. Wheel rotational speeds are calculated by integrating accelerations; vehicle speeds and steering angle are used to evaluate wheel slips parameters, while accelerations impact on vertical tyre loads. Slips, vertical loads and camber angles represent the input data of the tyre model which estimates the interaction forces along longitudinal and lateral directions. More, vehicle dynamics equations evaluate vehicle accelerations through dynamic balance among tyre forces, aero drag force and rolling resistance. Accelerations are integrated to get vehicle speeds and lastly, vehicle speeds are integrated to obtain step-by-step position of the vehicle. The general vehicle model architecture is depicted in Fig. 2.

In the following subsection, the models implemented will be described in detail.

3.2.1 Propulsion model: The engine/motor power is expressed as a function of throttle and rpm. The first attribute is the percentage of the power available at a specific rpm value rpm is the rotational speed of the motor calculated as the average speed of drive wheels reduced to the motor shaft through gear and differential ratio.

The power of the ICE is evaluated as follows:

$$P = P_{\max}\left[-\left(\frac{n}{n_{P_{\max}}}\right)^3 + \left(\frac{n}{n_{P_{\max}}}\right)^2 + \left(\frac{n}{n_{P_{\max}}}\right)\right] \quad (1)$$

In the previous equation, the power is expressed by a cubic equation that is a function of the ratio between effective rpm (n) and rpm at maximum power ($n_{P_{\max}}$), multiplied for the max power ($P_{\max}$). The torque is simply obtained dividing the power for the rpm.

Instead, the EM model is realised using Simulink 'Mapped Motor' and 'Datasheet Battery'. Mapped Motor estimates the motor torque and the current output which goes into the battery model to calculate the battery voltage and his state of charge.

3.2.2 Wheel dynamics model: The rotational acceleration of the wheels is the result of a dynamic balance of a 1-DOF system. During acceleration or braking, manoeuvre the tyre, due to the speed difference between vehicle and wheel, deforms itself generating a traction force. The product of this force and the distance between hub and road generates an opposite torque to the drive one. Performing a dynamic balance of the generic wheel is possible to obtain the equation of its acceleration:

$$\dot{\omega}_{\text{wheel}} = \frac{(T_{\text{motor/engine}}/(\tau_{\text{gearbox}} \cdot \tau_{\text{differential}})) - T_{\text{brake}} - F_x \cdot R}{J_{\text{wheel}}} \quad (2)$$

Analysing the equation above, it is possible to note the net torque (numerator) is divided by the total moment of inertia reflected the wheel axle (denominator). More, the net torque is calculated as the difference among drive torque at the wheels (given by the ratio of motor/engine torque ($T_{\text{motor/engine}}$) and the ratios of gearbox and differential (respectively τ_{gearbox} and $\tau_{\text{differential}}$)), brake torque ($T_{\text{brake}}$) and the resistance torque due to the traction force generated by the tyres (F_x) multiplied for the wheel radius (R). The inertial moment of the wheels (J_{wheel}) is the sum of two contributions: wheel mass inertia and rotating mass inertia. About those contributions, the rotating mass is non-zero as regards drive wheels only, because of mechanical connection with the drive line, while the wheel inertia is constant and depends on the mass, radius and width of the wheel.

3.2.3 Gearbox model: The driveline ratio is given by the product of gear and differential ratios. Despite differential ratio, the first one is variable or not according to the type of motor. ICE works by a 6-speed gearbox: the gear is shifted up when rpm reach the value at maximum power and the gear is shifted down when power is too low. EM supplies good torque at low speed, is high revving and efficient across a broad rpm range. Its features make it unnecessary to have a transmission, unless for performance purposes. Since the vehicle model refers to a conventional passenger car, it is assumed to use a single-speed gearbox.

3.2.4 Tyre model: The vehicle is assumed to move like a rigid body which can only translate or rotate on the road surface. Its trajectory depends on how tyres compensate the resistant and inertial forces acting on the vehicle. Both during acceleration and steering manoeuvre, tyres deform to comply the velocity difference between vehicle and wheels. In this regard, two parameters have to be defined: slip ratio (k_{ij}) and slip angle (α_{ij}) indicate how much tyre deforms along its longitudinal and lateral direction [54]. Each wheel is indicated by subscripts i and j: i is used to refer to the wheel axle (1. Front axle, 2. Rear axle); j is used refer to the side of the axle (1. Left wheel; 2. Right wheel)

$$\kappa_{ij} = \frac{\omega_{ij} R_{ij} - V_{x_{ij}}}{V_{x_{ij}}} \quad (3)$$

$$\alpha_{ij} = -\,a\tan\left(\frac{V_{y_{ij}}}{V_{x_{ij}}}\right) \quad (4)$$

The slips depend on wheel rotational speeds (ω_{ij}), and the longitudinal and lateral speeds of the vehicle in correspondence of the wheel hubs ($V_{x_{ij}}$), reflected in the wheel reference frame. The latter, in turn, are a function of the longitudinal, lateral and yaw speed of the vehicle (u, v and r, respectively), steering angles at the wheels (δ_{ij}), front/rear wheelbase (a_i) and track (t_i); they are calculated as follows [54]:

$$V_{x_{11}} = (u - rt_1/2)\cos(\delta_{11}) + (v + ra_1)\sin(\delta_{11}) \quad (5a)$$

$$V_{x_{12}} = (u + rt_1/2)\cos(\delta_{12}) + (v + ra_1)\sin(\delta_{12}) \quad (5b)$$

$$V_{x_{21}} = (u - rt_2/2)\cos(\delta_{21}) + (v - ra_2)\sin(\delta_{21}) \quad (5c)$$

$$V_{x_{22}} = (u + rt_2/2)\cos(\delta_{22}) + (v - ra_2)\sin(\delta_{22}) \quad (5d)$$

$$V_{y_{11}} = (v + ra_1)\cos(\delta_{11}) - (u - rt_1/2)\sin(\delta_{11}) \quad (5e)$$

$$V_{y_{12}} = (v + ra_1)\cos(\delta_{12}) - (u + rt_1/2)\sin(\delta_{12}) \quad (5f)$$

$$V_{y_{21}} = (v - ra_2)\cos(\delta_{21}) - (u - rt_2/2)\sin(\delta_{21}) \quad (5g)$$

$$V_{y_{22}} = (v - ra_2)\cos(\delta_{22}) - (u + rt_2/2)\sin(\delta_{22}) \quad (5h)$$

Resistant and inertial forces generate a load transfer phenomenon which affects the vertical load of tyres and so their longitudinal and lateral stiffness. To closely estimate vertical loads, roll motion has to be considered. The load transfer generally depends on the longitudinal and lateral acceleration (a_x, a_y) geometric characteristics of the vehicle (i.e. height (h), wheelbase (l), front/rear axle distance from CG (a_i)) and its suspensions (front/rear axle roll stiffness (k_{ϕ_i}), total roll stiffness (k_ϕ) front/rear roll centre (d_i), roll centre at CG coordinate (d)) and the aerodynamics contribution to the loads (a function of air density (ρ_a), cross-sectional area (S_a), front/rear lift coefficient (C_{z_i})) and the square of longitudinal speed [55]

$$F_{z_{1\,1,2}} = \frac{1}{2}\left[\frac{mga_2}{l} + \frac{1}{2}\rho_a S_a C_{z_1} u^2 - \frac{ma_x h}{l}\right] \mp \Delta F_{z_1} \quad (6a)$$

$$F_{z_{2\,1,2}} = \frac{1}{2}\left[\frac{mga_1}{l} + \frac{1}{2}\rho_a S_a C_{z_2} u^2 + \frac{ma_x h}{l}\right] \mp \Delta F_{z_2} \quad (6b)$$

$$\Delta F_{z_1} = \frac{ma_y}{t_1}\left(\frac{a_2}{l} d_1 + \frac{k_{\phi_1}}{k_\phi}(h - d)\right) \quad (7a)$$

$$\Delta F_{z_2} = \frac{ma_y}{t_2}\left(\frac{a_1}{l} d_2 + \frac{k_{\phi_2}}{k_\phi}(h - d)\right) \quad (7b)$$

The tyre model returns the tyre/road interaction forces according to the slips, load and camber conditions. Here is used the Magic Formula 5.2 by Pacejka [55] that does not take into account the pressure variance of the tyre. A non-linear model allows to emulate more critical drive conditions than linear one and to return more realistic dynamic and kinematic output.

The Magic Formula is defined as follows:

$$y(x) = D\sin\{C a\tan[Bx - E(Bx - a\tan(Bx))]\} \quad (8)$$

where x is either slip ratio or slip angle according to the force we are evaluating, $y(x)$ is either F_x or F_y, B is the stiffness factor, C is the shape factor, D is the peak value, E is the curvature factor.

3.2.5 Vehicle dynamics equations: Each wheel exerts longitudinal and lateral forces along his own reference axis. These forces can be expressed in the vehicle body-fixed reference frame as follows [54]:

$$X_{ij} = F_{x_{ij}}\cos(\delta_{ij}) - F_{y_{ij}}\sin(\delta_{ij}) \quad (9a)$$

$$Y_{ij} = F_{x_{ij}}\sin(\delta_{ij}) + F_{y_{ij}}\cos(\delta_{ij}) \quad (9b)$$

where δ_{ij} is the wheel steering angle.

Then vehicle dynamics equations are defined as follows [54]:

$$ma_x = m(\dot{u} - vr) = X = X_1 + X_2 - F_a - F_r \tag{10a}$$

$$ma_y = m(\dot{v} + ur) = Y = Y_1 + Y_2 \tag{10b}$$

$$J_z\dot{r} = N = Y_1a_1 - Y_2a_2 + \Delta X_1t_1 + \Delta X_2t_2 \tag{10c}$$

where F_a is the aereo drag force and F_r is the rolling resistance

$$X_1 = X_{11} + X_{12} \tag{11a}$$

$$X_2 = X_{21} + X_{22} \tag{11b}$$

$$Y1 = Y_{11} + Y_{12} \tag{12a}$$

$$Y2 = Y_{21} + Y_{22} \tag{12b}$$

$$\Delta X_1 = \frac{X_{12} - X_{11}}{2} \tag{13a}$$

$$\Delta X_2 = \frac{X_{22} - X_{21}}{2} \tag{13b}$$

Finally, by integrating longitudinal, lateral and yaw speed, the position and orientation of the vehicle are obtained [54]:

$$\psi(t) = \psi(0) + \int_0^t r(t)\mathrm{d}t \tag{14a}$$

$$x^{CG}(t) = x^{CG}(0) + \int_0^t [u(t)\cos(\psi(t)) - v(t)\sin(\psi(t))]\mathrm{d}t \tag{14b}$$

$$y^{CG}(t) = y^{CG}(0) + \int_0^t [u(t)\sin(\psi(t)) + v(t)\cos(\psi(t))]\mathrm{d}t \tag{14c}$$

3.2.6 Bicycle model: It is possible to simplify the model in order to reduce the computational effort through the so-called *bicycle* model. It differs from the *quadricycle* model since that the vehicle is outlined as single wheel both at front and rear axle, and the lateral load transfer is neglected [56]. This is due to the assumptions made on the manoeuvre; small steering angles imply insignificant yaw rate contribution to longitudinal speed ($u \gg r \cdot t_i$). Another assumption is considering as rigid the steering system and approximating the steering angles to the same value per axle ($\delta_{i1} \simeq \delta_{i2} \simeq \delta_i$).

Since both the slip ratio and slip angle could be considered similar for the wheels of the same axle ($\kappa_{i1} \simeq \kappa_{i2} \simeq \kappa_i$ and $\alpha_{i1} \simeq \alpha_{i2} \simeq \alpha_i$), is possible to represent each axle as a single wheel whose dynamic features refer to the whole axle response. Assumed manoeuvre implies a low lateral inertial force, then small tyre slip angles which let both to linearise trigonometric functions and to consider as linear the behaviour of tyres As regards the vehicle dynamic equations, the bicycle model allows to reduce the numbers of terms (forces F_x and F_y) and neglect the contribution to the yaw motion dynamic balance given by the different longitudinal forces generated by axles ($\Delta X_1t_1 + \Delta X_2t_2$).

3.2.7 Vehicle control actuators: The implemented control actuators system is based on a Proportional–Integral (PI) regulator and a mapping block that adds a contribution to the actuation signal. The functioning of this block is to translate on vehicle actuators the driver's decisions (typically a reference speed or acceleration).

The input of the PI controller is the speed error (difference between desired speed and actual speed of the Ego-vehicle) and applies a correction based on proportional and integral terms, calculating Throttle and Brake signals, in mutual exclusion, corresponding to the accelerator/brake valve's opening (in percentage). It has been tuned using Ziegler–Nichols method on open loop, getting the following values: K_p is equal to 90 and K_i is equal to 0.3. The only use of a PI controller would be ineffective on a complex non-linear system, so the control system has been improved adding a Throttle Mapping block (TMb). The input and output of TMb are summarised in Table 2.

Table 2 Control system input/output

Input	Output
Gear	Throttle, %
reference speed, m/s	Brake, %
vehicle speed, m/s	—

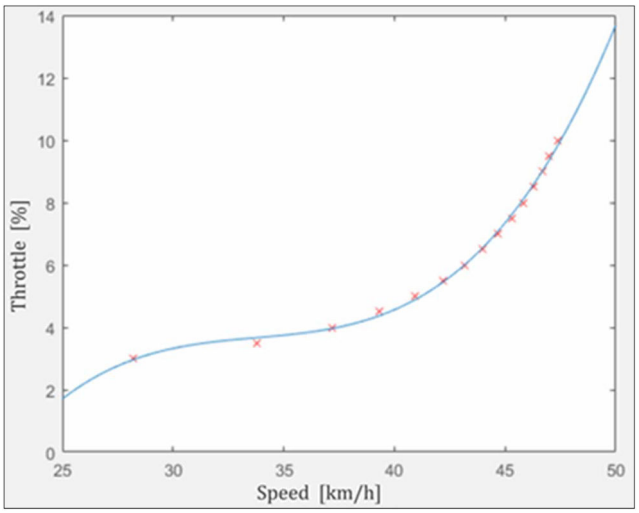

Fig. 3 *Interpolating curve for second gear*

More specifically, the mapping is based on lookup tables (one for each gear) where different values of association between throttle and reachable speed are stored. Then the points of the Speed-Throttle plane have been interpolated through cubic polynomials, obtaining functions that take in input the desired speed and give in output the throttle value:

$$f(V_d) = c_1V_d^3 + c_2V_d^2 + c_3V_d + c_4 \tag{15}$$

In Fig. 3 a typical profile for the interpolating curves is reported. Lastly, the interpolation error is corrected by the throttle output of PI controller. Instead, during braking dynamics, the error signal takes negative values and the Brake signal becomes a scaled version of the absolute value of PI's output. Within the Simulink block named *acc or brk logic* (see Fig. 4) is implemented the control logic that compute the accelerator/brake actuation. The block evaluates the difference between the actual Ego-Vehicle speed and the desired speed given by C-ITS logic. If the difference exceeds the upper threshold, fixed to 8.0 km/h, the brake is activated; when the speed difference has a value between 8.0 km/h and 3.0 km/h (i.e. the fixed lower threshold), the brake is deactivated. Lastly, for values lower than the latter threshold, the vehicle slows down because of the motion resistance. The Ego-Vehicle receives the throttle signal, coming from the mapping block and corrected by the PI controller, and it leads to the target speed. The double threshold value is used to optimise the brake actuation and avoid instability due to continuous switching of the output signal (acceleration or braking) given by *acc or brk logic* block. Then, to ensure the mutual exclusion of Throttle and Brake signals, the Throttle/Brake switch block selects, based on *acc_or_brk* value, the right actuation signal.

To allow the Ego-Vehicle to follow a desired path, its steering angle is commanded as follows [57]:

$$\dot{\delta}_s(t) = -K_s \cdot \omega_{err}(t - \tau) \tag{16}$$

where $\delta_s(t)$ (rad) is the steering angle, K_s is a gain constant, $\omega_{err}(t)$ (rad/s) is the yaw rate error and τ (s) is the response delay. In the considered model, $\omega_{err}(t)$ (rad/s) is defined as the yaw rate that, starting from the actual vehicle position and heading angle, would make the vehicle intersect the desired path after a preview time TP; it is computed as follows:

$$\omega_{err}(t) = \omega(t) - \omega^*(t) \tag{17}$$

where $\omega(t)$ (rad/s) is the current yaw rate and $\omega^*(t)$ (rad/s) is the desired yaw rate. Finally, according to [57], the values of K_s, τ and TP were set, respectively, equal to 10.0, 0.1 and 2.0 s, respectively.

3.3 Procedure

To setup the simulation, some preliminary operations are mandatory. It is necessary to create a simulation scenario in SUMO that it is composed by a network model and a traffic mobility demand where specific traffic flows, defined with their characteristics, surround the EGO-vehicle along its route.

The Ego-Vehicle model defines its kinematic status (longitudinal, lateral, and yaw speed) processing driver inputs such as throttle and braking pedal and steering angle. It provides engaged gear as well to power the control logic and estimates the fuel or energy consumption according to the selected propulsion. The model implemented act as an actuator where the inputs are external. More, those inputs are not necessarily reachable due to mechanical restrains. In this case, the vehicle will reach a different state, different with respect to the desired one.

The external inputs are settled using a Simulink block representing a realistic driver behaviour. The block uses the data collected from the environment, as information of road infrastructure, and turn those into a driving task that has to be implemented. In addition to the environmental data, the Ego-Vehicle status and features and leader features, are sent to the driver model via TRACI4Matlab exploiting the communication architecture depicted in Fig. 5.

Data are used to define target speed and steering angle. The target speed is determined using a car following model, Krauss, that it is the same one implemented in the microscopic simulator software. For lateral dynamics, in order to convert vehicle's yaw rate in steering angle, the Desired Path Yaw Rate Error (DPYRE) driver model [57] has been implemented. The determination of the steering angle is a typical Trajectory Planning problem. Generally, it is assumed that the driver first plans a trajectory to follow, and then applies the steering input to follow it according to the road geometry.

A MATLAB script allows the communication through the tools used to implement the system. The script is composed by multiple functions and is split into two sections. A first one is an *initialisation section* that sets all the necessary parameters and starts the scenario and the vehicle model. Then the second section is dedicated to the *simulation phase*. It allows the continuous exchange of information between the traffic environment and the vehicle model in Simulink. This architecture allows the Ego-Vehicle to be set in the traffic environment with a mutual influence of the two environments.

The *simulation section* is composed by six operational actions for each simulation step (Δt). Firstly, at time t, the data regarding the vehicles surrounding the controlled one are sent to the driver model; then it defines the new status that has to be implemented. As third step the established behaviour is actuated on the vehicle and then the vehicle dynamic model implements the driver inputs to reach the new status defined in the second step. At the end of this step it is defined a reached status that could be not the same defined in the second step. Then, the script moves to the time $t + \Delta t$ and the new status for the controlled vehicle are sent to SUMO. Lastly, SUMO advances at the time $t + \Delta t$ and the scenario interact with the new status of the EGO-vehicle. The new status of the traffic flow is the starting point for the new simulation step and the cycle restarts from the first step.

4 Experiments and results

In this section, we validate the effectiveness of the proposed integrated simulation environment. Specifically, we design two simulation scenarios to understand potentiality, benefits and limits of the realised tool that integrate both traffic micro-simulator and vehicle dynamics.

4.1 Test 1

The first test aimed to compare the performance of the Ego-Vehicle using SUMO first and then the proposed integrated simulation environment. The two different simulations run are based on the same driving scenario. In both cases, the Ego-Vehicle travels along a 3.0 km long road, starting from an initial null speed value. We suppose that speed limits change along the road, in order to show differences in motion of considered vehicles, due to acceleration and deceleration manoeuvres. The two speed profiles are reported in Fig. 6. In this latter, specifically in the first acceleration phase, approximately in the first 15 s, the two speed profiles are very close to each other, while after this point Ego-Vehicle dynamics in the integrated environment leads to a very different motion compared to SUMO, reaching the reference speed with a smoother profile.

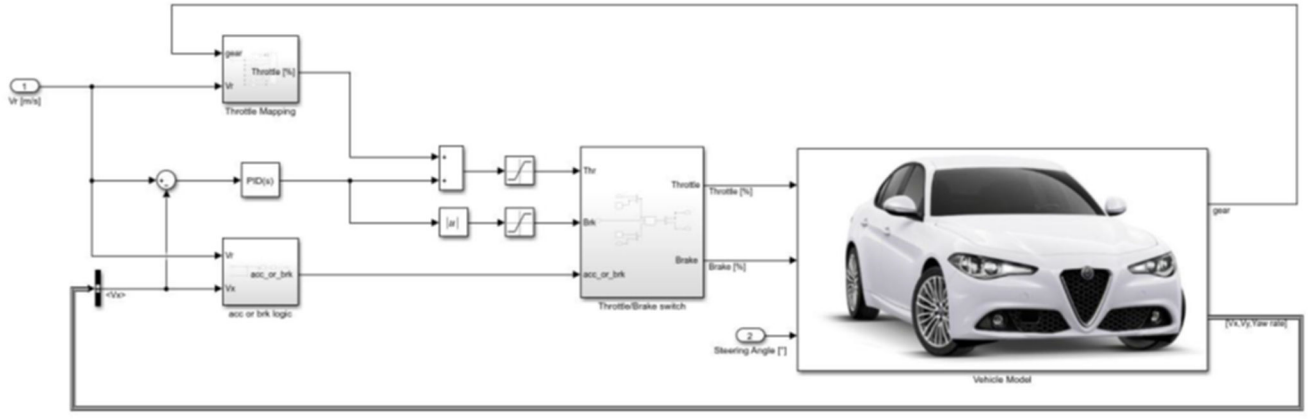

Fig. 4 *Control system in Simulink*

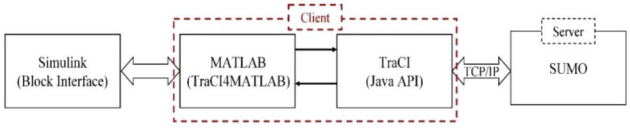

Fig. 5 *Communication Architecture*

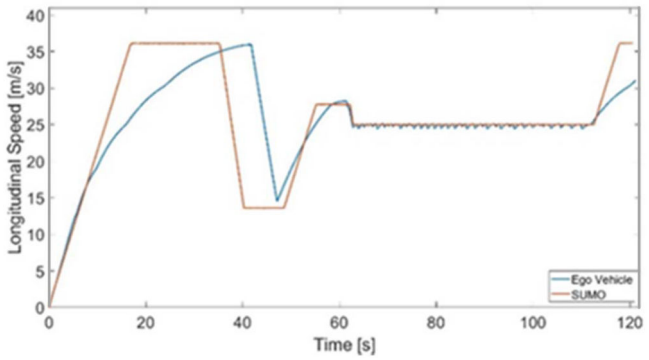

Fig. 6 *Speed profile for the Ego-Vehicle in Test 1*

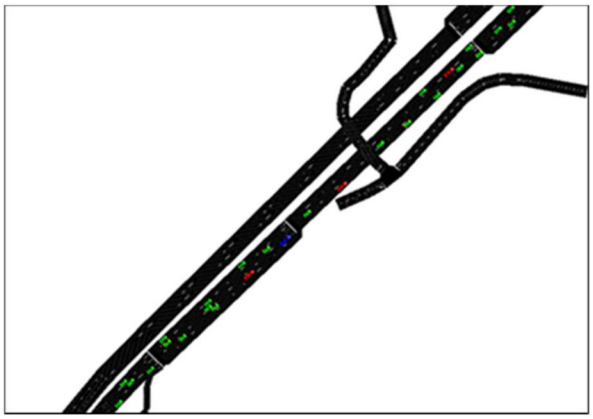

Fig. 7 *Snapshot of the SUMO reference scenario: Ego-Vehicle is represented in blue*

Table 3 IDM parameters

V_0	22.23 m/s
T	1.2 s
S_0	1.5 m
δ	4
a	1 m/s^2
b	2 m/s^2

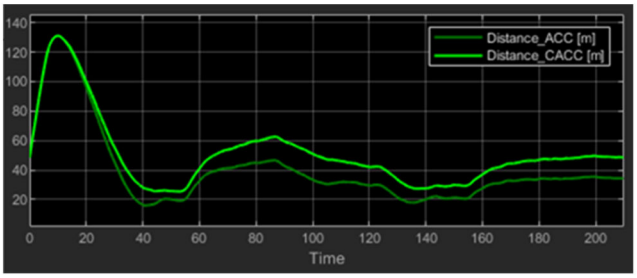

Fig. 8 *Comparison between distances from leader in ACC and C-ACC test condition*

The use of a highly detailed representation of the Ego-Vehicle shows that speed variation is not always linear, not instantaneous, not always constant (e.g. it depends on the speed value itself). This simple example shows how the integration can affect the vehicle behaviour in the simulation and some key variables such travel time, fuel consumption etc.

4.2 Test 2

In the second test, the integrated simulation environment has been used to test, in a realistic traffic condition, an *ACC* and a *C-ACC* control logic.

In Fig. 7 a snapshot of the SUMO road scenario used to test both control logics is depicted. Specifically, we considered a two-lane freeway segment and we represented the Ego-Vehicle in blue to distinguish it from the other surrounding vehicles. The two control logics are first tested and compared to each other, then an example of testing in a complex traffic scenario is showed.

For the design of the ACC control logic, the well-known Intelligent Driver Model (IDM) has been used as a reference; it is a car-following model which generates a target acceleration for the Ego-Vehicle by the following equation [58]:

$$a_{\text{IDM}} = a\left[1 - \left(\frac{V}{V_0}\right)^{\delta} - \left(\frac{S_0 + VT + (V\Delta_V/(2\sqrt{ab}))}{S}\right)\right] \quad (18)$$

where V_0 (m/s) is the desired speed of Ego-Vehicle in free-flow traffic conditions; V (m/s) is the current Ego-Vehicle speed; S_0 (m) is the minimum desired distance at standstill; Δ_V (m/s) is the difference between Ego-Vehicle speed and its leader speed; T (s) is the minimum steady-state time gap; S (m) is the current distance of Ego-Vehicle from its leader; a and b (m/s^2) are the maximum acceleration and comfortable deceleration of the Ego-Vehicle, respectively; δ is the acceleration exponent. In Table 3 the IDM parameters and their values are summarised.

The algorithm has been implemented in Simulink and the reference of the control system has been calculated deriving the target acceleration. The C-ACC algorithm imagines the establishment of a V2V connection, used to evaluate information about other vehicles in a platoon. A logic that evaluates the distance between the nth vehicle in the platoon and its predecessor has been implemented. When specific conditions related to the movement of the vehicles in the platoon occurs, a contribution to the time gap of IDM algorithm are added, varying the reference speed/acceleration of the Ego-Vehicle. This contribution is equal to a scaled version of the ratio between the controlled vehicle velocity and the nth vehicle velocity.

In the first case, the two algorithms have been compared simulating a platoon of four vehicles, in which the last one is the controlled vehicle. For the ACC control system only leader's position and speed are considered, while for the C-ACC position and speed of all platoon vehicles are considered. The results of the application of the two algorithms are shown in Figs. 8 and 9. It is worth noting that in the C-ACC test the speed reference has fewer steep variations and the distance from leader has a higher minimum threshold. This is related to unsafe driving condition of the leader of the Ego-Vehicle that have not been considered by the ACC control logic. Another simulation provides an intense traffic flow on a multi-lane road. In this way it is possible to see the behaviour of the controlled vehicle based on the numerous changes of leader and lane.

In order to make the experiment more understandable, the time series of speed (both for the leading and the Ego-Vehicle), distance and actuators signals have been given in Figs. 10 and 11. It is evident from discontinuities in the spacing signal how complex scenarios could be simply obtained by using the integrated testing environment. At the beginning of the trajectory, the position of the leading vehicle is outside the radar range, and the Ego-Vehicle tends to the desired speed. After that the Ego-Vehicle starts to be strongly influenced by leader and the ACC is enabled. Lane changes, and, more general, stochastic phenomena create several stimuli for the control logic. The corresponding inputs to the vehicle dynamics are reported in Fig. 12.

5 Discussion and conclusions

In this paper, we have provided an overview of the possibilities, the limits, and the possible future developments of the proposed integrated simulation platform. The possibility to build up such a simulation environment within a holistic vision, in which each component of the system is emulated through detailed models, leads to some considerations. On vehicle dynamic side, the use of the proposed platform enhances both testing and validation activities, allowing to analyse the behaviour of both the whole vehicle system and of some of its single component. Furthermore, realistic road traffic scenarios allow to simulate the presence of several different external agents, which can interact with the

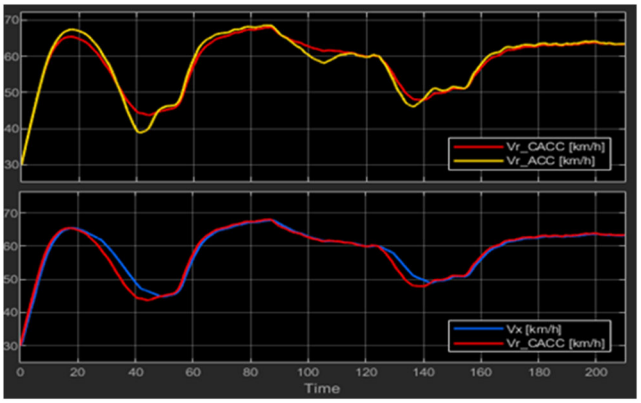

Fig. 9 *Comparison between ACC/C-ACC references and vehicle speed*

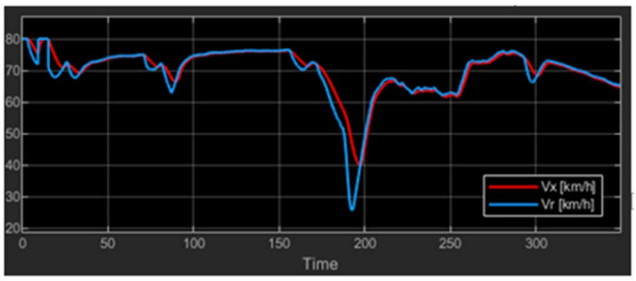

Fig. 10 *ACC reference speed and vehicle speed*

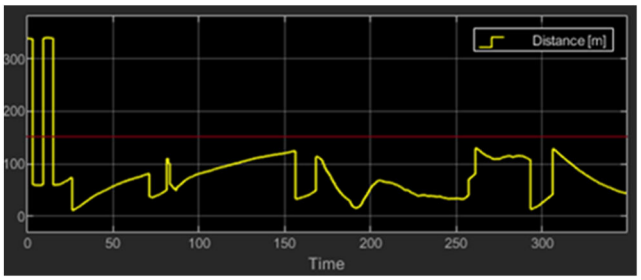

Fig. 11 *Distance from leader and radar threshold*

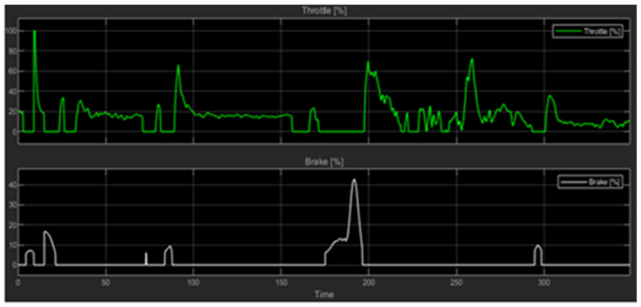

Fig. 12 *Throttle and brake signals*

vehicle, and to modify the characteristic of the traffic flow, both macroscopic and microscopic ones.

On the other hand, i.e. micro-simulation users side, the simulations realism can be improved and, hence, is possible to obtain better results from test activities, such as: (i) more accurate speed profiles and trajectories; (ii) more detailed assessment of consumption and pollutant emissions; (iii) unrealistic behaviour of the driver are avoided; (iv) simple development of ADAS and/or V2X communication systems to observe their impact on the road traffic environment.

By exploiting ease of testing, the low-time required to perform simulation tests and the possibility to vary several road traffic parameters, it is possible to obtain a very wide range of scenarios to test and verify the vehicle behaviour and/or specific control logics. Thereby, the vehicle development process is enhanced since that is possible to promptly fix an eventually non-correct behaviour so to simplify and speed up the subsequent steps.

In this way, it is not necessary anymore to define a priori the set of the worst-case scenarios in which to test the vehicle or one of its components. All above mentioned assessments may be related to the technological context that can be found in the automotive field. Indeed, the latest trend is to create develop vehicles more technologically complex, i.e. automated and even autonomous. Finally, these systems have also to guarantee a high level of safety, i.e. they do not cause damage to other things or people. With the aim to combine complexity and safety requirements, it is necessary to carry out a high number of simulation tests. To achieve these goals, the using of such virtual testing environment seems to be fundamental.

6 References

[1] World Health Organization, 2018, Global status report on road safety 2018;, Summary (No. WHO/NMH/NVI/18.20). World Health Organization

[2] Pariota, L., Bifulco, G.N., Markkula, G., *et al.*: 'Validation of driving behaviour as a step towards the investigation of connected and automated vehicles by means of driving simulators'. Proc. 5th IEEE Int. Conf. Models Technologies Intelligent Transportation System (MT-ITS), Naples, Italy, June 2017, pp. 274–279

[3] Höfflinger, B., Estève, G.C.D., Weisglas, P.: 'Integrated electronics for automotive applications in the EUREKA program PROMETHEUS', Grenoble, France, 1989, pp. 13–17

[4] Jim, F., Naranjo, E., Serradilla, F., *et al.*: 'Communications and driver monitoring aids for fostering SAE level-4 road vehicles automation', *Electronics*, 2018, **7**, (10), p. 228

[5] Reichardt, D., Miglietta, M., Moretti, L., *et al.*: 'CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication', *Intelligent Vehicle Symposium*, 2002, **2**, pp. 545–550

[6] Ma, Y., Chowdhury, M., Sadek, A., *et al.*: 'Real-time highway traffic condition assessment framework using vehicleInfrastructure integration (VII) with artificial intelligence (AI)', *IEEE Trans. Intell. Transp. Syst.*, 2009, **10**, (4), pp. 615–627

[7] Petrillo, A., Pescapé, A., Santini, S.: 'A collaborative control strategy for platoons of autonomous vehicles in the presence of message falsification attacks'. Proc. 5th IEEE Int. Conf. Models Technologies Intelligent Transportation Systems (MT-ITS), Naples, Italy, June 2017, pp pp. 110–115

[8] Spiliopoulou, A., Perraki, G., Papageorgiou, M., *et al.*: 'Exploitation of ACC systems towards improved traffic flow efficiency on motorways'. Proc. 5th IEEE Int. Conf. Models Technologies Intelligent Transportation Systems (MT-ITS), Naples, Italy, June 2017, pp. 37–43

[9] Jiménez, F.: 'Connected vehicles, V2V communications, and VANET', *Electron.*, 2015, **4**, (3), pp. 538–540

[10] Li, L., Lin, Y.-L., Zheng, N.-N., *et al.*: 'Artificial intelligence test: a case study of intelligent vehicles', *Artificial Intell. Rev.*, 2018, **50**, (3), pp. 441–465

[11] Krajzewicz, D., Erdman, J., Behrisch, M., *et al.*: 'Recent development and application of SUMO-simulation of urban MObility', *Int. J. Adv. Syst. Meas.*, 2012, **5**, (3 & 4), pp. 128–138

[12] PTV AG, Vissim homepage [Online], available at http://www.ptvvision.com/en-uk/products/vision-traffic-suite/ptvvissim/overview/, accessed July 03, 2012

[13] MATSim homepage [Online], available at http://www.matsim.org/, accessed July 03, 2012

[14] Barcelò, J.: *'Fundamentals of traffic simulation'* (Springer, New York, 2010)

[15] Cartenì, A.: 'Urban sustainable mobility. Part 2: simulation models and impacts estimation', *Transp.Probl.*, 2015, **10**, (1), pp. 5–16

[16] Yu, L., Zhang, Y., Song, G.: 'Applicability of traffic microsimulation models in vehicle emissions estimates: case study of VISSIM', *Transp. Res. Rec. J. Transp. Res. Board*, **2270.1**, 2012, pp. 132–141

[17] Boubaker, S., Rehimi, F., Kalboussi, A.: 'Impact of intersection type and a vehicular fleet's hybridization level on energy consumption and emissions', *J. Traffic Transp. Eng. (English Ed.*, 2016, **3**, (3), pp. 253–261

[18] Samaras, C., Tsokolis, D., Toffolo, S., *et al.*: 'Improving fuel consumption and CO_2 emissions calculations in urban areas by coupling a dynamic micro traffic model with an instantaneous emissions model', *Transp. Res. Part D Transp. Environ.*, 2018, **65**, (November 2017), pp. 772–783

[19] Hoogendoorn, S.P., Minderhoud, M.M.: 'ADAS impact assessment by micro-simulation', *Traffic Eng.*, 2001, **1**, (3), pp. 255–275

[20] Tapani, A.: 'Vehicle trajectory effects of adaptive cruise control', *J. Intell. Transp. Syst. Technol. Planning, Oper.*, 2012, **16**, (1), pp. 36–44

[21] Albert, G., Hakkert, S., Toledo, T.: 'Impact of active speed limiters on traffic flow and safety: simulation-based evaluation', *Transportation research record*, 2007, **2019**, (1), pp. 169–180

[22] Hegeman, G., Tapani, A., Hoogendoorn, S.: 'Overtaking assistant assessment using traffic simulation', *Transp. Res. Part C Emerg. Technol.*, 2009, **17**, (6), pp. 617–630

[23] Benz, T., Christen, F., Lerner, , *et al.*: 'Traffic effects of driver assistance systems-the approach within INVENT', *IEEE Trans. Intell. Transp. Syst.*, 2018, **7**, (4), pp. 103–113

[24] Semrau, M., Erdmann, J., Friedrich, B., *et al.*: 'Simulation framework for testing ADAS in Chinese traffic situations'. SUMO 2016 Conf. Proc., Berlin, Germany, 2016, pp. 103–113

[25] Jung, J., Bae, S.H.: 'Real-time road lane detection in urban areas using LiDAR data', *Electron.*, 2018, **7**, (11), pp. 1–14
[26] Bifulco, G.N., Simonelli, F., Di Pace, R.: 'Endogenous driver compliance and network performances under ATIS'. IEEE Conf. Intell. Transp. Syst. Proc., ITSC, Seattle, WA, USA, 2007, pp. 1028–1033
[27] Segata, M., Joerer, S., Bloessl, B., *et al.*: 'Plexe: A platooning extension for Veins', *IEEE Veh. Netw. Conf. VNC*, 2015, **2015-Janua**, (January), pp. 53–60
[28] Segata, M.: 'Platooning in SUMO: an Open source Implementation'. SUMO User Conf., Berlin, Germany, 2017, pp. 51–62
[29] Petrillo, A., Salvi, A., Santini, S., *et al.*: 'Adaptive multi-agents synchronization for collaborative driving of autonomous vehicles with multiple communication delays', *Transp. Res. Part C Emerg. Technol.*, 2018, **86**, (November 2017), pp. 372–392
[30] Di Vaio, M., Fiengo, G., Petrillo, A., *et al.*: 'Cooperative shock waves mitigation in mixed traffic flow environment', *IEEE Trans. Intell. Transp. Syst.*, 2019, **PP**, pp. 1–15
[31] İsmet Gökşad, E., Mehmet Ali, S., Hilmi Berk, Ç.: 'Emission effects of cooperative adaptive cruise control: a simulation case using SUMO'. SUMO User Conf. 2019, Berlin, Germany, 2019, Vol. 62, pp. 92–100
[32] Yang, L., Yang, J.H., Feron, E., *et al.*: 'Development of a performance-based approach for a rear-end collision warning and avoidance system for automobiles'. IEEE Intelligent Vehicles Symp. Proc., Columbus, OH, USA, 2003, pp. 316–321
[33] Chang, C.Y., Chou, Y.R.: 'Development of fuzzy-based bus rear-end collision warning thresholds using a driving simulator', *IEEE Trans. Intell. Transp. Syst.*, 2009, **10**, (2), pp. 360–365
[34] Isermann, R., Schaffnit, J., Sinsel, S.: 'Hardware-in-the-loop simulation for the design and testing of engine-control systems', *Control Eng. Pract.*, 1999, **7**, (5), pp. 643–653
[35] Gietelink, O., Ploeg, J., De Schutter, B., *et al.*: 'Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations', *Veh. Syst. Dyn.*, 2006, **44**, (7), pp. 569–590
[36] Kang, Y., Yin, H., Berger, C.: 'Test your self -driving algorithm: an overview of publicly available driving datasets and virtual testing environments', *IEEE Trans. Intell. Veh.*, 2019, **4**, (2), pp. 171–185
[37] Himmler, A.: 'Hardware-in-the-loop technology enabling flexible testing processes', *Hardware-in-the-Loop Technol. Enabling Flex. Test. Process*, 2013
[38] Wymann, B., Espíe, E., Guionneau, C., *et al.*: 'TORCS, the open racing car simulator', Software available at http://torcs. sourceforge. net, vol. 4, no. 6, 2000
[39] Naumann, M., Poggenhans, F., Lauer, M., *et al.*: 'CoInCar-Sim: an open-source simulation framework for cooperatively interacting automobiles'. 2018 IEEE Intelligent Vehicles Symp. (IV), Changshu, Suzhou, China, 2018, pp. 1–6
[40] Dosovitskiy, A., Ros, G., Codevilla, F., *et al.*: 'CARLA: an open urban driving simulator', arXiv preprint arXiv:1711.03938, 2017
[41] Shah, S., Dey, D., Lovett, C., *et al.*: 'AirSim: high-fidelity visual and physical simulation for autonomous vehicles,' in: *'Field and service robotics'* (Springer, Cham, 2018), pp. 621–635
[42] Bojarski, M., Del Testa, D., Dworakowski, D., *et al.*: 'End to end learning for self-driving cars', arXiv preprint arXiv:1604.07316, 2016
[43] Kaths, J., Krause, S.: 'Integrated simulation of microscopic traffic flow and vehicle dynamics', IPG Apply & Innovate 2016
[44] Nalic, D., Eichberger, A., Hanzl, G., *et al.*: 'Development of a Co-simulation framework for systematic generation of scenarios for testing and validation of automated driving systems'. 2019 IEEE Intelligent Transportation Systems Conf. (ITSC), Auckland, NZ, October 27–30 2019
[45] Cottignies, A., Daley, M., Newton, E., *et al.*: 'rFpro & SUMO: the road to a complete real-time simulation of urban environments for DIL, ADAS and autonomous testing'. In SUMO 2017: Towards Simulation of Autonomous Mobility, 2017
[46] (Jason) So, J., Motamedidehkordi, N., Wu, Y., *et al.*: 'Estimating emissions based on the integration of microscopic traffic simulation and vehicle dynamics model', *Int. J. Sustain. Transp.*, 2018, **12**, (4), pp. 286–298
[47] Rajendran, A.V., Hegde, B., Ahmed, Q., *et al.*: 'Design and development of traffic-in-loop powertrain simulation'. 2017 IEEE Conf. on Control Technology and Applications (CCTA), Hawaii, USA, 2017, pp. 261–266
[48] Campāna, E.M., Mullner, N., Mubeen, S.: 'Interfacing a brake-bywire simulink model with SUMO'. 2018 Int. Conf. on Intelligent and Innovative Computing Applications (ICONIC), Plaine Magnien, Mauritius, 2018, pp. 1–6
[49] Griggs, W.M., Ordonez-Hurtado, R.H., Crisostomi, E., *et al.*: 'A large-scale SUMO-based emulation platform', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (6), pp. 3050–3059
[50] Punzo, V., Ciuffo, B.: 'Integration of driving and traffic simulation: issues and first solutions', *IEEE Trans. Intell. Transp. Syst.*, 2011, **12**, (2), pp. 354–363
[51] Hou, Y., Zhao, Y., Hulme, K.F., *et al.*: 'An integrated traffic-driving simulation framework: design, implementation, and validation', *Transp. Res. Part C Emerg. Technol.*, 2014, **45**, pp. 138–153
[52] OpenStreetMap homepage [Online], available at http://www.openstreetmap.org/, accessed July 03, 2012
[53] OpenDRIVE consortium, OpenDRIVE homepage [Online], available at http://www.opendrive.org/, accessed July 03, 2012
[54] Guiggiani, M.: *'The science of vehicle dynamics'* (Springer, Pisa, Italy, Netherlands, 2014)
[55] Pacejka, H.: *'Tyre and vehicle dynamics'* (Elsevier, 2005)
[56] Liniger, A., Domahidi, A., Morari, M.: 'Optimization-based autonomous racing of 1: 43 scale RC cars', *Optimal Control Appl. Meth.*, 2015, **36**, (5), pp. 628–647
[57] Markkula, G., Romano, R., Jamson, A.H., *et al.*: 'Using driver control models to understand and evaluate behavioural validity of driving simulators', *IEEE Trans. Human-machine Syst.*, 2018, **48**, (6)
[58] Milanés, V., Shladover, S.E.: 'Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data', 2014, pp. 285–300