Integration of Driving and Traffic Simulation: Issues and First Solutions

Vincenzo Punzo and Biagio Ciuffo

Abstract—**Driving simulators are very suitable test beds for the evaluation and development of intelligent transportation systems (ITSs). However, the impact of such systems on the behavior of individual drivers can properly be analyzed through driving simulators only if autonomous vehicles in the driving scenario move according to the system under evaluation. This condition means that the simulation of the traffic surrounding the interactive vehicle should already take into account the driver's behavior as affected by the system under analysis. Currently, this "loop" is not properly tackled, because the effects on individuals and traffic are, in general, separately and, often, independently evaluated. The integration of traffic and driving simulations, instead, may provide a more consistent solution to this challenging evaluation problem. It also opens up new scenarios for enhancing the credibility of both traffic modeling and driving simulation and for their combined development. For instance, because drivers directly interact with driver/traffic models in a driving simulation environment, such models may also be tested against nonnormative behavior, and this case seems the only way to test driver/traffic models for safety applications. Based on this idea, this paper describes the integration of a driving simulation engine known as SCANeR and a traffic-flow microsimulation model known as AIMSUN. Methodological and technical issues of such integration are first presented, and future enhancements for higher consistency of the simulation environments are finally envisaged.**

Index Terms—**Driver simulation, driving simulation, intelligent transportation systems (ITSs), ITS evaluation, traffic-flow microsimulation.**

I. Introduction

DRIVING simulators may very usefully be employed to evaluate and develop intelligent transportation systems (ITSs) such as cooperative systems and advanced driver-assistance systems (ADASs). A number of studies have been carried out with this aim (to provide a few recent examples, see [1] and [2], in which driving simulation experiments were used to test a rear-end collision warning and avoidance system, [3], in which traffic-calming strategies were tested in a simulated environment, or [4], where driving simulations were applied to analyze drivers' behavior during road accidents).

Manuscript received February 5, 2009; revised February 11, 2010 and August 16, 2010; accepted October 23, 2010. Date of publication December 23, 2010; date of current version June 6, 2011. The Associate Editor for this paper was R. Liu.

V. Punzo is with the Department of Transportation Engineering "L. Tocchetti," University of Naples Federico II, 80125 Napoli, Italy (e-mail: vinpunzo@unina.it).

B. Ciuffo is with the Institute for Environment and Sustainability, Joint Research Centre, European Commission, 21020 Ispra (Varese), Italy (e-mail: biagio.ciuffo@jrc.ec.europa.eu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2010.2095846

The main advantage of using a driving simulator is the chance to perform safe experiments in a controlled environment, ensuring the same driving conditions for all test drivers. Nonetheless, the reliability of results is seriously affected by the believability of the simulation environment. The behavior of the surrounding vehicles, which is generally referred to as the traffic scenario, is recognized as a key element in realistic simulations. Indeed, the impact of ITS applications on the behavior of individual drivers can thoroughly be analyzed through driving simulators only if autonomous vehicles in the driving scenario move according to the same system under evaluation. This condition means that the simulation of the traffic surrounding interactive vehicles should already take into account the driver's behavior as affected by the system under analysis. Unfortunately, such a "loop" is not properly tackled.

Indeed, a realistic traffic simulation generally requires the following conditions.

1) Autonomous vehicles move around the interactive vehicle in a realistic manner, i.e., showing realistic kinematics and vehicle interactions. In particular, because ITSs may impact driver behavior, traffic flow in the driving simulation should accordingly behave. For instance, in the case of ADAS, the ability of autonomous vehicles to compatibly move with such systems is clearly advisable for the comprehensive evaluation of the impact on drivers and traffic flows.
2) Vehicles move according to the macroscopic behavior of traffic streams (e.g., the higher the number of vehicles on the road, the lower the average speed of the traffic stream, which can also reproduce distinctive characteristics of congested traffic patterns).
3) In some experiments (e.g., for studies with regard to the effect of traffic information on driving and path/choice behavior), while moving on a network, the driver of the interactive vehicle should meet the "same" traffic that he/she expects to find in the real network, i.e., the dynamic pattern of the traffic flow on the simulated network should closely resemble the real pattern.

To fulfill the first requirement, accurate driver models are necessary. This case is the point on which the field research community has mostly focused its attention [5]–[14]. The second point requires that the aggregate behavior of a traffic stream, resulting from individual behavior, is consistent with the macroscopic laws and characteristics of traffic flow [14]–[16]. This feature can be achieved, for example, by calibrating individual driver model parameters through aggregate traffic data such as time series of counts or speeds at detectors

[17], [18] instead of microscopic data [19]. Finally, the third point requires that traffic simulation is extended to the whole network [14].

When these conditions are accomplished through a macroscopic or mesoscopic model, the effects of some ITS strategies on the network flow propagation cannot properly be accounted for, which is also the case of oversimplified microscopic models (e.g., [20] and [21] implemented a microscopic model only for the vehicles that surround the interactive one, also simplifying the road infrastructure and the representation of traffic lights). Indeed, when the aim of a study is to design and evaluate systems that affect individual driver behavior and choices (e.g., in-vehicle information systems [22]), detailed microscopic models are needed.

To satisfy these requirements, some authors have proposed to integrate the driving simulation environment with commercial microscopic traffic-simulation tools (e.g., [13], [22], and [23]). In the former paper, the traffic-simulation tool applied was Paramics [24], whereas VISSIM [25] was used in the latter two papers. The findings in these papers showed the usefulness of the integration and shed light on the existing shortcomings. However, in [13], due to issues that concern the import/export capabilities of the simulation programs and to data transfer delays, only a one-way communication flow was established. In [23], even if real-time integration between driving and traffic simulation is strongly advised, due to implementation difficulties, the authors only tested offline integration, using results from one instrument as input to the other. By contrast, in [22], full integration is implemented, but the software used in that case (since 2003, there have been advances in both driving and traffic simulation) did not allow the potential of the integrated environment to fully be exploited. Hence, we present a new attempt to integrate a microscopic traffic-simulation model and a driving simulation environment. The motivations behind this attempt are provided in Section II. Then, in Section III, issues behind the integration are outlined, whereas a solution for the case study is proposed in Section IV, together with the description of the main features concerning the selected vehicle movement model. Further comments on the strategy adopted and final remarks on the experiences carried out conclude this paper in Sections V and VI.

II. Traffic Simulation in Driving Simulation Experiments

Current major limitations of microscopic traffic simulation lie in the scarce ability of models to reproduce some behavior such as conflicting traffic situations (e.g., merging at freeway on-ramps and roundabouts) and in the difficulty in calibrating existing models on actual traffic data to increase model reliability.

The rising attention of the scientific community on traffic-simulation calibration and validation (see, for example, the recent activation of the European COST Action TU0903—MULTITUDE [26]) is likely to increase traffic-flow model reliability in the medium term.

Nonetheless, structural flaws in current traffic-flow models still exist, independent of the actual possibility of calibrating them. Indeed, microscopic traffic-simulation models basically consider the following three main dynamics to date: 1) car following; 2) lane changing; and 3) gap acceptance or merging. The first approach seeks to capture the longitudinal behavior of vehicles, traditionally as a response to the stimuli from the front vehicle. The second approach concerns the decision of a vehicle to change lanes to overtake its leader or follow its route, whereas the third approach mainly regards the behavior of a vehicle in merging situations.

However, such classes of models usually lead to a coarse representation of vehicle movements. Major limitations are recognized in the following three cases: 1) The previous dynamics are independently modeled in most commercial simulation software (e.g., lane-changing decisions do not affect acceleration); 2) the representation of movements is monodimensional (i.e., no lateral acceleration is considered, and as a consequence, lane changing is quite instantaneous), and there is no actual representation of road space, which does not allow us to properly account for nonnormative behavior or to model, e.g., motorcycles or even overtaking, in two-lane rural highways; and 3) such models do not account for high-level tactical tasks that affect driving, such as cooperative behavior in merging situations.

Furthermore, the reason that these models are commonly used is their ability, once calibrated, to reproduce the macroscopic behavior of the traffic flow, which is one of the requirements listed in the previous section and is the main feature required for traffic studies. In this context, the integration of traffic- and driving-simulation models may contribute to the development and enhancement of both traffic modeling and driving simulation

From a driver/traffic modeling perspective, a driving simulation environment is a powerful (as well as “dangerous”) tool to gather data to develop and validate driver and traffic models, partly given the repeatability of experiments as well as the chance to test hazardous driving situations. The driving simulation environment makes drivers directly interact with driver/traffic models and allows such models to be tested and analyzed, for instance, against nonnormative driver behavior. This condition is the only current way of testing models for such situations and open up traffic modeling to safety evaluation. In addition, visual validation, which is a fundamental step in driver/traffic model development already enhanced by the recent development of 3-D visualization of traffic simulation, is becoming an ever more powerful tool within the virtual reality environment of driving simulators.

From a driving simulation perspective, when it is required to test the driver’s reaction and its compliance with advanced traffic management or information systems, a built-in network microsimulation tool allows easy implementation of each traffic controller, and more importantly, it allows network traffic dynamics to be reproduced within the driving scenario. The last feature is fundamental, for example, in path-choice-related applications (e.g., route guidance and navigation systems evaluation).

In this paper, it is claimed that these features become essential when complex road/traffic technologies, such as cooperative systems, have to be evaluated. Indeed, the impact

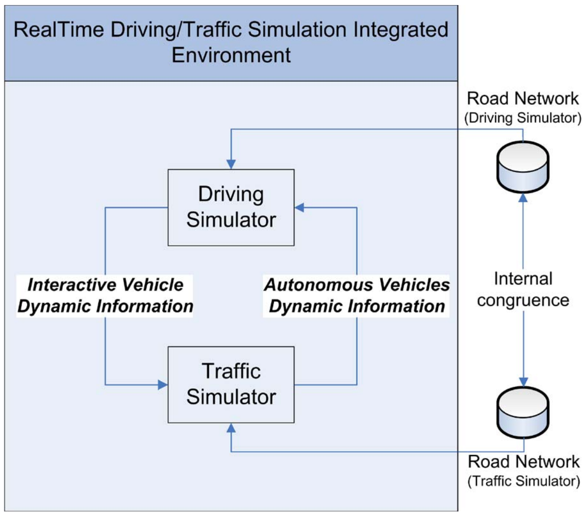

RealTime Driving/Traffic Simulation Integrated Environment
Road Network (Driving Simulator)
Driving Simulator
Interactive Vehicle Dynamic Information
Autonomous Vehicles Dynamic Information
Internal congruence
Traffic Simulator
Road Network (Traffic Simulator)

Fig. 1. Communication framework.

of these systems on the behavior of individual drivers can properly be analyzed through driving simulators only if autonomous vehicles in the driving scenario move according to the system under evaluation. This condition would mean that the simulation of the traffic surrounding the interactive vehicle would already take driver behavior into account. Currently, this "fixed-point problem" is not properly tackled, and effects on individuals and traffic are separately evaluated (and generally independently). In other words, initial effects of technology on individual behavior are evaluated through driving simulators, and then, the effects on traffic are evaluated by using traffic models. Instead, the coupling of traffic and driving simulations could well provide a more consistent solution to this problem.

Due to rapid enhancements in traffic simulation, commercial packages suggest improving such existing models, rather than developing them from scratch, beginning in a driving scenario, to take advantage of their numerous features.

The integration of driving and traffic simulations is both a technical and methodological challenge. From a technical point of view, the main issue is to synchronize the two simulation environments both in time and space and to make them exchange information in real time. From a methodological point of view, the consistency of the levels of detail of the two modeling environments has to be guaranteed. In the following discussion, solutions to technical issues are first presented, and future enhancements are then envisaged to allow higher consistency of the two simulation environments.

III. Integration Framework of Driving and Traffic Simulation

The integration of a driving simulation and traffic simulation environment may be achieved by implementing a communication structure that can effectively exchange data between the two applications (see Fig. 1). The integrated environment has to ensure that, during the driving simulation, the autonomous vehicles are consistently moved by the traffic microsimulation model with the movements of the interactive vehicle. Furthermore, during the simulation, the software that manages the driving simulator needs to send the kinematic characteristics of the interactive vehicle (e.g., position, angle, and speed) to the traffic-simulation model. Once such information has been received, the traffic-simulation model can calculate kinematics of the surrounding traffic for the step ahead and send them back to the driving simulation environment, which has the task of updating this information in the driving scenario.

To fulfill these objectives, it is necessary to choose a communication strategy that can perform the data exchange in real time and ensure the internal congruence between the road networks created in both environments.

A. Main Issues for Integration

To implement the communication framework, it is necessary to tackle several issues, which are summarized as follows.

Accurate Road Matching Between Traffic and Driving Modules: The first issue for integration is the consistency between the road networks: Position related data to be exchanged have to be framed within a well-known reference system to allow appropriate conversion between the two environments during communication. The two road networks thus have to exactly match to have strict correspondence between the two simulation environments.

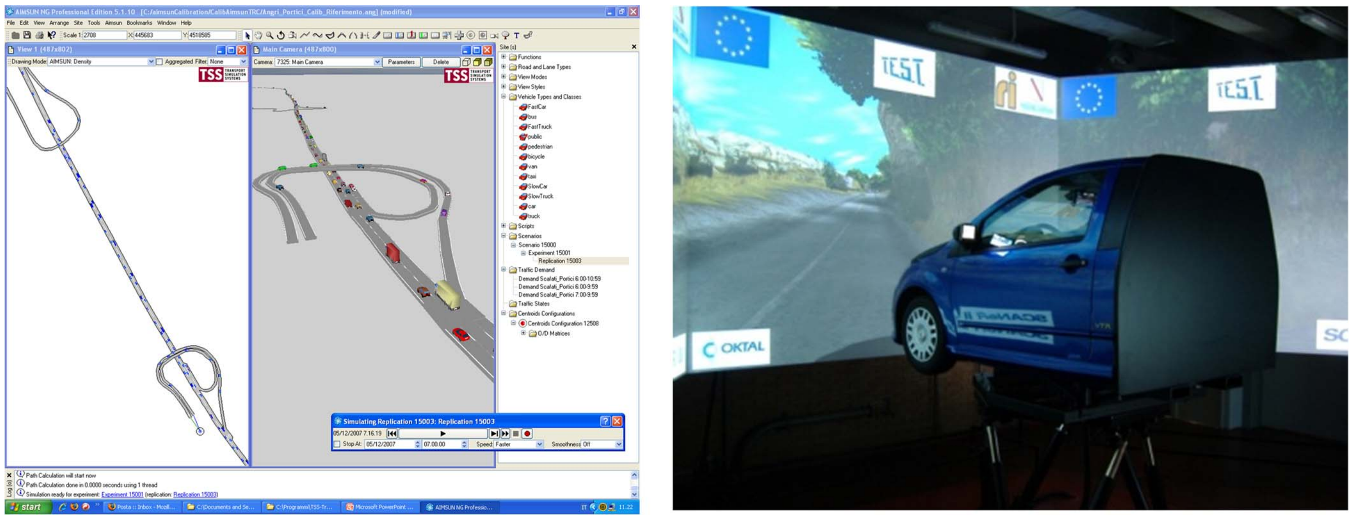
With regard to the first aspect, there is a simple way of overcoming the problem. Network creation usually starts in both software applications with importing the road alignment. Both types of applications should import drawing files (.dwg) or shape files (.shp). Thus, it is only important to avoid any transformation of geographical information during the import process to ensure the consistency of the two reference systems. This way, data retrieved by the two environments are ready to be exchanged.

As for the road matching problem, this case is mainly because a driving simulator requires very detailed road geometry to present the road as realistically as possible for the driver, which is obviously far from the purposes of a microscopic traffic simulator, where, in most cases, for instance, the road alignment curvature, even when represented in detail, does not affect vehicle behavior.

Synchronization of Traffic and Driving Modules With Real Time: As aforementioned, the communication framework has to allow real-time data exchange required to perform a driving simulation. Unlike the driving simulation engine, traffic microsimulators are not designed to work in real time, but according to the hardware performance and scenario complexity, they try to perform all the needed calculations in the shortest time. This condition means that the time needed for a simulation is usually shorter than the real time and generally involves the need to slow down the computing speed of the traffic-simulation package.

However, if the amount of data to be exchanged is large, the traffic module may slow down to less than real time. This case would naturally rule out any chance of successfully integrating the modules.

Consistency of the Updating Calculation Frequency: Each simulation model performs calculations at a given frequency. The finer the desired outputs are, the higher the updating


Simulating Replication 15003: Replication 15003
Stop At: 05/12/2007 07.00.00
Speed: Faster
Smoothness: Off
TSS
T.E.S.T.
OKTAL

Path Calculation will start now
Path Calculation done in 0.0000 seconds using 1 thread
Simulation ready for experiment: Experiment 15001 (replication: Replication 15003)

Fig. 2. Road Safety Laboratory, Technology for Transport, Environment and Safety (T.E.S.T. S.c.a.r.l.). Driving and traffic simulation.

frequency of calculations becomes. Both the microscopic traffic-simulation model and driving simulation software work this way. The difference is that the minimum simulation step in traffic simulation is usually 0.1 s, whereas realistic visualization in a driving simulation environment needs information to be updated at a higher frequency (generally 60 Hz). Therefore, if higher simulation frequency is not allowed by the microsimulation software (or it would slow down the simulation speed to below the real-time speed), outputs have to be augmented, e.g., through interpolation.

Management of Autonomous Vehicle Visualization: This issue is twofold. First, a microscopic traffic simulation can manage thousands of vehicles at the same time. The management of such a large number of vehicles in a driving simulation environment is unfeasible—for the aforementioned real-time issue—and useless for driving simulation purposes. Hence, only the information of vehicles that surround the interactive one is exchanged during the simulation, which can be done by defining a "bubble" around the interactive vehicle. It is thus necessary to create, on the one hand, a vehicle in the traffic model to reproduce the movements of the interactive vehicle (maneuvered by the driving simulator) in the traffic-simulation environment. On the other hand, in the driving-simulation environment, a certain number of "ghost" vehicles have to be created, which come into action when some vehicles enter the bubble in the traffic simulation. When a vehicle enters the driving simulation, a clear one-to-one correspondence has to be created with the corresponding one in the traffic simulation. More specifically, as long as a vehicle in the traffic simulation stays within the surroundings of the interactive vehicle, it always has to be represented by the same vehicle in the driving simulation.

Both aspects of this issue have to specifically be dealt with and, depending on how the solutions are implemented, can seriously affect the possibility of having a real-time data exchange.

The aforementioned issues do not constitute the exhaustive list of problems to deal with when implementing integration. They are, however, the most basic issues to obtain an effective framework. Additional issues may arise, depending on the particular driving and traffic simulation models used, as well as on the particular strategy adopted. The framework implemented in a particular case study and the way in which such issues have been taken into account are shown as follows.

IV. Case Study

This section describes the experience carried out in the integration of the traffic platform AIMSUN [27] and the dynamic driving simulator Virtual Environment for Road Safety (VERA) at the Road Safety Laboratory, Technology for Transport, Environment and Safety (T.E.S.T. S.c.a.r.l.) [28] (see Fig. 2).

The openings provided for plugging in any user-defined driver's model into the simulation framework and for customizing the behavior of individual vehicles, e.g., through the application programming interface (API) or software developer's kit (SDK), proves essential for straightforwardly implementing and emulating systems such as ADAS or reproducing non-normative behavior [29] and was one of the main reasons for choosing that traffic-simulation platform.

In the following discussion, some details on the traffic modeling in AIMSUN will be briefly presented. In addition, the software architecture adopted will be described, along with some practical details of the implementation.

A. Traffic Modeling in AIMSUN

As already pointed out, AIMSUN is a complex platform for traffic simulation. Although it is usually known as a microscopic traffic simulator, in recent years, it has broadened its capabilities, allowing for static macrosimulation and dynamic mesosimulation. In addition, apart from the standard capabilities offered for transportation supply system modeling (e.g., different roads and vehicle categories, intersections, signalization, and tamp metering), it can model the interaction of road traffic with public transport and pedestrian flows (with the new integration with the Legion pedestrian simulator [30]).

With regard to modeling vehicle movements at a microscopic level, AIMSUN simulations imply models for the drivers' car-following and lane-changing behavior.

Car-Following Model in AIMSUN: The car-following model implemented in AIMSUN is based on the well-known Gipps model [31]. This model pertains to the class of the "safety distance" or "collision avoidance" models. Models of this class aim at specifying a safe following distance and adapt the driver's behavior to always maintain it. In practice, the Gipps model assumes that the following driver chooses his/her speed such that he/she can keep the minimum distance at a standstill whenever the leader brakes at its maximum deceleration rate.

Thus, according to Gipps model, the speed $\dot{\mathrm{s}}_\mathrm{f}$ attained by a vehicle at a given time instant $(t+\tau)$ (in which the delay τ is the "apparent" driver's reaction time [31]) is given by

$$\dot{\mathrm{s}}_\mathrm{f}(t+\tau) = \min(\dot{\mathrm{s}}_{\mathrm{f,acc}}, \dot{\mathrm{s}}_{\mathrm{f,dec}}) \tag{1}$$

in which $\dot{\mathrm{s}}_{\mathrm{f,acc}}$ and $\dot{\mathrm{s}}_{\mathrm{f,dec}}$ are, respectively, the vehicle's speed in acceleration and deceleration phase, which are given by

$$\dot{\mathrm{s}}_{\mathrm{f,acc}} = \dot{\mathrm{s}}_\mathrm{f}(t) + \alpha \cdot \ddot{S}_f \cdot \tau \cdot \left(1 - \frac{\dot{\mathrm{s}}_\mathrm{f}(t)}{\dot{S}_f}\right) \cdot \left(\beta + \frac{\dot{\mathrm{s}}_\mathrm{f}(t)}{\dot{S}_f}\right)^{\gamma} \tag{2}$$

$$\dot{\mathrm{s}}_{\mathrm{f,dec}} = A + \sqrt{A^2 - B} \tag{3}$$

in which

$$A = b_f \cdot \left(\frac{\tau}{2} + \theta\right) \tag{4}$$

$$B = b_f \cdot \left[2\left(s_l(t) - s_f(t) - (L_l + \Delta S_0)\right) - \dot{s}_f(t) \cdot \tau - \frac{\dot{s}_l^2(t)}{\hat{b}}\right] \tag{5}$$

where f and l indicate the follower and the leader, respectively, s is the space travelled by a vehicle, and $\dot{s}$ is its speed. $\dot{S}_f$ and $\ddot{S}_f$ are, respectively, the follower's maximum desired speed and maximum acceleration. b_f and $\hat{b}$ are, respectively, the follower's maximum deceleration and the estimate of the leader's maximum deceleration (in the formulation, they are considered with their sign). L is the leader's vehicle length, and θ is a safety margin introduced by Gipps to keep the follower from always decelerating at his/her maximum rate. ΔS_0 is the intervehicle spacing at a stop. α, β, and γ are model parameters that, in the original formulation by Gipps [31], are assumed equal to 2.5, 0.025, and 0.5, respectively. The model in (1) is a delayed differential equation (τ being the delay). In Gipps' original paper, the solution of (1) is made simple by adopting an integration step just equal to the delay τ. A forward Euler method on acceleration (i.e., a trapezoidal integration scheme on speed) is then adopted for calculations. The strategy adopted in AIMSUN for the model's implementation is slightly different. See [27] for further details.

Lane-Changing Model in AIMSUN: The lane-changing model implemented in AIMSUN is an adaptation of the model provided in [32]. Essentially, the decision to change the lane is modeled as a process, analyzing the necessity of the lane change (e.g., to follow the vehicle's determined route), the desirability of the lane change (e.g., to keep the desired speed when the leader vehicle is slower), and the feasibility of the lane change itself (depending on the location of the vehicle in the road network).

Once, for a certain vehicle, the need to change lanes has been ascertained, the model checks whether, in the destination lane, the vehicle will find the desired characteristics (e.g., whether a vehicle wants to change lanes to attain its desired speed, the model checks whether traffic conditions in the destination lane will allow this condition). Again, for further details, see [27].

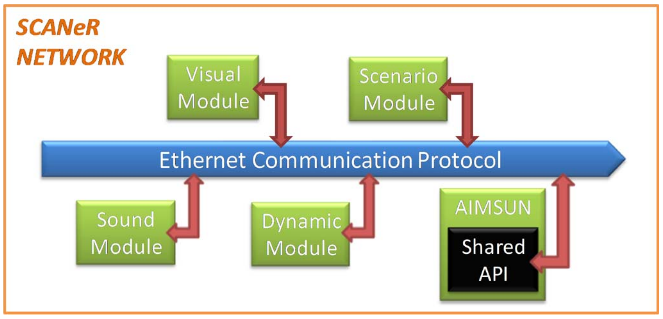


Fig. 3. Schematic of the software architecture.

The need to change lanes usually involves another decision. Indeed, changing lanes usually means using the existing gap between two vehicles on the destination lane. The decision to take is therefore whether this gap is acceptable. The gap-acceptance model is then used to reproduce this additional decision. Further details are, again, available in [27].

B. Software Architecture

Several strategies can be used for implementing the communication framework. However, not all the possible options can be carried out for several reasons. In particular, as already explained, one fundamental requirement is communication speed. A highly time-consuming strategy would cause problems during the simulation in terms of real-time input restitution.

Moreover, the strategy should involve, as far as possible, functions and methods already available in both software applications. Indeed, the use of tools created ad hoc for users by the software developers could nullify, for example, the possibility of upgrading the software.

For such reasons, the approach chosen here was only to make use of standard APIs available in both software applications. One of the main shortcomings of using APIs concerns the speed involved, but in this case, due to their peculiar structure, the problem was overcome, as will be shown in the following discussion, using a shared environment (see Fig. 3).

Shared Environment: The problem of using APIs mainly lies in the way in which data to be exchanged are stored. If an external database is used, the time required would make real-time transmission impossible. The only solution is to use a single environment for both APIs to directly use data retrieved during the simulations. This approach is possible, in this case study, due to their particular structure.

In SCANeR [33] (the software that manages VERA), each model used during simulation is included in a specific module. These modules communicate with each other through messages sent and gathered on the connecting network during the simulation. The API (based on the C++ programming language) may be used to create an additional module (an executable application) that can communicate with the others in the same way [34].

By contrast, in AIMSUN, this modular structure does not exist, and the API (here also based on the C++ programming

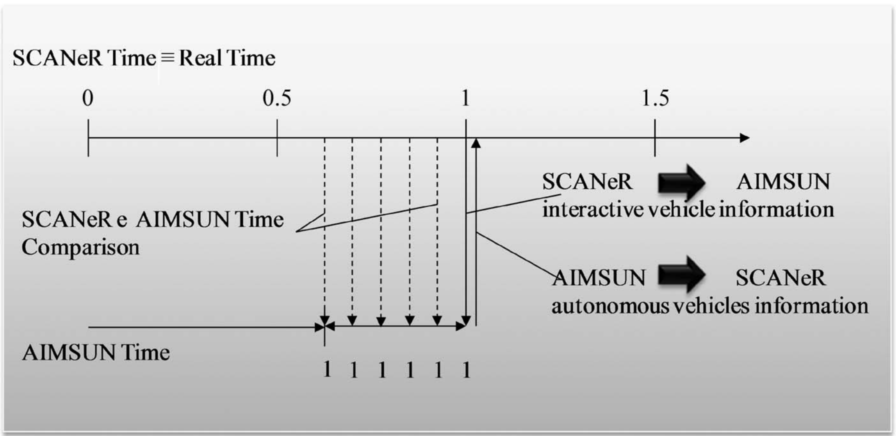

SCANeR Time ≡ Real Time
0
0.5
1
1.5
SCANeR e AIMSUN Time Comparison
SCANeR ➡ AIMSUN interactive vehicle information
AIMSUN ➡ SCANeR autonomous vehicles information
AIMSUN Time
1 1 1 1 1 1

Fig. 4. Time consistency and real-time data exchange. Considering a simulation step of AIMSUN equal to 0.5 s, the code checks (dot arrows) the consistency between the SCANeR time (i.e., real time) and the AIMSUN simulation time until they become equal (in this case, at 1 s). Then, information is exchanged (continuous arrows).

language) consists of a dynamic load library (DLL) in which the user can update the content of six functions that the software calls during its execution [35].

In light of these features, the strategy carried out consisted of using the AIMSUN API as a shared environment, which included all the programming codes. Obviously, it required the inclusion in the DLL of all the libraries needed by SCANeR. Thus, to all intents, AIMSUN itself is a new SCANeR module (see Fig. 3), which is highly suitable from a conceptual point of view.

During its execution, AIMSUN calls the following six functions from the DLL used as an API [35]:

1) AAPILoad();
2) AAPIInit();
3) AAPIManage();
4) AAPIPostManage();
5) AAPIFinish();
6) AAPIUnload().

These functions allow the user to access information with regard to every simulation phase. It is thus possible to properly manage all the simulation phases of both applications. In particular, we have the following five cases.

1) AAPILoad and AAPIUnload are called when the module created by the API is loaded and unloaded.
2) AAPIInit is called just at the beginning of the simulation and is used here for the following reasons:
    a) to load the SCANeR module;
    b) to retrieve the starting position of the interactive vehicle in SCANeR;
    c) to create a tracked vehicle in AIMSUN;
    d) to set its position at the starting position of the interactive vehicle;
    e) to set the starting position of all the autonomous vehicles in SCANeR (outside the network).
3) AAPIManage is called at every AIMSUN simulation step at its beginning, and it is thus used to retrieve the position of the interactive vehicle in SCANeR and to update the position of the tracked vehicle in AIMSUN.
4) AAPIPostManage is called at every AIMSUN simulation step at its end and is thus used to retrieve the position of all AIMSUN vehicles and update the position of all autonomous vehicles in SCANeR.
5) AAPIFinish is called at the end of the simulation and is used to close the SCANeR module.

C. Proposed Solutions to the Integration Issues

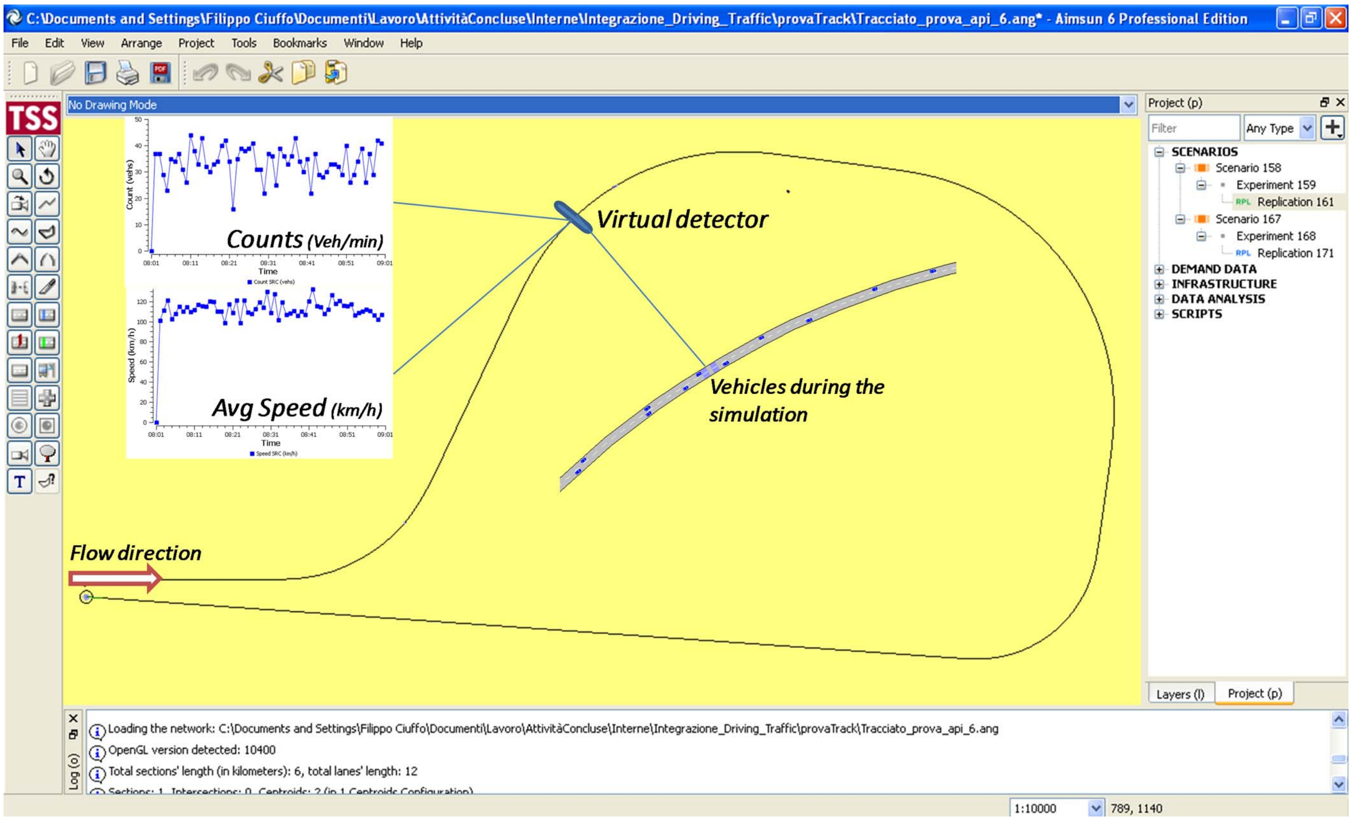
In the following sections, we describe the way in which the aforementioned integration issues were tackled.

Accurate Road Matching Between Traffic and Driving Modules: Both AIMSUN and SCANeR applications can import drawing files (.dwg) or shape files (.shp). For this reason, the consistency between the two reference systems is automatically granted. With regard to the road-matching problem, tools available in AIMSUN for network creation already allow the refinement required, at least for road horizontal alignment. However, currently, with regard to road vertical alignment, it is only possible to set the initial and final altitude of every road section (i.e., it is only possible to set the section slope). Hence, to also achieve high accuracy for the vertical profile of the road, a large number of road segments need to be created, which can be problematic for the AIMSUN lane-changing models and for the time needed to perform a simulation (see [27] for more details).

Real-Time Traffic Simulation: Synchronization of Traffic and Driving Modules: As aforementioned, the main issue is the reduction of the simulation speed of the traffic simulation to have a real-time (and thus consistent) simulation.

This case was obtained by simply synchronizing the simulation clock of AIMSUN and SCANeR. A schematic of the synchronization is reported in Fig. 4. In particular, at each simulation step, AIMSUN compares (dot arrows in Fig. 4) the time of its simulation clock with the SCANeR simulation time (i.e., real time). As long as the AIMSUN simulation time is less than SCANeR, data exchange is not allowed for (in Fig. 4, it is allowed only at 1 s when the two times become equal). This procedure ensures the temporal consistency of the information exchanged between the two applications during the simulation.

In addition, the problem of an efficient data exchange is overcome through the shared software environment performed at the level of environment variables.



```
Loading the network: C:\Documents and Settings\Filippo Ciuffo\Documenti\Lavoro\AttivitàConcluse\Interne\Integrazione_Driving_Traffic\provaTrack\Tracciato_prova_api_6.ang
OpenGL version detected: 10400
Total sections' length (in kilometers): 6, total lanes' length: 12
Sections: 1, Intersections: 0, Centroids: 2 (in 1 Centroids Configuration)
```

Fig. 5. Road network used in the preliminary evaluations and the traffic conditions that result from a simulation with AIMSUN.

Consistency of the Updating Frequency of Calculations: As for most traffic-simulation software, the smallest possible simulation step in AIMSUN is 0.1 s. In SCANeR, the situation is more complex. As already explained, the simulation software is made up of different submodules, each working at its own frequency. Communication among the different modules can be achieved in an asynchronous or in a pseudosynchronous way. In both ways, the SCANeR process runs at its own frequency, but in the asynchronous strategy, it always gets the most recent values for data, whereas in the pseudosynchronous strategy, it can get data values only when they are explicitly sent by each module. The pseudosynchronous strategy allows AIMSUN to send data to SCANeR just after each calculation, regardless of the specific instant at which this case happens (as schematically represented in Fig. 4).

Unfortunately, updating autonomous positions at a 10-Hz frequency will result in their noncorrect visual restitution in the driving simulation environment. For this reason, between two successive simulation steps, the position of each autonomous vehicle in the surroundings of the interactive one is interpolated, keeping the acceleration constant and, thus, deriving a new speed and position. This way, vehicles are updated 30 times/s, giving realistic visualization. The hypothesis of constant acceleration within the 0.1-s time intervals was considered acceptable, given the usual human–machine response frequency.

Management of Autonomous Vehicle Visualization: This issue, which, in principle, is not particularly difficult to tackle, has instead posed serious questions on the possibility of performing information exchange in real time.

Indeed, in the API, and in particular within the function AAPIManage(), the code detects the vehicles that are within a certain distance from the interactive vehicle by evaluating the Euclidean distance for all the vehicles on the network. This case is a very time-consuming way of operating and compromises the use of the integrated framework for large networks. A more efficient way would be to identify the road sections within a certain distance from the interactive vehicle and then consider all the vehicles on those road sections. Unfortunately, the AIMSUN APIs currently do not provide the so-called *forward* and *backward section stars*, not allowing such implementation.

The one-to-one correspondence among vehicles in both simulations is thus obtained by preserving their memory between two successive time steps. Then, for each vehicle in the surroundings of the interactive vehicle, it is checked whether it already existed in the previous simulation step. In case of existence, the vehicle in the traffic simulation is assigned to the same vehicle (i.e., with the same visual features) in the driving simulation; otherwise, a new vehicle in the driving simulation is assigned to it. This approach is also done in the AAPIManage() function of the API. This step is rather time consuming and may lead to problems for the real-time exchange of information.

D. Preliminary Tests

The first tests were carried out on a 6.5-km two-lane French road (i.e., constant altitude). SCANeR 2.18 and AIMSUN 6 releases were used in the integration framework. The traffic flow considered in the simulation consisted of 2000 veh/h. The

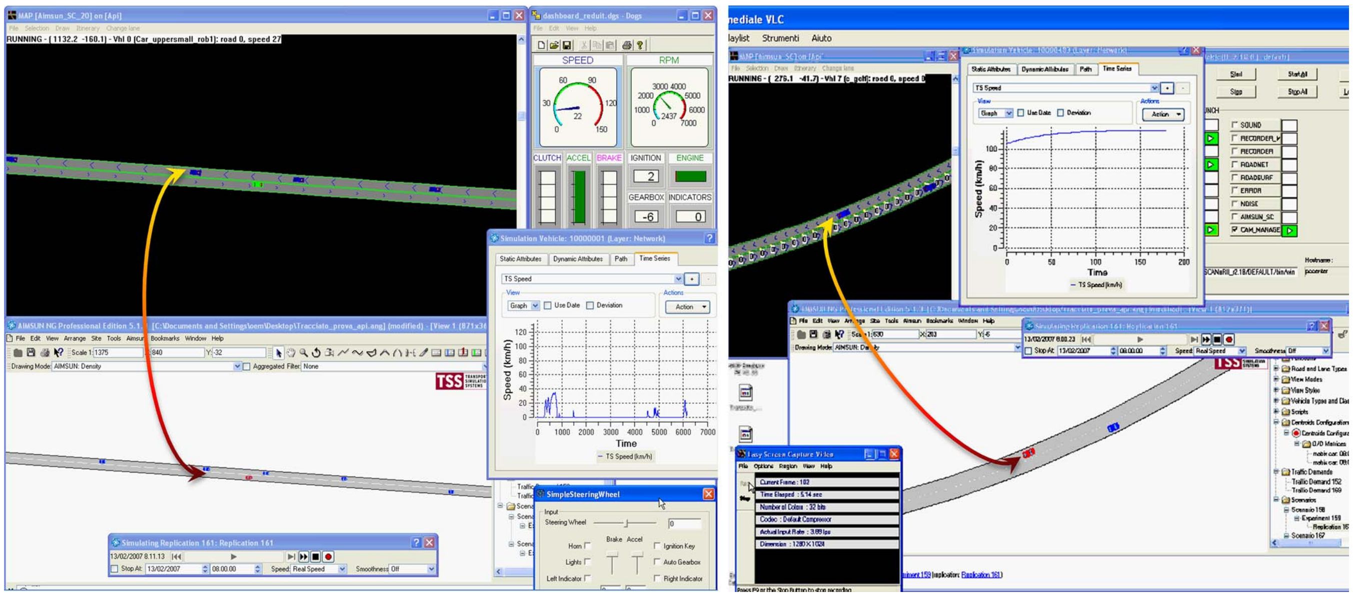


Fig. 6. Vehicle correspondence in the two simulation environments.

network layout and traffic conditions simulated with AIMSUN are reported in Fig. 5. Four bends (minimum radius of 350 m) are present in the layout, whereas traffic conditions are gathered through a virtual detector. In particular, just to have an idea of the traffic conditions simulated and of their variability in space and time, the detector (see Fig. 5) recorded during the simulation an average count of 13 veh/min (with a standard deviation of 3 veh/min) and an average speed of 97 km/h (with a standard deviation of 14 km/h) in the rightmost lane, whereas it recorded an average count of 20 veh/min (standard deviation of 5 veh/min) and an average speed of 120 km/h (standard deviation of 18 km/h) for the leftmost lane.

In the integrated framework, information was exchanged only for the vehicles within a distance of 200 m from the interactive vehicle (150 m in front and 50 m behind it). After several tests, the maximum number of vehicles allowed for in the SCANeR simulation was set at 50: higher numbers of vehicles slowed down the traffic simulation almost to real time. However, in the particular scenario tested, 35 vehicles were sufficient to comprehensively simulate the traffic surrounding the interactive vehicle, and data exchange proved consistent both in time and space. Fig. 6 gives a snapshot of the two synchronized networks and shows how the combination of output visualization tools from both software applications allows users to thoroughly analyze the simulation during its execution, e.g., giving in real time the speed profile of each single vehicle simulated in the driving/traffic scenario, as well as traffic conditions at any point in the network.

V. Further Comments

Additional comments can be provided to further support the idea behind the presented integration. In AIMSUN 6, a vehicle can deal with "nonvalid" positions (the interactive vehicle can also move outside the network, e.g., on the side walk), but they are not modeled by AIMSUN (if the interactive vehicle stops in the middle of the lane, the other vehicles also stop, without considering overtaking it also outside the network, or even worse, if the interactive vehicle stops in the middle of two lanes, vehicles in both lanes also stop). In addition, vehicle kinematics in curve sections and turnings has to be improved, as well as the lane-changing movements, e.g., by rotating overtaking vehicles according to their speed. These very detailed aspects of the simulation can greatly benefit from the use of the integrated environment and the detailed information on driving behavior that can be achieved from it.

The same case holds for the lane-changing simulation in two-lane rural highways. Indeed, even the few models proposed so far (e.g., [36]) use an approach that is in line with common lane-changing models. Instead, the logic behind the simulation of this kind of dynamic needs changing.

Finally, the results of this paper led to further developments carried out by the software houses, where the adopted framework is slightly different: All the vehicles are simulated by the traffic-microsimulator model until they enter the surroundings of the interactive vehicle. From that point on, the vehicles are simulated by the traffic module of the driving simulator. This framework was a way of overcoming the inconsistency of vehicle movements entering and exiting from the surroundings of the interactive vehicle and the reduced accuracy of the traffic simulator in representing vehicle movements. Furthermore, from a research perspective, this framework does not allow new driver models implemented in the traffic model to be tested in the driving simulator environment, which was one of the main motivations for the work presented in this paper.

VI. Conclusion

This paper has described the methodological and technical issues of integrating a driving simulation engine and a traffic-flow microsimulation model. It has also presented preliminary results of the tests carried out on the integration prototype.

Synchronization of simulations both in time and space, and real-time data exchange were the main challenges in designing and realizing the integration. However, these issues were overcome through APIs alone, without resorting to native codes. Major improvements in the visual restitution of vehicles are expected with the new AIMSUN versions, e.g., by introducing a variable for lateral acceleration and a deviation angle with respect to road alignment during a lane-changing maneuver. Given the flexibility of the combined environment, new research perspectives are expected both in driving and traffic simulation.

Acknowledgment

The authors would like to thank A. Troale of Oktal for his valuable support for this paper.

References

[1] L. Yang, J. H. Yang, E. Feron, and V. Kulkarni, "Development of a performance-based approach for a rear-end collision warning and avoidance system for automobiles," in *Proc. IEEE Intell. Vehicles Symp*, 2003, pp. 316–321.

[2] C.-Y. Chang and Y.-R. Chou, "Development of fuzzy-based bus rear-end collision warning thresholds using a driving simulator," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 360–365, Jun. 2009.

[3] F. Galante, F. Mauriello, A. Montella, M. Pernetti, M. Aria, and A. D'Ambrosio, "Traffic calming along rural highways crossing small urban communities: Driving simulator experiment," *Accid. Anal. Prev.*, vol. 42, no. 6, pp. 1585–1594, Nov. 2010.

[4] F. Yang, K. Zhang, and X. Sun, "Behavior analysis of traffic accidents with high-fidelity driving simulator," in *Proc. 2nd ICTE 345*, 2009, pp. 3231–3235.

[5] Y. Suda, Y. Takahashi, M. Kuwahara, S. Tanaka, K. Ikeuchi, M. Kagesawa, T. Shraishi, M. Onuki, K. Honda, and M. Kano, "Development of universal driving simulator with interactive traffic environment," in *Proc. IEEE Intell. Vehicles Symp.*, 2005, pp. 743–746.

[6] S. Espié, "ARCHISIM: Multiactor parallel architecture for traffic simulation," in *Proc. 2nd World Congr. Intell. Transp. Syst.*, Yokohama, Japan, 1995.

[7] S. El Hadouaj and S. Espié, "A generic road traffic simulation model," in *Proc. ICTTS (Traffic and Transportation Studies)*, Guilin, China, 2002.

[8] A. Champion, M.-Y. Zhang, J.-M. Auberlet, and S. Espié, "Behavioral simulation of a high-density traffic network involving an adaptive ramp metering system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2002, vol. 5, pp. 418–423.

[9] O. Ahmad and Y. Papelis, "A comprehensive microscopic autonomous driver model for use in high-fidelity driving simulation environments," in *Proc. 81st TRB Annu. Meeting*, Washington, DC, 2002.

[10] Y. Papelis, O. Ahmad, and H. German, "Adaptive controllers for vehicle velocity control for microscopic traffic simulation models," *Adv. Transp.*, vol. 14, pp. 151–160, 2003.

[11] S. Wright, "Supporting intelligent traffic in the Leeds driving simulator," Ph.D. dissertation, School Comput., Univ. Leeds, Leeds, U.K., 2000.

[12] S. Wright, N. J. Ward, and A. G. Cohn, "Enhanced presence in driving simulators using autonomous traffic with virtual personalities," *Presence: Teleoperators Virtual Environ.*, vol. 11, no. 6, pp. 578–590, Dec. 2002.

[13] J. M. Jenkins, "Integrating driving simulation and traffic simulation: A case study," in *Proc. Annu. Conf. Can. Soc. Civil Eng.*, 2005, pp. TR-168-1–TR-168-9.

[14] J. J. Olstam, J. Lundgren, M. Adlers, and P. Mastoms, "A framework for simulation of surrounding vehicles in driving simulators," *ACM Trans. Model. Comput. Simul.*, vol. 18, no. 3, 2008, n. 9.

[15] M. Treiber, A. Kesting, and D. Helbing, "Three-phase traffic theory and two-phase models with a fundamental diagram in the light of empirical stylized facts," *Transp. Res. Part B*, vol. 44, no. 8/9, pp. 983–1000, Sep.–Nov. 2010.

[16] V. Punzo and A. Tripodi, "Steady-state solutions and multiclass calibration of Gipps' microscopic traffic flow model," *Transp. Res. Rec.*, no. 1999, pp. 104–114, 2007.

[17] T. Toledo, M. Ben-Akiva, D. Darda, M. Jha, and H. Koutsopoulos, "Calibration of microscopic traffic simulation models with aggregate data," *Transp. Res. Rec.*, no. 1876, pp. 10–19, 2004.

[18] B. Ciuffo, V. Punso, and V. Torrieri, "Comparison of simulation-based and model-based calibration of traffic flow microsimulation models," *Transp. Res. Rec.*, no. 2088, pp. 36–44, 2008.

[19] V. Punzo and F. Simonelli, "Analysis and comparison of microscopic traffic flow models with real-traffic microscopic data," *Transp. Res. Rec.*, no. 1934, pp. 53–63, 2005.

[20] J. Maroto, E. Delso, M. J. Cabanellas, and J. Félez, "A microscopic model of urban traffic for implementation in driving simulators," *Proc. Inst. Mech. Eng.—Part D: J. Automobile Eng.*, vol. 220, no. 10, pp. 1345–1361, 2006.

[21] J. Félez, J. Maroto, G. Romero, and M. J. Cabanellas, "A full driving simulator of urban traffic including traffic accidents," *Simulation*, vol. 83, no. 5, pp. 415–431, May 2007.

[22] M. Jin and S.-H. Lam, "A virtual-reality based integrated driving-traffic simulation system to study the impacts of intelligent transportation systems (ITS)," in *Proc. Int. Conf. Cyberworlds*, Nanyang, Singapore, 2003, pp. 158–165.

[23] I. Vladisavljevic, J. M. Cooper, P. T. Martin, and D. L. Strayer, "The importance of integrating driving and traffic simulations, illustrated through a case study that examines the impact of cell phone drivers on traffic flow," in *Proc. 88th TRB Annu. Meeting*, Washington, DC, 2009.

[24] Paramics Version 5.0 User Guides, Quadstone Ltd., Edinburgh, U.K., 2004.

[25] PTV—Planning Transport and Vehicles: VISSIM, 2000. [Online]. Available: www.ptv.de

[26] MULTITUDE: Methods and tool for supporting the use, calibration and validation of traffic simulations models—COST Action TU0903, [Online]. Available: www.multitude-project.eu

[27] AIMSUN 6. Microsimulator and Mesosimulator in AIMSUN 6. User's Manual. Transport Simul. Syst. 1997–2008, Transp. Simul. Syst., Barcelona, Spain, Jan. 2009.

[28] Last accessed Oct. 19, 2009, TEST scarl, Centro Regionale di Competenza 'Trasporti.' [Online]. Available: www.crdctest.it/on-line/en/Home.html

[29] J. J. Olstam and S. Espié, "Combination of autonomous and controlled vehicles in driving simulator scenarios," in *Proc. Int. Conf. RSS*, Rome, Italy, 2007.

[30] Legion Pedestrian Simulator. [Online]. Available: http://www.legion.com/

[31] P. G. Gipps, "A behavioral car-following model for computer simulation," *Transp. Res. Part B*, vol. 15, no. 2, pp. 105–111, Apr. 1981.

[32] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transp. Res. Part B*, vol. 20, no. 5, pp. 403–414, Oct. 1986.

[33] SCANeR II User's Guide v2.18a, Oktal, Paris, France, Nov. 28, 2006.

[34] SCANeR II API Programming Guide v2.18a, Oktal, Paris, France, Nov. 28, 2006.

[35] AIMSUN 6. Aimsun Microsimulator API Manual. Transport Simulation Systems 2006–2009, Transp. Simul. Syst., Barcelona, Spain, Jan. 2009.

[36] A. Tapani, "A versatile model for rural road traffic simulation," in *Proc. 84th TRB Annu. Meeting*, Washington, DC, 2005.

Vincenzo Punzo received the M.Sc. (first-class honors) degree in civil engineering in 1998 and the Ph.D. degree in transportation engineering from the University of Naples Federico II, Napoli, Italy, in 2002.

From 2003 to 2004, he was a Postdoctoral Fellow, and since 2005, he has been Assistant Professor with the Department of Transportation Engineering, University of Naples "Federico II," holding the course of "Traffic flow simulation and control." One of his main research topics has been microscopic simulation of traffic flow, covering several aspects ranging from estimation techniques for trajectory data to calibration methods, model development, integration of pollutant emission, and traffic flow models. His other major research interests include driving simulation and active road safety. Since 2003, he has taken part in the development of the dynamic driving simulator VERA at the Road Safety Laboratory, Technology for Transport, Environment and Safety (T.E.S.T. S.c.a.r.l.), Naples. He is the primary author of more than 30 papers in peer-reviewed journals and conference proceedings and a Reviewer for several journals such as *Transportation Research C, Transportation Research Record*, and *European Journal of Operational Research*.

Dr. Punzo is Member of the Traffic Flow Theory and Characteristics Committee, Transportation Research Board, National Academy of Sciences. He is an Associate Editor for the 13th International IEEE Conference on Intelligent Transportation Systems and the Chairman of the COST Action TU0903 MULTITUDE "Methods and tool for supporting the use, calibration, and validation of traffic simulations models."

Biagio Ciuffo received the M.Sc. (first-class honors) degree in engineering for environment and land use from the University of Naples Federico II, Napoli, Italy, in 2004 and the Ph.D. degree in hydraulic engineering, land use, and transportation networks from the Department of Transportation Engineering, University of Naples "Federico II," in 2008.

He is currently with the Institute for Environment and Sustainability, Joint Research Centre, European Commission, Ispra, Italy. He is the author of more than 20 papers in peer-reviewed journals and conference proceedings in transportation engineering and is a Reviewer for journals such as *Energy Policy* and *Transportation Research Record*. He is currently part of the COST Action MULTITUDE "Methods and tool for supporting the use, calibration and validation of traffic simulations models," which will be completed in 2013.

Dr. Ciuffo is a Friend Member of the Traffic Flow Theory and Characteristics Committee, Transportation Research Board, National Academy of Sciences. He received two national prizes from the Italian Society of Professors in Transportation Engineering and the Italian Society of Transportation Engineering for his Ph.D. dissertation on the integration of a microscopic traffic simulator within a driving simulator.