A Large-Scale SUMO-Based Emulation Platform

Wynita M. Griggs, Rodrigo H. Ordonez-Hurtado, Emanuele Crisostomi, Florian Haeusler, Kay Massow, Robert N. Shorten

Published in *IEEE Transactions on Intelligent Transportation systems* (25 May 2015)

DOI: 10.1109/TITS.2015.2426056

A Large-Scale SUMO-Based Emulation Platform

Wynita M. Griggs, *Member, IEEE,* Rodrigo H. Ordóñez-Hurtado, Emanuele Crisostomi, *Member, IEEE,*
Florian Häusler, Kay Massow, and Robert N. Shorten, *Senior Member, IEEE*

***Abstract*—A hardware-in-the-loop simulation platform for emulating large-scale intelligent transportation systems is presented. The platform embeds a real vehicle into SUMO, a microscopic road traffic simulation package. Emulations, consisting of the real vehicle, and potentially thousands of simulated vehicles, are run in real time. The platform provides an opportunity for real drivers to gain a feel of being in a large-scale, connected vehicle scenario. Various applications of the platform are presented.**

I. Introduction

The inadequacy of many of our current transport systems, when it comes to coping with factors such as rising volumes of traffic and environmental pollution, together with rapid advancements in information and communications technology (ICT), has sparked much investment in developing smarter transportation systems. Projects such as EcoGem [1] (Cooperative Advanced Driver Assistance System for Green Cars), funded in the context of the European Green Cars Initiative [2], aim to integrate technology and intelligence into vehicles and the physical transportation infrastructure in order to provide increased efficiency, safety, and capacity on our roads, and improve the general road travelling experience.

The question is how to best test and evaluate intelligent transportation systems (ITS) when they are in the experimental stage? This often proves to be challenging given that intelligent transportation systems are frequently intended to eventually be deployed in large urban areas and major cities. Access to fleets of thousands of vehicles equipped with the prototype technologies and communications abilities necessary for testing is usually impractical. There also exists the contemporary consequence of the rapid development and deployment of ITS, in that the associated experience of being in a connected vehicle scenario for many drivers will be brand new. Simulators may be used to emulate large scale, but cannot accommodate for all of the complexities, uncertainties, technical issues, and driver attitudes and responses that might arise in the real world [3]. On the other hand, small, real-world test fleets of 1–20 vehicles demonstrate proof of concept, but cannot accurately predict the outcomes of ITS in the context of much larger fleet sizes and city-wide scenarios.

In this paper, we consider the idea of merging large-scale traffic simulation and the proof of concept capability provided by real-world vehicles. In particular, we present a prototype platform for embedding, in real time, real vehicles into SUMO (Simulation of Urban MObility), an open source, microscopic road traffic simulation package primarily being developed at the Institute of Transportation Systems at the German Aerospace Centre (DLR) [4].

The main contribution of this paper is to describe a platform to evaluate experimental ITS that we have realised, that provides the drivers of real vehicles the opportunity to somewhat experience what it would feel like to be part of a large-scale connected vehicle scenario, where most of traffic in the scenario is simulated by SUMO in order to avoid the necessity of having large, real vehicle, test fleets. Specifically:

(i) our hardware-in-the-loop simulation platform permits real human driver responses to the advice and directions given by the simulator, to be observed and fed back to the simulator, in real time, so that the simulator may process human inputs otherwise possibly not predicted;

(ii) factors, such as how often a driver would like to receive behavioural recommendations, or how likely a driver is to obey a series of recommendations, or how comfortable a driver is with a series of recommendations, can be explored;

(iii) our platform is able to incorporate an element of safety into testing with real drivers, in that field demonstrations can be performed on empty roads or test tracks while scenarios such as wide-spread traffic congestion, busy intersections or accidents are simulated;

(iv) provided that a real vehicle is available, our platform is relatively inexpensive to construct (as opposed to purchasing or building and embedding a driving simulator into SUMO).

The following list outlines a number of ITS applications that we envisage could benefit from demonstration using our platform. The list is, of course, by no means exhaustive, and there are certainly scenarios in which testing with the platform does not make sense. However, our focus is on applications in which there is feedback between the environment and the vehicle. It is this closed-loop setting that is somewhat novel in the context of this paper.

(a) Intelligent Speed Advisory Systems: these include systems that attempt to detect approaching traffic bottlenecks, such as roadwork zones or traffic jams, and provide drivers with resultant updates on recommended travelling speeds; for instance, see [5]. The goal is to reduce road chaos; ensure that vehicles travel at safe speeds and at safe distances from the vehicles ahead of them as they approach, enter and leave the bottleneck; and to increase the overall throughput of traffic

This work was supported in part by Science Foundation Ireland grant 11/PI/1177.

Parts of this work were presented at the 2nd International Conference on Connected Vehicles and Expo in Las Vegas, Nevada, USA, 2013. Some parts have been accepted for presentation at the 3rd International Conference on Connected Vehicles and Expo in Vienna, Austria, 2014.

W. Griggs, R. Ordóñez-Hurtado and R. Shorten are with the Hamilton Institute, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland. Corresponding author: Wynita M. Griggs. Phone: +353-(0)1-7086100. Fax: +353-(0)1-7086269. Email: wynita.griggs@nuim.ie

E. Crisostomi is with the Department of Energy, Systems, Territory and Constructions Engineering (DESTEC), University of Pisa, L.go L. Lazzarino, Pisa, Italy.

F. Häusler and K. Massow are with Fraunhofer FOKUS, Berlin, Germany.

R. Shorten is also with IBM Research Ireland, Dublin, Ireland.

in the congested zone. One benefit of testing these kinds of systems with our proposed platform is that a traffic bottleneck can be emulated on demand in the simulator while proof of concept of the system is being demonstrated in a real vehicle. Driver comfort and compliancy with the frequency of incoming speed recommendations can also be assessed.

(b) Local Obstacle Avoidance via Rerouting: the objective here is to intelligently reroute traffic around a local obstacle, e.g. a traffic accident or congestion [6]. Again, a benefit of testing an intelligent system of this kind with our platform is that a road obstacle can be emulated on demand in the simulator while proof of concept of the system and driver reaction is being demonstrated in a real vehicle. Furthermore, real-life situations can arise where a driver may visually not see an oncoming obstacle because it is, for example, around a corner or too far ahead. Yet, at this very early stage of approaching an obstacle, the driver may be expected to reroute. This situation is mirrored with the platform, in that both the real life obstacle, in this case, and a simulated one, is not seen by the driver. The platform allows for the assessment of driver comfort with these very early, unexpected reroutes.

(c) Pollution Control: systems such as twinLIN [7] offer advanced vehicle engine control based on environmental information. Our platform allows one to simulate regional vehicle emissions and air quality while, through the use of additional devices, such as the automated mechanical switch described in [7], imposing mandatory changes to the engine mode of a real vehicle while a driver travels. The impact on the driver of these mandatory engine mode changes can be examined.

(d) Parking: various solutions to the challenges associated with finding a parking spot in densely populated areas have been proposed, which include pricing mechanisms and online reservation systems. Stochastic policies that balance parking loads while aiming to avoid “flapping” were introduced in [8]. “Flapping” is the oscillatory behaviour observed when drivers attempt to go to a car park that they have been informed is empty, only to arrive and find that the car park has filled, given that other drivers received similar information, and thus had also attempted to park. In [8], vehicles make a decision on whether to go to a car park depending on the probability (broadcasted by the ITS) of there being an available space to park in when they arrive. Driver behaviour, in general, with respect to parking choice is typically complicated. Our platform provides a benefit of incorporating the behaviour of real drivers attempting to park into an emulation.

(e) Cooperative Intersection Collision Avoidance Systems: a goal of these systems is to provide information to drivers performing maneuvers through intersections about the state of activities within that intersection. This information might concern gaps in oncoming traffic, or when other users, such as pedestrians or bicyclists, pose hazards. With our platform, user acceptance of such systems can be evaluated in a real vehicle under safe testing conditions, where other vehicles at the intersection are simulated, and the success rates of accident avoidance in the emulation explored.

The remainder of our paper is sectioned as follows. In Section II, we explore some of the state of the art in relation to ITS simulation. In Section III, we describe the key components of our hardware-in-the-loop simulation platform. In Section IV, we begin to investigate the potential of the platform in the context of a number of ITS applications. Conclusions and future objectives are outlined in Section V.

II. State of the Art

Before proceeding with a component-wise description of our platform, it is worth taking a brief look at some of the state of the art in ITS simulation. Simulators have been widely used in the development of intelligent transportation systems, to perform safe experiments in a controlled environment, or to predict the behaviour of new devices or new traffic strategies that involve thousands of vehicles. More recently, simulators have been used to develop and test cooperative driving behaviours as well.

Three main categories of simulators can be identified: traffic simulators, driving simulators, and networking simulators. Traffic simulators are typically used to develop strategies to improve the mobility and safety of urban and rural travel; driving simulators are convenient to test human responses to new applications in a controlled environment; and finally, networking simulators are used to evaluate the ability of Vehicular Ad-hoc NETworks (VANETs) to exchange information in a challenging environment, like the urban one, in order to implement new cooperative strategies (e.g. cooperative routing, safety alerts, and congestion messages).

In this framework, recent years have witnessed a growing interest in developing more realistic environments to test and validate ITS applications. For instance, [9] illustrates preliminary benefits in integrating driving and traffic simulators. Clearly, mobility simulators would benefit from realistically taking into account drivers’ behaviours, which is considered in driving simulators, while driving simulators would provide an improved experience if implemented in a reasonably believable traffic environment. Similarly, [10] describes efforts aiming to develop a virtual IntelliDrive testbed within a microscopic traffic simulator. IntelliDrive is a kind of vehicle-to-infrastructure integrated platform that uses advanced wireless technologies to implement applications such as “systems warning drivers of traffic slowdowns ahead, systems warning about cross-street vehicles that may potentially run through a red light, and systems notifying drivers of roadway features, such as sharp curves”. Reference [11] describes the modelling and the implementation of an architecture to integrate a robotics and a traffic simulator to facilitate the study of the driving behaviour of autonomous vehicles in human steered urban traffic scenarios. The integration of more simulators was also investigated in [12], where the Simulink/MatLab model of an electric bus powertrain subsystem was exported into a mobility simulator. Finally, similar concepts are further extended in [13] where traffic, driving and networking simulators are all integrated in one only research and development tool, which is used to implement new applications for connected vehicles. See, also, [14].

Other works in the literature have tried to increase the level of realism of current simulators by considering some real vehicles travelling in the road network as well. In this context, we note [15]. In this work, the objective was to implement a mixed reality platform, where intersection control policies for autonomous vehicles (formerly only tested in simulation) were tested with a real autonomous vehicle, interacting with multiple virtual vehicles (in a simulation), at a real intersection, in real time. In such an example, having all real vehicles would have been expensive in case some control policies failed. At the same time, the experiment proved to provide results that were different from those obtained using a fully simulated environment. Similarly, [16] describes GrooveNet, which is a hybrid vehicle-to-vehicle network simulator, capable of communication between simulated vehicles, real vehicles, and between real and simulated vehicles. With such an approach it becomes feasible to deploy a small fleet of vehicles (in the example, in the order of a dozen), to test protocols that in truth involve hundreds or thousands of vehicles, which are simulated. The hardware-in-the-loop approach of [16] is close to what we propose here, though GrooveNet is designed with others objectives in mind. We are interested in providing a platform in which ITS applications can be investigated and experienced by a human driver. In particular, we are most interested in investigating closed-loop applications (involving feedback). GrooveNet is designed to investigate V2V issues; in particular, with respect to wireless communication issues in mobility networks. This latter issue is of no interest in our context. Furthermore, while GrooveNet does include some simple car-following, traffic lights, lane changing and simulated GPS models to simulate traffic, it is not as realistic as a more comprehensive mobility simulator, like the one that we used in our approach (i.e. SUMO), nor is it designed to investigate ITS applications. As a consequence, the applications described in the following sections, which are in essence closed-loop control applications, can be investigated with a greater level of ITS realism using the SUMO-based platform.

III. Description of the Platform

We now describe the main components of our proposed hardware-in-the-loop simulation platform. A general overview of the architecture of the platform is shown in Fig. 1. The main objective of the platform architecture is to provide the possibility for feedback between simulated vehicles and the human driver in the real vehicle. The main components are the applications deployed on the base station computer interfacing hundreds of vehicles in SUMO, and the SumoEmbed component, deployed on the smartphone, which serves as the interface to the driver and the vehicle.

a) Large-Scale Traffic Simulator: SUMO [4] is an open source, microscopic road traffic simulation package primarily being developed at the Institute of Transportation Systems at the German Aerospace Centre (DLR). SUMO is designed to handle large road networks, and comes with a "remote control" interface, TraCI (short for Traffic Control Interface) [17], that allows one to adapt the simulation and to control singular vehicles on the fly. In our setup, SUMO provides inputs for the applications, e.g. pollution levels or traffic flows on edges of the network.

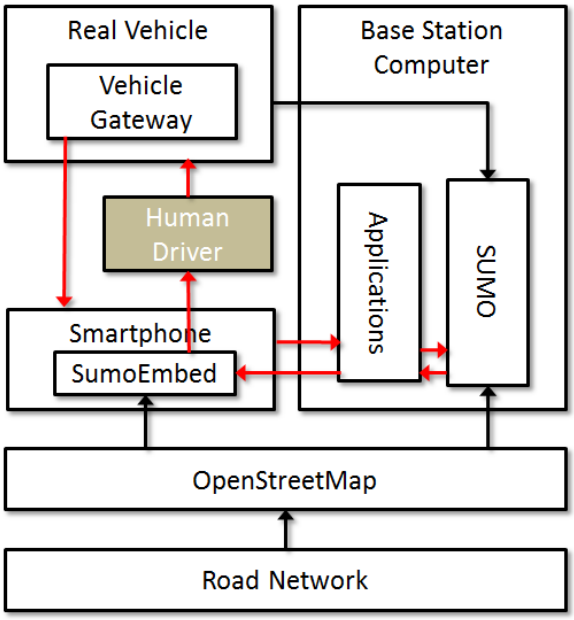


Fig. 1. Main components of the platform. The red arrows highlight the dynamic connections (i.e. the feedback-loop) between the components.

Fig. 2. Field-test vehicle: 2008 Toyota Prius.

b) Real Vehicle: our field-test vehicle is a 2008 Toyota Prius 1.5 5DR Hybrid Synergy Drive and is pictured in Fig. 2. Any vehicle with an accessible interface or gateway (such as an OBD-II diagnostic connector, in our case) is suitable to use as a field-test vehicle. The specifics of the ITS application being tested may place further restrictions on vehicle choice. The road network that we generated in SUMO, and the road network that we drove the real vehicle around on, were topographically the same. A virtual vehicle representing the Toyota Prius (i.e. an avatar of the real vehicle) was created in SUMO. This was partially achieved by assigning physical characteristics to the virtual vehicle that were approximately the same as those of the real car. Further details on defining vehicle types and routes in SUMO are available in the user documentation found on the SUMO website [18].

c) Road Network: various ways of generating a road network exist in SUMO. For example, they can be defined by the user in XML; abstract road networks can be generated using the SUMO application *netgenerate*; or they can be imported from different formats. For our road network, we imported maps of the local Maynooth area (Maynooth, Co. Kildare, Ireland) from OpenStreetMap. In particular, we downloaded and edited the maps using JOSM (Java OpenStreetMap Editor) [19] and cleaned them with XMLStarlet [20] before applying SUMO's *netconvert*.

d) Smartphone: in the real vehicle, we mounted a Samsung Galaxy S III mini (model no. GT-I8190N) running the Android Jelly Bean operating system (version 4.1.2). The purpose of the smartphone is to relay, over a cellular network, periodic information from the vehicle's onboard computer (e.g. the speed of the vehicle) to the base station computer running SUMO, and to receive messages from the base station computer and display them on the smartphone user interface for the driver (e.g. recommendations for alternative driving behaviours). We utilised the mobile data services of a commercial mobile phone operator, where these services were provided using a 3G UMTS 900/2100 network, for the relay of data. The plug-in application that we developed for the smartphone is called SumoEmbed and is described further below.

e) Python Script: we additionally developed application-specific scripts in Python 2.7.3 that were run simultaneously with SUMO on the base station computer. The scripts typically consist of two parts: (i) a main part that acts as a client to SUMO and adapts the traffic scenario simulations, according to the ITS application being implemented, online via TraCI (TraCI uses a TCP-based client/server architecture to provide access to SUMO); and (ii) a second part that acts as a TCP server, listening for incoming calls from the smartphone and then handling the data transfer between the smartphone and the base station computer running SUMO, on the base station computer's end.

f) Vehicle Gateway: the hardware device that we used to connect the smartphone and the Toyota Prius' onboard computer was the Kiwi Bluetooth OBD-II Adaptor by PLX Devices.[1] The device plugs into the vehicle's OBD-II diagnostic connector and communicates to the smartphone via Bluetooth. A variety of existing smartphone applications are compatible for use with Kiwi Bluetooth. We decided upon Torque Pro[2] given that an Android Interface Definition Language (AIDL) application programming interface (API) is included for third party plug-in applications, which we utilised, as follows.

g) SumoEmbed: SumoEmbed is the name of a plug-in application that we designed for Torque Pro.

Before proceeding, some comments are appropriate.

Comment (scalability): In our prototype design, a single, real vehicle was embedded into SUMO. An obvious question concerns the logistics of embedding further vehicles. In the parent process of our Python script, Traci provides access to SUMO via a single port, and multiple real vehicles can be represented in SUMO by different vehicle IDs. Passing or sharing the information of multiple vehicles between this parent process of the Python script, and our Python subprocess that communicates with the smartphone, may require the use of, for example, advanced shared memory formats in Python. A single port was reserved for communication with the smartphone in our Python subprocess; handling multiple incoming calls (i.e. from multiple smartphones) is a server programming exercise. Dealing with multiple incoming calls then leads to the question of data synchronisation. Indeed, synchronisation issues would need further investigation for applications sensitive to, for example, communication delays between the smartphones and the base station computer, and is an issue that we would like to examine in the future. However, in principle, embedding multiple vehicles is indeed possible.

Comment (information exchange): The specific information exchanged between the real vehicle and the simulator is application-driven and, in our current set-up, depends on what data can be sent via the vehicle gateway (e.g. OBD-II diagnostic connector) and/or via the smartphone (e.g. human input through a user interface), and whether this information can concurrently be managed in SUMO. In principle, any information on the OBD-II can be used in the emulation set-up.

Comment (transmission frequency): Information between the real vehicle and the base station computer, in our current set-up, is transmitted once every second. This steady, periodic rate is maintained throughout the duration of the simulation, for simplicity. Changing the information transmission frequency, e.g. to event-driven information transmission, is something that makes sense for a number of applications, and would be worth implementing in the future.

Comment (vehicle-to-vehicle communication): The integration of our platform with a network simulator (e.g. ns-3) to incorporate vehicle-to-vehicle communication into our simulations is an enhancement that we would like to make in a future prototype design. It is not currently included.

IV. Applications

For the remainder of the paper, we demonstrate some ITS applications for which our platform may be of use.

Comment: It should be clearly noted that our platform is not suitable for all ITS applications and we make no claim that it is. For example, in applications where the driver's visual expectation reflects the ITS scenario, our platform cannot be used to give meaningful results. Obstacle avoidance would be one such example. However, there are many examples which do not fall into this category. Some of these are reported in this section.

A. Speed Advisory System

For our first application, we demonstrated the intelligent speed advisory system presented in [5], which provides recommendations to drivers regarding their cruise speed and safe travelling distance with respect to a certain point of interest (POI). Two main stages in the methodology of the system can be distinguished, which are briefly described as follows (for further details, see [5], [21]).

[1] PLX Devices Inc., 440 Oakmead Parkway, Sunnyvale, CA 94085, USA. Phone: +1 (408) 7457591. Website: http://www.plxdevices.com. (Last accessed website on 23 July, 2013.)

[2] *Torque Pro* by Ian Hawkins. Available from Google Play: https://play.google.com/store/apps/details?id=org.prowl.torque. (Last accessed website on 23 July, 2013.)

A first stage is dedicated to determining the traffic scenario in which the Host Vehicle (HV), i.e. the vehicle receiving the recommendations, is travelling. As determining the traffic scenario is a spatial-temporal problem, we choose a POI along the future trajectory of the HV, which will serve as an additional source of environmental information. This POI will be represented by a vehicle referred to as the Next Vehicle (NV), which can be a real vehicle, or a virtual vehicle (i.e. a point with vehicular-type dynamics), depending on the compliance of certain conditions. With an estimate of vehicular density and speed information from both the HV and NV, the traffic scenario is determined using a rule-based inference engine.

A second stage is devoted to the calculation of the recommended speed and distance, based on the traffic scenario information, the current HV and NV speed, and a decision matrix. Due to the dependence of the recommended speed on the NV speed, and the fact that the NV may be a virtual vehicle, the definition of a model to update the speed of such a virtual vehicle is also needed.

The experimental results presented in [5] were obtained through a conventional off-line simulation (every component is simulated) under some ideal environmental conditions (a straight road). Here, we want to go one step further and evaluate the performance of the methodology in a more realistic way by including the driving behaviour of a human being driving a real car through a less ideal road.

A.1. Further Particulars of the Demonstration Setup

In order to evaluate the speed advisory system using the proposed simulation platform, we included the following.

(i) Matlab-SUMO communication: The SUMO simulation is interfaced using TraCI (Python), while the recommendation algorithm to be evaluated with the simulation was developed using the Matlab FL Toolbox (Version 2.2.14, R2011b) [22]. In order not to be forced to port the recommendation algorithm developed in Matlab to Python, we integrated Matlab directly into the loop. For this purpose, we created a Remote Procedure Call (RPC) framework based on TCP-Sockets, which allows for directly calling Matlab functions from Python scripts.

(ii) The HMI: We used the set of displays shown in Fig. 3 (inspired by those presented in [23]), and the HMI shown in Fig. 4 to present the relevant information to the driver.

A.2. Experimental Results

We adopted the Toyota Prius as the real HV (represented by an avatar in the SUMO simulation) and drove it around a street circuit on the north campus of the National University of Ireland Maynooth, as shown Fig. 5, at a time of day when other (real) traffic on the road was minimal (in order to have a more controlled scenario, to avoid as much as possible external elements not represented in the simulation such as other real vehicles or pedestrians). We also simulated twenty-two vehicles with different characteristics (maximum speed, size, acceleration/deceleration capabilities, etc.) and emulated a variety of local speed limits (lower than the real speed limits on the road) to create bottlenecks of traffic involving the simulated vehicles and our real one.

Data from our test is recorded in Fig. 6. The objective was to monitor the consequences of a real driver trying to track a changing recommended speed under the safe conditions provided by the simulation platform. The following observations were made:

- 326 time steps of interest were analysed in total (where, in our experiment, one time step was equal to one second);
- for 186 (not necessarily consecutive) time steps, the driver followed the recommendations and received
  - 180 type 1 or 2 distance recommendations (i.e. “OK” or “Close” messages), all of these obtained for the FT (109), CT (36), and PB (35) traffic scenarios described in the caption of Fig. 3;
  - 6 type 3 distance recommendations (i.e. a “Very Close” message), most of these for FT when a vehicle suddenly appeared in front of the Host Vehicle when it was travelling at high speed;
- for 140 (not necessarily consecutive) time steps, the driver ignored the recommendations and received
  - 120 type 1 and 2 distance recommendations (i.e. “OK” or “Close” messages), most of these obtained for both FT (68) at very low density of vehicles around the Host Vehicle, and PB (38) and CT (11) when the speed violation w.r.t. the recommended speed was small and at low speeds, and so not very relevant (e.g. 305 s to 332 s);
  - 20 type 3 distance recommendations (i.e. a “Very Close” message), most of these obtained for FT (14) at high density of vehicles around the Host Vehicle, and for AP (4) when reaching congestion with high, unsafe speed.

Our tests thus proved useful for evaluating some interesting situations in a more realistic way. Warnings about unsafe distances were shown to the real driver, but with regards to emulated cars (not real cars), testing thus confirming to us that even challenging situations can be easily and safely tested, thus allowing a more comprehensive evaluation of the speed advisory system. The driver felt comfortable trying to comply

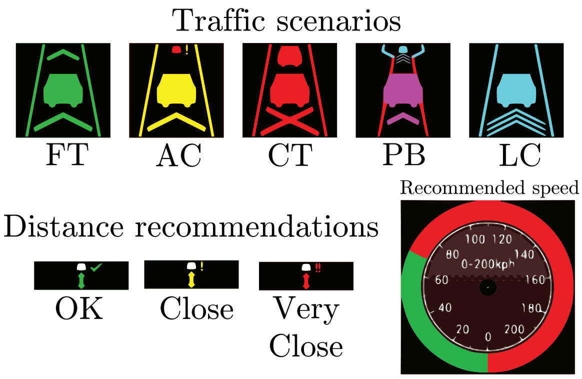


Fig. 3. Displays chosen for traffic scenarios, distance recommendations, and the recommended speed. (FT: Free Traffic, AC: Approaching Congestion, CT: Congested Traffic, PB: Passing Bottleneck, LC: Leaving Congestion).

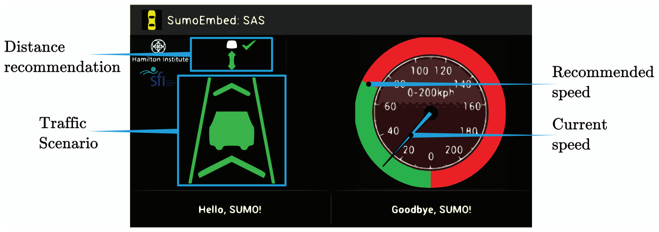


Fig. 4. HMI used to show the traffic scenario and recommendations to the driver.

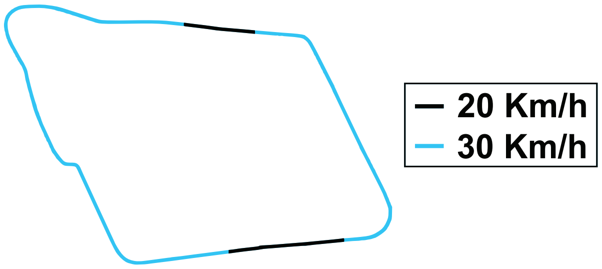


Fig. 5. Street circuit and local speed limits used for speed advisory system tests.

with the speed recommendations, provided that they were not issued too frequently.

B. Emissions Regulation

The second application that we implemented with our platform was the one introduced in [7], which aims to regulate traffic-related pollution in urban environments using hybrid vehicles. The notion is to individually direct each hybrid vehicle in a large fleet on which power source to utilise via simple communication signals from a central infrastructure. More specifically, power-split hybrid vehicles like the Toyota Prius can be operated in fully electric mode and in internal combustion engine mode, and thus the idea is to orchestrate and coordinate switching between drive modes to achieve regulated pollution levels. Based on estimates of regional pollution levels, this is achieved by incorporating a feedback-loop such that the collective vehicle engine mode behaviour maintains pollution below a safe, predefined level. While in [7], several control algorithms are proposed, here we are going to use the basic one: a simple integral controller in a stochastic framework. (For more detailed information, refer to [7].)

Although some of the experimental results provided in [7] were obtained using real traffic information, all of the developed simulations are of the conventional off-line type, putting full attention on the overall performance of the system but ignoring the individual driver experience. Therefore, here we expect to evaluate this important issue in parallel using the proposed simulation platform. We assume that we have access to all necessary environmental information (e.g. pollution quantity emitted from each vehicle, regional pollution targets).

B.1. Demonstration Setup

We implemented the above described control algorithm in Matlab, and used the same Matlab-SUMO communication device as we did for the speed advisory system presented in Section IV.A.1. Further application-specific aspects of our demonstration setup are described as follows.

(i) Mechanical interface: A mechanical machine-machine interface was used to physically interact with the (real) hybrid vehicle (i.e. the Toyota Prius). This was to permit automatic engine mode control (i.e. to permit automatic switching between petrol and electric driving mode) based on the output of the control algorithm. This interface consists of an artificial finger which receives the engine mode command from the smartphone, via bluetooth communication, and then pushes/releases the drive mode switch in the vehicle accordingly.

(ii) Visual interface: Concerning the human-machine interface to monitor the current state of the vehicle and the overall network, we used the visual display and avatars as presented in Fig. 7.

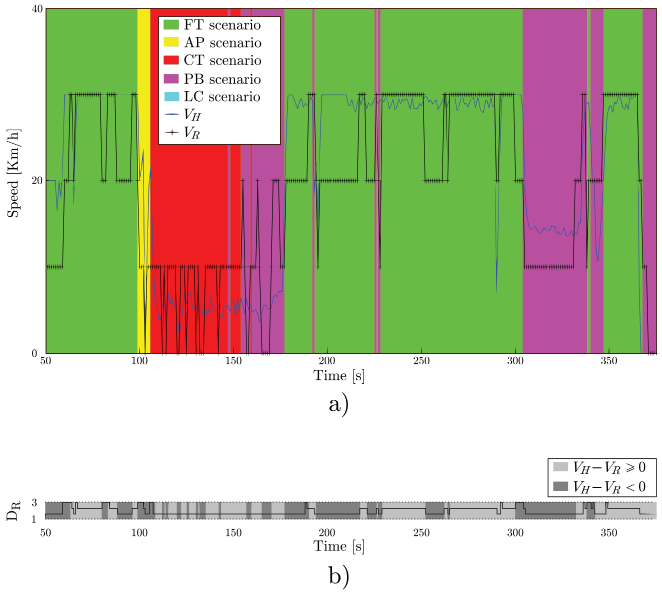


Fig. 6. Experimental results for the speed advisory system: a) Host Vehicle speed (V_H) and maximum speed recommendation (V_R) versus traffic scenarios; and b) distance recommendations (D_R). Acronyms for traffic scenarios are those in the legend of Fig. 3, and values for distance recommendation are according to the following convention: 1 for “OK,” 2 for “Close,” and 3 for “Very close.” Speed recommendations are provided to the driver in intervals of 10 km/h in an effort to make compliance with them easier.

B.2. Experimental results

We drove the twinLIN car around the same street circuit presented in Fig. 5, using the same SUMO configuration (local speed limits, number/type of simulated vehicles, etc) as we did for the speed advisory system demonstration. The results of our experimental test are presented in Fig. 8.

Since engine mode changes were automatic, meaning that the driver did not have to interact with the car manually based on the output of the pollution control algorithm, we focused our attention on the driver experience while the regulation task was ongoing during the demonstration. Concerning this, results in Fig. 8 show that pollution regulation is achieved as expected, and the feedback from the driver was that the automatic engine mode regulation did not generate any inconvenience while driving the real vehicle. A future aim is to experiment with different battery loads. A different driver experience is to be expected if made to switch to electric engine mode when limited battery power is available.

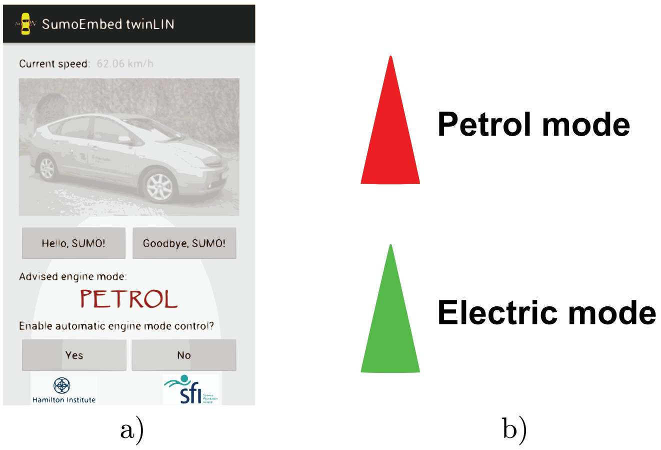
C. Local Rerouting Around an Obstruction

There exist many ways to influence congestion or pollution in urban areas. In our third application, we concentrate on routing with feedback regulation as an instrument [24]. We consider the problem of routing vehicles around a partial road obstruction: for example, a school zone, with limits on traffic throughput, or traffic pollution levels, that come into effect at 9 o'clock in the morning. Assuming a sudden enforced road capacity decrease on this now partially obstructed road, we wish to instantly begin to reroute some of the affected vehicles along alternative roads while, at the same time, permitting some vehicles access to the obstructed road but in a regulated manner such that the vehicular flow (number of vehicles per unit of time) on this road is reduced. We want to achieve this in an *ad hoc* manner that avoids all vehicles choosing the same (new) route, even though all vehicles share the same origin and destination (i.e. all vehicles originate from a common point in the road network prior to the road obstruction, and desire to get to a common point after the road obstruction). Further details of our approach are as follows.

C.1. Demonstration Setup

Again, for our demonstration, we utilised the road network of the National University of Ireland Maynooth's north campus; in particular, the roads depicted in Fig. 9. We used Google Maps APIs to embed a map into our smartphone User Interface (see Fig. 10). Our embedded vehicle with real driver was the Toyota Prius.

At the beginning of the scenario, there is no obstruction, and vehicles may go freely from the start node to the end node according to any route planning algorithm (e.g. shortest path). In our setup, this route is the inner lap going via the link with the traffic alert (which does not take effect until 120 time steps after the start of the demonstration). At time step 120, we emulated a partial obstruction, and hence a decrease in the link capacity on that path (i.e. the red link). As a consequence, we desire that not all approaching vehicles stick to this initial route choice (i.e. through the obstructed route s_1) but choose from one of the two other possible routes, s_2 and s_3, with probabilities p_1, p_2 and p_3 according to a feedback control loop for the vehicular flow F_1 on s_1 (see Fig. 11), and a balancing algorithm for vehicular flows F_2 and F_3 on s_2 and s_3, respectively.


SumoEmbed twinLIN

Current speed: 62.06 km/h

Advised engine mode:

PETROL

Enable automatic engine mode control?

Fig. 7. Human-machine interface for the regulation of emissions methodology: a) visual display at the smartphone side, and b) colour convention for avatars at the SUMO-GUI side.

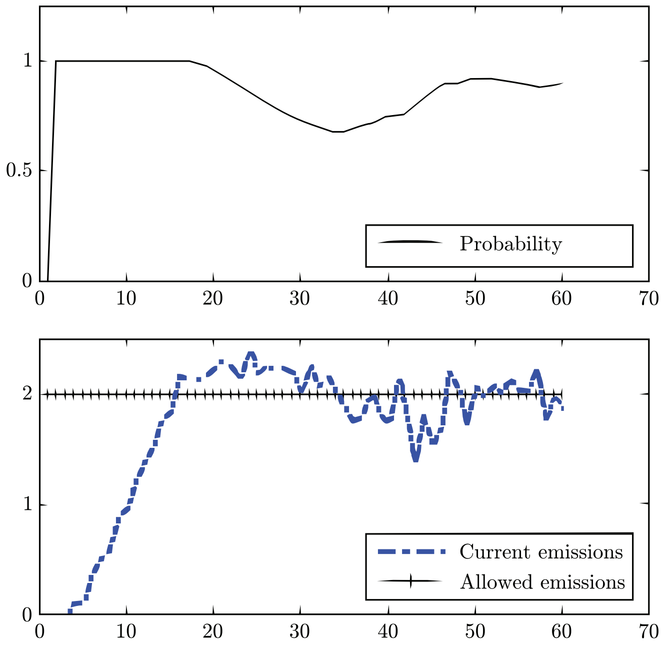


Fig. 8. Experimental results for the twinLIN application: evolution of the overall CO emissions (blue) in grams per $\frac{1}{12}$ minute (each step equal to 5 s), the set-point for CO emissions (yellow), and the probability for all of the vehicles to be at petrol mode (black).

For our example shown in Fig. 9 and Fig. 11, we have (the controller)

$$p_1(k) = \begin{cases} 0, & \text{if } e(k) < 0 \\ 0.5, & \text{otherwise} \end{cases}$$

and

$$p_{23}(k) = 1 - p_1(k),$$

where we wish to control the vehicular flow on route s_1, and where $p_{23} = p_2 + p_3$ is such that $\{p_2, p_3\}$ guarantee that F_2 and F_3 are balanced. This balancing is done by letting vehicles, which have not chosen s_1, choose between the route alternatives s_2 and s_3 stochastically. Probabilities may be related to the current conditions, e.g. noise emissions on that particular route. For the sake of simplicity, we want the vehicular flows F_i for $i \in 2, 3$ on both routes balanced, i.e. the probability p_i to choose route s_i with $i \in 2, 3$ is calculated according to

$$p_i = \alpha\left(1 - \frac{F_i}{F_2 + F_3}\right), \quad \alpha = \begin{cases} 1, & \text{if } p_1 = 0, \\ 0.5, & \text{otherwise.} \end{cases}$$

We ran our demonstration for 278 seconds (i.e. time steps) with an initial vehicular flow of 30 cars per minute released

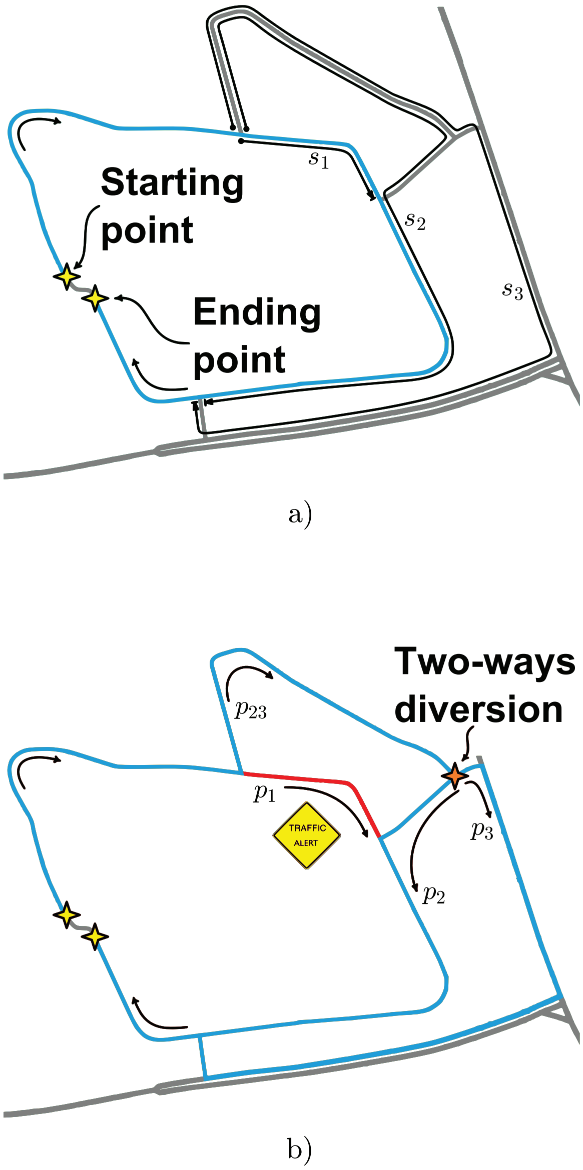


Fig. 9. Road network: a) default street circuit (blue) through route s_1; and alternative routes s_2 and s_3; and b) probabilities to choose the available routes due to a partially blocked default path (i.e. the red section).

at the start node. The partial obstruction and thus our control and balancing approach began at timestep 120. Results from the test are shown in Fig. 12.

C.2. Results

Fig. 12 shows that, at 74 seconds, vehicles reach the beginning of s_1. They go freely through s_1 until time step 120, when the obstruction occurs. From time step 121, s_1 is partially obstructed, and the control approach begins. Note that s_1 is made up only by the obstructed section (coloured red in Fig. 9) and, after going through s_1, every car belongs to s_2. That is why Fig. 9 shows vehicles associated with s_2 before time step 120. Finally, after a transient of about 100 time steps, vehicular flow F_1 is properly controlled (see the upper part of Fig. 12), which can be easily identified through the evolution of $\bar{F}_1$ (average of F_1 with a moving window of the last 50 time steps); and vehicular flows $F_{2,3}$ are balanced (the lower part of Fig. 12).

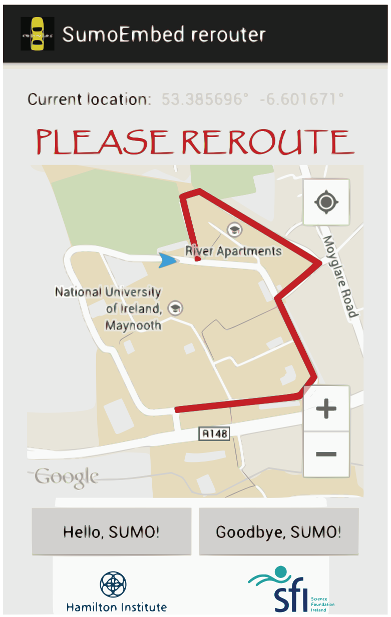


Fig. 10. Smartphone User Interface for the rerouting system.

Our hardware-in-the-loop simulation platform allowed a real driver to gain a feel for the rerouting system being tested. The Prius was equipped with the smartphone that displayed the reroute dealt to the driver in our test on a map. The driver indeed felt some initial irritation at having to reroute to s_3 (the dealt route), especially since s_3 took the driver off-campus; however, the driver was able to follow the reroute successfully.

V. Conclusions and Future Work

A hardware-in-the-loop simulation platform for emulating large-scale intelligent transportation systems was presented. The platform embeds a real vehicle into SUMO. A goal of the platform is to provide drivers with some sense of what it would feel like to participate in a large-scale, connected vehicle scenario, and thus allow ITS developers to better examine real driver reactions in regards to feedback control for applications in an ITS context.

We envision making several immediate enhancements to the platform, including: (i) better map-matching for those applications that are location-aware; (ii) integration of our platform with a network simulator (e.g. ns-3), and the addition of further embedded real vehicles, to incorporate vehicle-to-vehicle

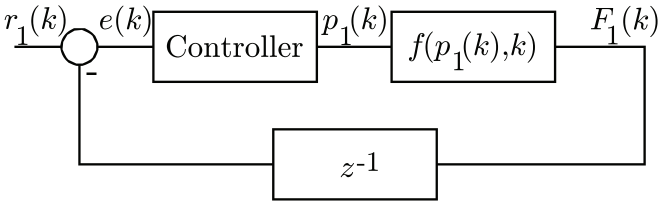


Fig. 11. Feedback control for vehicular flow $F_1(k)$ through the partially obstructed section, where r_1 is the set point, and p_1 is the probability to take such a section, which leads to the vehicular flow F_1 on that route.

communication into our simulations; and (iii) improvements in the modular design of the platform (e.g. a custom-designed, common welcome menu on the smartphone user interface that presents the user with a full choice of ITS applications to test, and to which new ITS applications can easily be installed). We aim to demonstrate further ITS applications; and we aim to conduct further applications testing with a greater number of drivers, and to gather more data on their experiences (e.g. survey questions).

References

[1] EcoGem. Available at: http://www.ecogem.eu. (Last accessed: 17 February 2014.)

[2] European Green Cars Initiative. Available at: http://www.green-cars-initiative.eu. (Last accessed: 17 February 2014.)

[3] T. Tielert, M. Killat, H. Hartenstein, R. Luz, S. Hausberger and T. Benz, The impact of traffic-light-to-vehicle communication on fuel consumption and emissions, in *Proceedings of the Internet of Things 2010 Conference*, Tokyo, Japan, 2010.

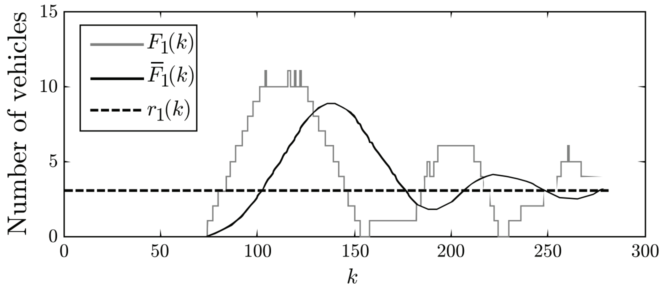
[4] D. Krajzewicz, J. Erdmann, M. Behrisch and L. Bieker, Recent development and applications of SUMO – Simulation of Urban MObility, *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3 & 4, 2012, pp. 128-138.

[5] R. H. Ordóñez-Hurtado, W. M. Griggs, K. Massow and R. N. Shorten, Intelligent speed advising based on cooperative traffic scenario determination, in H. Waschl, I. Kolmanovsky, M. Steinbuch and L. del Re (Eds.), *Lecture Notes in Control and Information Sciences: Optimization and Optimal Control in Automotive Systems*, vol. 455, 2014, pp. 77-92, Springer.

[6] J. Pan, M. A. Khan, I. Sandu Popa, K. Zeitouni and C. Borcea, Proactive vehicle re-routing strategies for congestion avoidance, in *Proceedings of the 8th IEEE International Conference on Distributed Computing in Sensor Systems*, Hangzhou, China, 2012, pp. 265-272.

[7] A. Schlote, F. Häusler, T. Hecker, A. Bergmann, E. Crisostomi, I. Radusch and R. Shorten, Cooperative regulation and trading of emissions using plug-in hybrid vehicles, *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, 2013, pp. 1572-1585.

[8] A. Schlote, C. King, E. Crisostomi and R. Shorten, Delay-tolerant stochastic algorithms for parking space assignment, to appear in *IEEE Transactions on Intelligent Transportation Systems*. DOI: 10.1109/TITS.2014.2304356



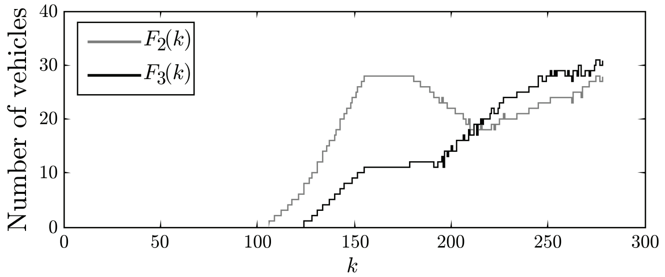


Fig. 12. Evolution of the vehicular traffic, where $F_i\,(k)$ is the vehicular flow on route s_i at instant k, $\bar{F}_1$ is the average of F_1 with a moving window of the last 50 timesteps, and r_1 is the set point for F_1.

[9] V. Punzo and B. Ciuffo, Integration of driving and traffic simulation: issues and first solutions, *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, 2011, pp. 354-363.

[10] F. Dion, J.-S. Oh and R. Robinson, Virtual testbed for assessing probe vehicle data in IntelliDrive systems, *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, 2011, pp. 635-644.

[11] J.L.F. Pereira and R.J.F. Rossetti, An integrated architecture for autonomous vehicles simulation, *Proceedings of the 27th annual ACM Symposium on Applied Computing*, pp. 286-292, 2012.

[12] J. Macedo, Z. Kokkinogenis, G. Soares, D. Perrotta and R.J.F. Rossetti, A HLA-based multi-resolution approach to simulating electric vehicles in Simulink and SUMO, *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 2367-2372, 2013.

[13] Y. Zhao, A. Wagh, K. Hulme, C. Qiao, A. W. Sadek, H. Xu and L. Huang, Integrated traffic-driving-networking simulator: a unique R&D tool for connected vehicles, in *Proceedings of the 1st International Conference on Connected Vehicles and Expo*, Beijing, China, 2012, pp. 203-204.

[14] VSimRTI - Smart Mobility Simulation. Available at: https://www.dcaiti.tu-berlin.de/research/simulation. (Last accessed: 16 May 2014.)

[15] M. Quinlan, T.-C. Au, J. Zhu, N. Stiurca and P. Stone, Bringing simulation to life: a mixed reality autonomous intersection, in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 6083-6088.

[16] R. Mangharam, D. Weller, R. Rajkumar, P. Mudalige and F. Bai, GrooveNet: a hybrid simulator for vehicle-to-vehicle networks, in *Proceedings of the 2nd International Workshop on Vehicle-to-Vehicle Communications*, San Jose, California, USA, 2006.

[17] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer and J-P. Hubaux, TraCI: an interface for coupling road traffic and network simulators, in *Proceedings of the 11th Communications and Networking Simulation Symposium*, Ottawa, Canada, 2008, pp. 155-163.

[18] SUMO (Simulation of Urban MObility). Available at: http://www.sumo-sim.org. (Last accessed: 9 September 2013.)

[19] JOSM (Java OpenStreetMap Editor). Available at: http://josm.openstreetmap.de. (Last accessed: 10 September 2013.)

[20] XMLStarlet. Available at: http://xmlstar.sourceforge.net. (Last accessed: 10 September 2013.)

[21] R. H. Ordóñez-Hurtado, W. M. Griggs, K. Massow and R. N. Shorten, Evaluation of a new intelligent speed advisory system using hardware-in-the-loop simulation, in *Proceedings of the 2nd International Conference on Connected Vehicles and Expo*, Las Vegas, Nevada, USA, 2013.

[22] MathWorks, *Fuzzy Logic Toolbox^{TM} User's Guide*, The MathWorks, Inc., Natick, Massachusetts, USA, 1995-2014.

[23] E. Adell, A. Várhelyi, M. Alonso and J. Plaza, Developing human-machine interaction components for a driver assistance system for safe speed and safe distance, *IET Intelligent Transport Systems*, vol. 2, no. 1, 2008, pp. 1-14.

[24] F. Häusler, R. H. Ordóñez-Hurtado, W. M. Griggs, I. Radusch and R. N. Shorten, Closed-loop flow regulation with balanced routing, submitted to the *3rd International Conference on Connected Vehicles and Expo*, Vienna, Austria, 2014.

Wynita Griggs received her B.Sc. (Hons) degree in Mathematics from the University of Queensland, Brisbane, Australia in 2002 and her Ph.D. in Engineering from the Australian National University, Canberra, Australia in 2007. She is currently a postdoctoral research fellow at the Hamilton Institute, National University of Ireland, Maynooth. Her research interests include stability theory with applications to feedback control systems; and smart transportation.

Rodrigo H. Ordóñez-Hurtado received his degree in Engineering in Industrial Automatica from the University of Cauca, Colombia, in 2005. He commenced his Ph.D. in Electrical Engineering at the University of Chile in March 2008, and between April-November 2012, he was an intern at the Hamilton Institute, a multi-disciplinary research centre established at the National University of Ireland in Maynooth. In November 2012, he received his Ph.D. degree from the University of Chile, and since December 2012 he has held a postdoctoral position at the Hamilton Institute, working with Professor R. Shorten and his research group. Rodrigo's interests include robust adaptive systems (control and identification), stability of switched systems, swarm intelligence, large-scale systems and intelligent transportation systems. His focus is on applications to the mining industry and transportation systems.

Emanuele Crisostomi received the Bachelor's degree in computer science engineering, the Master's degree in automatic control, and the Ph.D. degree in automatics, robotics, and bioengineering, from the University of Pisa, Italy, in 2002, 2005, and 2009, respectively. He is currently an Assistant Professor of electrical engineering with the Department of Energy, Systems, Territory and Constructions Engineering, University of Pisa. His research interests include control and optimisation of large-scale systems, with applications to smart grids and green mobility networks.

Florian Häusler is currently working on vehicle-2-x topics; in particular, collaborative mobility. He is interested in Smart Mobility solutions including electric vehicles, car sharing and intelligent traffic control. He received his Diploma in Computer Science and Business Administration from the Technical University of Berlin and is now deputy head of the department of Automotive Services and Communication Technologies (ASCT) at Fraunhofer Institute FOKUS.

Kay Massow is currently working in the fields of Intelligent Transport Systems and Cooperative Driving Applications. He received his Diploma in Computer Engineering from the Technical University of Berlin in 2008. In the past, he has worked for Daimler and Volkswagen of America. Now he is team leader at the department of Automotive Services and Communication Technologies (ASCT) at Fraunhofer Institute FOKUS. Additionally, he assists teaching at the Technische Universität Berlin.

Robert Shorten graduated from University College Dublin (UCD) in 1990 with a First Class Honours B.E. degree in Electronic Engineering. He was awarded a Ph.D. in 1996, also from UCD, while based at Daimler-Benz Research in Berlin, Germany. From 1993 to 1996, Prof. Shorten was the holder of a Marie Curie Fellowship at Daimler-Benz Research to conduct research in the area of smart gearbox systems. Following a brief spell at the Center for Systems Science, Yale University, working with Professor K. S. Narendra, Prof. Shorten returned to Ireland as the holder of a European Presidency Fellowship in 1997. Prof. Shorten is a co-founder of the Hamilton Institute at NUI Maynooth, where he was a full Professor until March 2013. He was also a Visiting Professor at TU Berlin from 2011-2012. Professor Shorten is currently a senior research manager at IBM Research Ireland. Prof. Shorten's research spans a number of areas. He has been active in computer networking, automotive research, collaborative mobility (including smart transportation and electric vehicles), as well as basic control theory and linear algebra. His main field of theoretical research has been the study of hybrid dynamical systems.