Driver Behavior Injection in Microscopic Traffic Simulations

Manuel Lindorfer[1](✉), Christian Backfrieder[1], Christoph Mecklenbräuker[2], and Gerald Ostermayer[1]

[1] Research Group Networks and Mobility, FH Upper Austria, Hagenberg, Austria
{manuel.lindorfer,gerald.ostermayer}@fh-hagenberg.at
[2] Christian Doppler Lab Wireless Technology for Sustainable Mobility,
Vienna University of Technology, Vienna, Austria
cfm@nt.tuwien.ac.at

Abstract. The individual behavior of drivers has a significant influence on the characteristics of vehicular transportation systems such as safety, capacity or traffic flow. Apparently, considering such behaviors in the scope of microscopic traffic simulations is inevitable in order to accomplish simulations close to reality. In recent years, considerable efforts have been put into modeling longitudinal and lateral movements of vehicles or their lane-change behavior, respectively. However, sometimes it is necessary to deviate from the standard behavior prescribed by these models in order to study the effects of exceptional situations in road traffic such as sudden braking maneuvers. This paper addresses this specific use case by introducing a generic behavior injection model, allowing for the integration of predefined driver behaviors into microscopic traffic simulations. Furthermore, it enables the reconstruction of real traffic scenarios by incorporating data gathered from vehicular measurement campaigns. The result is a simple, yet flexible model applicable to a wide range of microscopic traffic simulators.

Keywords: Microscopic traffic simulation · Driver behavior · Computer modeling

1 Introduction

The area of traffic simulation has gained more and more importance in recent years, playing a crucial role in the development of technologies and applications designed for Intelligent Transportation Systems (ITS). Simulations are a widespread and frequently used tool to model complex transportation networks and to investigate scenarios that cannot be studied in a real experiment or by any other analytical method. Throughout the years, numerous simulation frameworks have been developed by researchers in the field, designed for the simulation of vehicular traffic at different levels of abstraction [1–5]. In contrast to macroscopic simulators, which describe entire traffic flows in their collectivity,

© Springer Nature Singapore Pte Ltd. 2017
M.S. Mohamed Ali et al. (Eds.): AsiaSim 2017, Part II, CCIS 752, pp. 237–248, 2017.
DOI: 10.1007/978-981-10-6502-6_21

microscopic traffic simulators provide the highest level of detail, as the movements of every single vehicle and its characteristics are modeled individually. Frameworks belonging to the class of the latter make use of a variety of models which encapsulate single tasks of the driver, including but not limited to behavioral models, such as lane-change models and longitudinal models, and fuel consumption models.

In the last decades, considerable efforts have been put into modeling driver behavior in various aspects, resulting in a vast number of behavior models available in literature (e.g. [6–15]). Although most of these models describe vehicular traffic adequately, they do not allow for simulating situations that deviate from their prescribed behavior. This, however, is sometimes useful, if not required, especially when studying the effects of exceptional situations in road traffic such as sudden braking maneuvers or the impacts of individual driving behavior on both traffic flow and safety.

In order to overcome this deficiency, we introduce a generic behavior injection model, which is capable of enriching microscopic traffic simulations by a number of predefined, customizable driver behaviors. Furthermore, it allows for the reconstruction of real traffic scenarios by integrating data gathered from vehicular measurement campaigns or driving simulator studies such as speed or acceleration traces. The model makes use of a flexible and decoupled data structure, allowing for a straightforward extension and modification of the very same.

The remainder of this paper is organized as follows. The next section gives an overview of related projects in the scope of this work. In Sect. 3, the developed behavior injection model is introduced, including a detailed description of its individual components and their interactions. Subsequently, we demonstrate the model's applicability by integrating it with the microscopic traffic simulator TraffSim [1]. Section 5 concludes the paper and gives a short overview of planned future work.

2 Related Work

The importance of microscopic traffic simulation and driver behavior modeling in particular is increasing continuously. Starting already in the 1960s, a multiplicity of behavioral models have been developed for that reason. While several models have been proposed allowing for the accident-free simulation of vehicular traffic under idealized conditions (e.g. [6–8]), others incorporate behavior such as a delayed response (e.g. [9–11]) or perceptual limitations (e.g. [12]) in order to model the impacts of human driving on traffic flow to a more copious extent. Additionally, various models try to capture the dynamics concomitant with lane-changes, e.g. [13–15]. Due to the ever growing demand for a even more realistic simulation of vehicular traffic, several efforts have been put into the coupling of traffic simulation frameworks and driving simulators in recent years. All these systems aim for integrating specific driving behavior with microscopic traffic simulations for the purpose of investigating scenarios that cannot be captured by the standardized behavior predetermined by general simulation models.

On that account, several attempts were made to integrate a driving simulator with commercial traffic simulation tools such as Paramics [16] and VISSIM [17], e.g. in [18–21]. Similar systems have been proposed by [22–24], who integrate the driving simulation engine SCANeR™ with the microscopic simulator Aimsun [25]. Maroto et al. [26], for example, proposed a micro-simulation model with a user-driven vehicle surrounded by simulated traffic.

The possibility to control a subject vehicle externally using driving simulator input, while all other vehicles move according to their behavioral models, allows to study the effects of individual driving behavior on traffic flow dynamics and vice versa. A major limitation of such co-simulation approaches, however, is the high computational and technological complexity concomitant with the coupling of two independent, complex systems (e.g. synchronization and latency, proprietary data formats, eventually hardware setup). The reliance on driving simulator input reveals another issue, that is the inability to guarantee full reproducibility of the performed simulations, which, apparently, is inevitable in order to obtain reasonable results, especially when investigating the effects of individual driving behavior in different situations.

The behavior injection model presented in the forthcoming section addresses these particular issues. It makes use of a flexible data structure allowing for the straightforward extension and integration with microscopic traffic simulators while at the same time allowing for full reproducibility of the scenarios under investigation.

3 The Behavior Injection Model

In this section, the developed behavior injection model is elaborated in more detail. Subsequently, we present an overview of its architecture and the functional principle as well as the types of behavior which are supported by the model.

3.1 Functional Principle

As mentioned previously, the behavior injection model proposed in this paper is applicable to the class of microscopic traffic simulators, i.e. the positions and velocities of vehicles within a simulation run as well as additional parameters such as fuel consumption are modeled individually. Hereinafter, we give a brief overview of the model's layout and its interfaces required for the successful integration with such microscopic simulation frameworks. Figure 1 outlines the individual model components and their interactions with other simulator components in an abstracted manner.

The behavior injection module constitutes the integral part of the proposed model, responsible for managing the execution of behaviors throughout a simulation run. These behaviors are parametrized using external configuration data which specify, among others, the behavior's time of execution and its duration, as indicated by (1). The injection module on the other hand is directly linked to the simulation time updater, a core component of microscopic traffic simulators

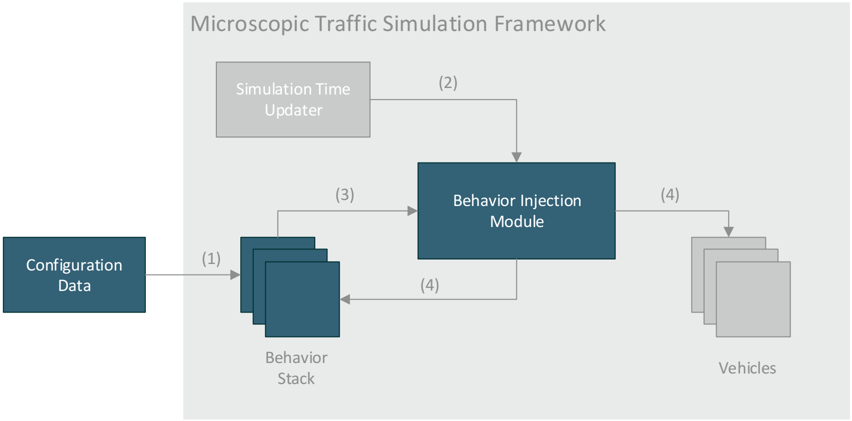


Fig. 1. Conceptional layout of the proposed behavior injection model. Interactions with model components (blue) and other simulator components (gray) are indicated by arrows, the parenthetic numbers indicate the sequence of interactions. (Color figure online)

which repeatedly updates other simulator components (e.g. crash detectors, longitudinal and lane-change models, traffic control models) in fixed time intervals in order to drive on the simulation (2). In every simulation step the injection module processes the given set of behaviors (3) and performs modifications either to this set or selected vehicles in the simulation, respectively (4). Thereby, it follows the functional principle outlined by the pseudo-code below.

```
function behaviorUpdate($simRunTime$)
    for $i = 1$ :sizeof($behaviorStack$) do
        $b \leftarrow$getBehavior($behaviorStack, i$)
        if canExecute($b$,$simRunTime$) then
            if canFinish($b$,$simRunTime$) then
                stopBehavior($b$)
                removeBehavior($behaviorStack$,$b$)
            else
                execute($b$,$simRunTime$)
            end if
        end if
    end for
end function
```

In every update step, the injection module verifies whether the execution of either one or multiple of the specified behaviors has to be started or stopped, respectively, depending on the currently elapsed simulation run time. For each behavior b the function canExecute ascertains whether the behavior is ready for execution or not, i.e. its time of execution is smaller than the simulation run time. If this condition evaluates to true, the function canFinish determines if the behavior has successfully finished execution, i.e. the current simulation run time exceeds the behavior's time of execution plus its specified duration. If this is the case, the behavior's execution is stopped. Furthermore, the behavior is removed from the set for performance reasons, and, thus, it is not considered in upcoming update time steps anymore. Otherwise, the behavior is executed just as desired, i.e. it is applied to the target vehicle.

This approach is not only a rather simple mechanism to alter the standardized behavior of selected vehicles in the scope of microscopic simulations, much more it also guarantees full reproducibility of the scenarios under investigation. This reproducibility is achieved by the behavior execution mechanism which ensures that behaviors with an identical time of execution and duration are injected at the same point in time and last for the very same period of time.

3.2 Behavior Types

Basically, the proposed model is capable of introducing any kind of behavior into microscopic traffic simulations. This comprises behaviors affecting a vehicle's longitudinal and lateral movements as well as such influencing its lane-change behavior, respectively. What's more, also the modeling of hazardous events such as driver distractions is conceivable. With reference to the exemplary integration of the injection model with the traffic simulator TraffSim [1], which is outlined in Sect. 4, we will delineate two specific behaviors that have been implemented for that particular reason.

Acceleration and Deceleration: The first type of behavior is related to acceleration and deceleration, respectively. It allows to force a subject vehicle to carry out an acceleration or a brake maneuver for a given period of time using a desired intensity. After behavior execution has finished, the vehicle's speed and acceleration values are either imposed by the underlying longitudinal model or remain unaffected until the end of the simulation, except for modifications caused by subsequent behaviors, depending on the behavior's parametrization. Such kind of behavior is useful e.g. when evaluating the stability of vehicle platoons in response to unexpected driving maneuvers of the leading vehicle (see e.g. in [11,27]).

Control Behavior: In contrast to the former, control behaviors allow to realize more complex driving scenarios by making use of predefined speed and acceleration traces. Such data could be obtained from real vehicular measurement campaigns or from driving simulator studies, respectively. Instead of setting the subject vehicle's acceleration to a predefined value, control behaviors continuously vary the vehicle's speed and acceleration in accordance to the provided data traces. Finally, the behavior's parametrization decides whether or not control is handed over to the vehicle's longitudinal model after the control behavior has finished. Control behaviors allows for the reconstruction of real driving scenarios in a simulation environment, which is useful not only when studying the effects of certain maneuvers under varying conditions, but also to validate car-following models under development on the basis of real data.

4 Integrating the Behavior Injection Model with the Microscopic Simulation Framework TraffSim

After having introduced the theoretical concepts related to the proposed behavior injection model, we will now demonstrate its applicability by integrating it with the microscopic simulator TraffSim [1]. TraffSim allows for the time-discrete and state-continuous simulation of vehicular traffic, supporting numerous configurable models and parameters. Throughout a simulation run a wide range of traffic-relevant data are recorded for each modeled vehicle, including but not limited to acceleration, current speed, position and fuel consumption, which guarantees full reproducibility of the performed simulations. TraffSim is implemented as Eclipse Rich Client Platform (RCP) application using the Java programming language, which reveals a number of benefits, including but not limited to platform independence, automated update mechanisms and a commonly known user interface [1].

4.1 Behavior Specification

TraffSim makes use of a number of XML files for defining all input parameters relevant for a simulation, e.g. road network, vehicles, traffic lights. We extend this data model by a separate input file which contains configuration parameters for the behavior types outlined in Sect. 3.2. Exemplary configurations of these types are shown in Fig. 2.

```
<Behavior id="1">
  <type>Acceleration</type>
  <vehicle>01</vehicle>
  <intensity>2.0</intensity>
  <duration>4.0</duration>
  <offset>60.0</offset>
</Behavior>
```

(a)

```
<Behavior id="2">
  <type>Control</type>
  <vehicle>01</vehicle>
  <offset>60</offset>
  <trace>
    <sample/>
    <sample/>
  </trace>
</Behavior>
```

(b)

Fig. 2. Configuration of an acceleration (a) and a control behavior (b) to be used in a TraffSim simulation.

It can be obtained that both configurations have several parameters in common, including the behavior *type*, an *offset* to determine the time of execution measured from simulation start, and the *vehicle* field identifying the target vehicle for the respective behavior. The acceleration and deceleration behaviors additionally require an *intensity* and a *duration* to be specified. Whilst the former indicates the absolute acceleration value (either positive or negative), the latter defines how long this value is actually applied.

For the control behavior, these two settings are defined implicitly by the *trace* parameter, which is composed of a set of any number of samples. Each sample, in turn, provides several parameters such as a time stamp, a speed and an acceleration value. Such traces can either be obtained by performing vehicular measurement campaigns, from driving simulators or from data provided by open-source projects such as NGSIM (**N**ext **G**eneration **SIM**ulation, [28]), respectively.

4.2 Data Model

Hereinafter, we outline the individual components of the behavior injection model and their relations to existing components within the TraffSim infrastructure. The model is strongly decoupled from the remaining simulator components, allowing for its modification or even replacement without affecting any of the other components. Figure 3 gives an overview of the relevant simulator architecture and the integration of the behavior injection model into the very same. Afterwards, we provide a more detailed description of the particular components and their respective responsibilities.

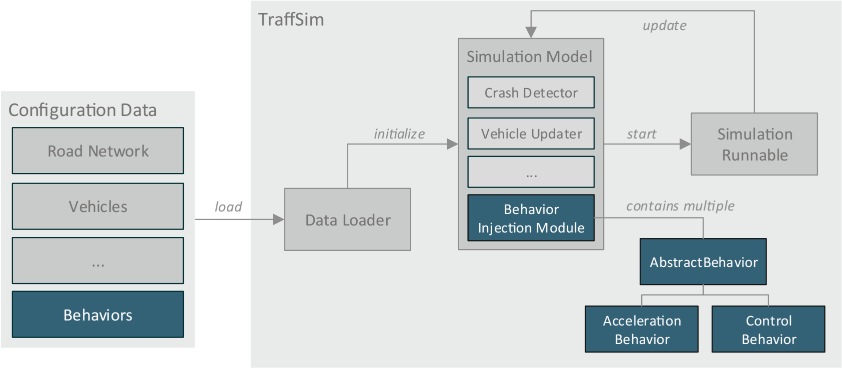


Fig. 3. Concrete implementation of the behavior injection model and its integration with the TraffSim infrastructure. Components associated to the behavior injection model are highlighted in blue. (Color figure online)

Data Loader: Before starting a simulation in TraffSim, all relevant configuration files have to be loaded into the simulation environment. The data loader provides the functionality to process the XML files containing the required input parameters (see Sect. 4.1) and to create the corresponding simulation objects from them, e.g. road segments, vehicles.

Simulation Model: Given the entities created by the data loader, a simulation model is created for every single simulation. This model provides all information required for a simulation run and manages a number of components which encapsulate particular simulation facilities, e.g. crash detection, statistics recording or, likewise, behavior injection.

Simulation Runnable: A simulation runnable constitutes the equivalent to the simulation time updater outlined in Fig. 1 and is one of the key components in TraffSim. It is responsible for continuously updating particular simulator components handled by the simulation model, e.g. crash detector, behavior injection module, in fixed time intervals in order to drive on the simulation. In every update step each of these components carries out its desired functionality. Each simulation runnable is executed in a separate thread, allowing to perform multiple simulation runs in parallel.

Vehicle Updater: One of the components controlled by the simulation runnable is the vehicle updater. This component is responsible for updating all vehicle-specific parameters such as position, speed and fuel consumption. In each simulation step it processes every individual vehicle in the simulation and updates its properties accordingly.

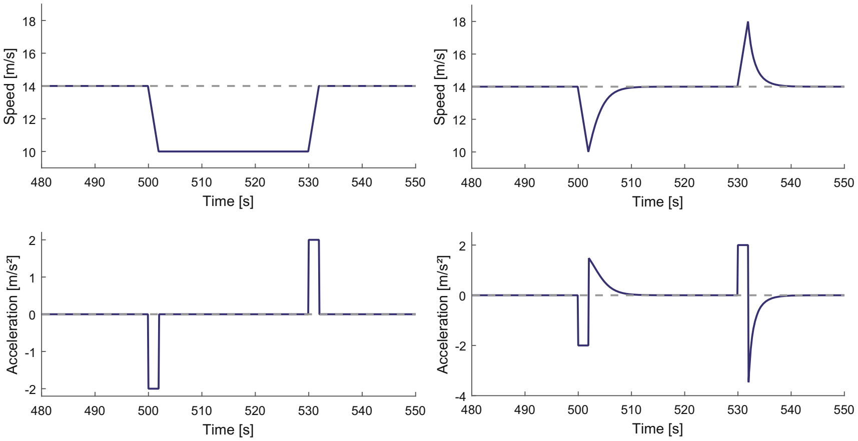
Behavior Injection Module: Among other components such as the vehicle updater or the crash detector, also the behavior injection module is updated in regular intervals by the simulation runnable. In every iteration the module executes the logic delineated in Sect. 3.1 in order to determine whether to execute a behavior or not. Once a behavior is ready for execution, the behavior injection module notifies the vehicle updater, which in turn updates the corresponding vehicles in simulation according to the speed and acceleration values provided by the respective behavior.

Abstract Behavior: This entity serves as a base class for all behaviors implemented in TraffSim. It provides the functionality to determine if a behavior is ready for execution and methods to obtain the desired manipulation values (e.g. speed or acceleration) for a given point in time and the target vehicle, respectively. The acceleration and control behavior represent concrete implementations of this abstract behavior. Whilst the former yields constant manipulation values at any point in time, these values vary over time for the latter, as in every update step a different sample of the associated data trace is processed. In that regard it is important to ensure that the sampling time of the underlying data trace matches the simulation interval in order to obtain reasonable results.

4.3 Simulation Results

In order to demonstrate the proposed behavior injection model's functionality we carried out a number of simulations using TraffSim. More precisely, we simulated different scenarios to outline how both the acceleration and deceleration behavior as well as the control behavior affect a target vehicle's longitudinal movement in a different manner. While in the first scenario a vehicle is being exposed to both an acceleration and a deceleration behavior, the target vehicle's behavior is imposed by a speed and acceleration trace obtained from the publicly available NGSIM project [28] in the second one. In all simulations performed in the scope of this work we used the popular Intelligent Driver Model [8] to model the vehicles' longitudinal movements.

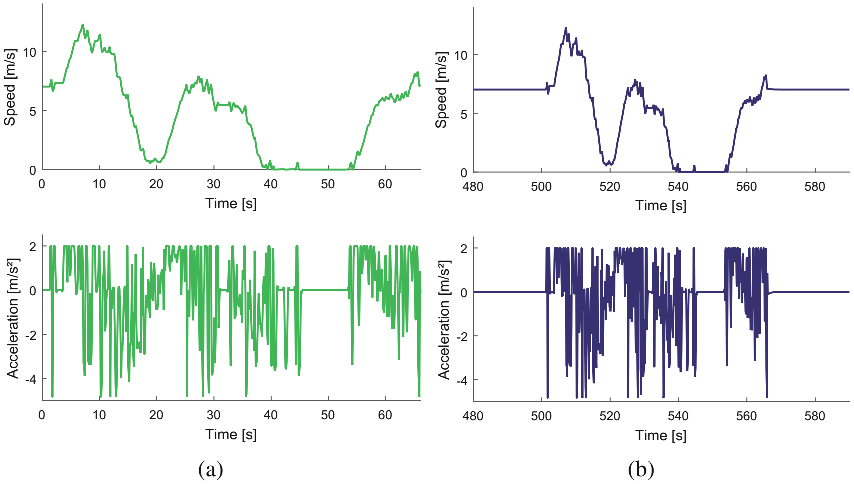
Acceleration and Deceleration: In this scenario we outline the implications of the most simple type of behavior, namely acceleration and deceleration maneuvers. To do so, we simulated a single vehicle traveling along a straight road at a desired speed of 14 m/s. After t = 500 s we inject a deceleration behavior which causes the vehicle to brake with an intensity of 2 m/s^2 until it reaches the new target speed of 10 m/s. Finally, an acceleration behavior is triggered at t = 530 s, lasting for two seconds and applying a constant acceleration of 2 m/s^2. Figure 4 outlines the acceleration and speed traces for the target vehicle in the relevant time interval. Figure 4a depicts the described scenario in the case where the vehicle's velocity and acceleration are affected only by subsequent behaviors after the first behavior has been executed. In contrast, Fig. 4b shows the very same scenario, however, control is handed over to the vehicle's longitudinal model after each of the behaviors has been executed. In case of the latter it can easily be seen that the longitudinal model immediately counteracts the deviated behavior introduced by the two behaviors in order to reach the desired target speed of 14 m/s by performing an according acceleration and brake maneuver, respectively.



(a) (b)

Fig. 4. Acceleration and speed traces of a target vehicle being exposed to a deceleration and an acceleration maneuver at t = 500 s and t = 530 s, respectively. The blue traces result from applying both a deceleration and an acceleration behavior, the greyed-out, dotted line corresponds to situations where no behavior is injected. (Color figure online)

Control Behavior: In the second scenario we demonstrate the proposed model's capability to reproduce realistic driving maneuvers by integrating speed and acceleration traces obtained from vehicular measurement campaigns or driving simulator studies, respectively. Therefore, we setup a control behavior which is parametrized using a corresponding trace from the publicly available Lankershim Boulevard dataset (refer to the NGSIM project [28]). The selected speed and acceleration traces have a length of roughly 66 s and were recorded in a congested arterial road, i.e. in dense traffic. Similar to the first scenario, we simulated a single vehicle traveling along a straight road at a desired speed of approximately 7 m/s, before the behavior is finally injected at t = 500 s. Figure 5a and b show the original traces as obtained from the NGSIM dataset and the corresponding traces reproduced with an accordingly parametrized control behavior, respectively.



(a) (b)

Fig. 5. Acceleration and speed traces of a selected sample from the Lankershim Boulevard dataset and its integration as control behavior on the right-hand side.

5 Conclusion and Outlook

In this paper, we introduced a behavior injection model allowing for the integration of predefined driver behaviors into microscopic traffic simulators. It is capable of manipulating a vehicle's longitudinal movement by altering its speed and acceleration in accordance to predefined acceleration and deceleration maneuvers or to real data traces obtained from vehicular measurement campaigns or driving simulator studies, respectively. In principle, also the integration of behaviors affecting the vehicles' lateral movements or lane-change behavior would be feasible. The proposed model is useful for a number of use cases, including but not limited to study the stability of vehicle platoons in response to unexpected driving maneuvers of the leading vehicle or to investigate the impacts of individual driving behavior under varying conditions. At all times, our model guarantees full reproducibility of the performed simulations by ensuring that identically parametrized behaviors are injected at the same point in time. We demonstrated the model's applicability by integrating it with the microscopic simulation framework TraffSim [1] and the aid of two exemplary simulation scenarios. However, it should be mentioned that the model is applicable to a wide range of microscopic traffic simulators.

We emphasize that, to this point, the behavior injection model is limited to modeling behaviors affecting the longitudinal movements of individual vehicles. Future work will include the extension of the model so as to allow for the integration of driving behaviors influencing both lateral movements and lane-change behavior, respectively.

Acknowledgments. The authors greatly acknowledge the support by the Austrian Research Promotion Agency (FFG) in the scope of the program "Industrienahe Dissertationen".

References

1. Backfrieder, C., Ostermayer, G., Mecklenbräuker, C.: Extended from EMS2013. TraffSim - a traffic simulator for investigations of congestion minimization through dynamic vehicle rerouting. Int. J. Simul. Syst. Sci. Technol. IJSSST **15**, 38–47 (2015)
2. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO - simulation of urban mobility - an overview. In: SIMUL 2011, The Third International Conference on Advances in System Simulation, pp. 55–60, October 2011
3. Gora, P.: Traffic simulation framework. In: 2012 UKSim 14th International Conference on Computer Modelling and Simulation, pp. 345–349, March 2012
4. Miller, J., Horowitz, E.: FreeSim - a free real-time freeway traffic simulator. In: IEEE Intelligent Transportation Systems Conference, ITSC 2007, pp. 18–23, September 2007
5. Treiber, M., Kesting, A.: An open-source microscopic traffic simulator. IEEE Intell. Transp. Syst. Mag. **2**(3), 6–13 (2010)
6. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y.: Dynamical model of traffic congestion and numerical simulation. Phys. Rev. E **51**(2), 1035–1042 (1995)
7. Gipps, P.: A behavioural car-following model for computer simulation. Transp. Res. Part B: Methodol. **15**(2), 105–111 (1981)
8. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E **62**(2), 1805–1824 (2000)
9. Bando, M., Hasebe, K., Nakanishi, K., Nakayama, A.: Analysis of optimal velocity model with explicit delay. Phys. Rev. E **58**(5429), 1035–1042 (1998)
10. Newell, G.F.: Nonlinear effects in the dynamics of car-following. Oper. Res. **9**(2), 209–229 (1961)
11. Treiber, M., Kesting, A., Helbing, D.: Delays, inaccuracies and anticipation in microscopic traffic models. Phys. A: Stat. Mech. Appl. **360**(1), 71–88 (2006)

12. Wiedemann, R.: Simulation des Strassenverkehrsflusses. In: Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, Germany (1974)
13. Kesting, A., Treiber, M., Helbing, D.: General lane-changing model MOBIL for car-following models. Transp. Res. Rec.: J. Transp. Res. Board **1999**, 86–94 (2007)
14. Toledo, T., Koutsopoulos, H., Ben-Akiva, M.: Integrated driving behavior modeling. Transp. Res. Part C: Emerg. Technol. **15**(2), 96–112 (2007)
15. Gipps, P.: A model for the structure of lane-changing decisions. Transp. Res. Part B: Methodol. **20**(5), 403–414 (1986)
16. Cameron, G., Wylie, B.J.N., McArthur, D.: Paramics: moving vehicles on the connection machine. In: Proceedings of the 1994 ACM/IEEE Conference on Supercomputing, Supercomputing 1994, pp. 291–300. IEEE Computer Society Press, Los Alamitos (1994)
17. Fellendorf, M.: VISSIM: a microscopic simulation tool to evaluate actuated signal control including bus priority. In: Proceedings of the 64th ITE Annual Meeting (1994)
18. Jenkins, J., Rilett, L.: Integrating driving simulators and micro-simulation models: a conceptualization. In: Transportation Research Board 81st Annual Meeting (2002)
19. Jin, M., Lam, S.H.: A virtual-reality based integrated driving-traffic simulation system to study the impacts of intelligent transportation systems (ITS). In: Proceedings, 2003 International Conference on Cyberworlds, pp. 158–165, December 2003
20. Vladisavljevic, I., Cooper, J.M., Martin, P.T., Stray, D.L.: 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: Proceedings of the 88th TRB Annual Meeting, Washington, D.C. (2009)
21. Hou, Y., Zhao, Y., Hulme, K.F., Huang, S., Yang, Y., Sadek, A.W., Qiao, C.: An integrated traffic-driving simulation framework: design, implementation, and validation. Transp. Res. Part C: Emerg. Technol. **45**, 138–153 (2014)
22. Punzo, V., Ciuffo, B.: Integration of driving and traffic simulation: issues and first solutions. IEEE Trans. Intell. Transp. Syst. **12**(2), 354–363 (2011)
23. That, T.N., Casas, J.: An integrated framework combining a traffic simulator and a driving simulator. Proc. Soc. Behav. Sci. **20**, 648–655 (2011)
24. Barceló, J., Casas, J.: Dynamic network simulation with AIMSUN. In: Kitamura, R., Kuwahara, M. (eds.) Simulation Approaches in Transportation Analysis, pp. 57–98. Springer, Boston (2005). doi:10.1007/0-387-24109-4_3
25. Aimsun: Microsimulator and Mesosimulator in Aimsun 6.1. User Manual. TSS - Transport Simulation Systems, Barcelona, Spain (2010)
26. Maroto, J., Delso, E., Felez, J., Cabanellas, J.: Real-time traffic simulation with a microscopic model. IEEE Trans. Intell. Transp. Syst. **7**(4), 513–527 (2007)
27. Davis, L.: Modifications of the optimal velocity traffic model to include delay due to driver reaction time. Phys. A **319**, 557–567 (2003)
28. Department of Transportation, U.S: NGSIM: Next Generation Simulation. http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm. Accessed 05 Jan 2017