Proceedings of the International Mechanical Engineering Congress and Exposition
IMECE 2014
November 14-20, 2014, Montreal, Canada

IMECE2014-39850

ASSESSING THE IMPACT OF A COOPERATIVE MERGING SYSTEM ON HIGHWAY TRAFFIC USING A MICROSCOPIC FLOW SIMULATOR

Ioannis A. Ntousakis, Kallirroi Porfyri, Ioannis K. Nikolos[1] and Markos Papageorgiou
Technical University of Crete, School of Production Engineering and Management
University Campus, GR-73100, Chania, Greece
[1]Contact Author

ABSTRACT

Vehicle merging on highways has always been an important aspect, which directly affects the capacity of the highway. Under critical traffic conditions, the merging of main road traffic and on-ramp traffic is known to trigger speed breakdown and congestion. Additionally, merging is one of the most stressful tasks for the driver, since it requires a synchronized set of observations and actions. Consequently, drivers often perform merging maneuvers with low efficiency. Emerging vehicle technologies, such as cooperative adaptive cruise control and/or merging-assistance systems, are expected to enable the so-called “cooperative merging”. The purpose of this work is to propose a cooperative merging system and evaluate its performance and its impact on highway capacity. The modeling and simulation of the proposed methodology is performed within the framework of a microscopic traffic simulator. The proposed model allows for the vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication, which enables the effective handling of the available gaps between vehicles. Different cases are examined through simulations, in order to assess the impact of the system on traffic flow, under various traffic conditions. Useful conclusions are derived from the simulation results, which can form the basis for more complex merging algorithms and/or strategies that adapt to traffic conditions.

Keywords: vehicle merging, cooperative merging, adaptive cruise control, V2V communication, speed synchronization.

1. INTRODUCTION

Microscopic simulation models have proven to be very useful and widely accepted tools for the analysis and management of transportation systems. Within these tools, lane-changing and merging models belong to the most complicated and critical ones [1]: lane-changing and merging are identified as significant sources of collisions and congestion on freeways under critical or heavy traffic conditions, while drivers’ psychological components have multiple dimensions that affect freeway merge decisions [2, 3]. Intelligent Transport Systems (ITS) applications, and by extension the forthcoming automation of vehicles, would probably deliver essential solutions to such congestion and conflict problems [4], along with the expected improvements in road safety, drivers’ comfort, accident reduction and increased road capacity [5].

Various techniques have been proposed for the representation of vehicle interactions at merges. Most micro-simulation models are based on a gap-acceptance, combined with a car-following model [1]. The gap acceptance theory implies that every driver has a critical gap to complete lane changing safely [6]. More specifically, each driver decides whether to accept or reject the available gap on the shoulder lane by comparing it to the critical gap (minimum acceptable gap). The first framework of modeling the structure of lane changing decisions was developed by Gipps [7]. His model leads to the decision whether it is physically possible, necessary and safe to change lanes, taking into account parameters such as traffic signals, obstructions, the presence of transit lanes and heavy vehicles, the speed, etc. Until today, there are various micro simulation models and packages, like CORSIM [8], Aimsun [9] and others [10], which apply lane changing behaviors based on Gipps’ model.

The micro-simulation model proposed by Hunt and Yousif [11] for the merging behavior at road networks was based on rules similar to those of the Gipps model. Yang and Koutsopoulos presented the Microscopic Traffic Simulator (MITSIM) package, which uses discrete choices for modeling lane changing behavior in combination with a gap acceptance model [12]. Also based on Gipps theory, the microscopic traffic simulator CORSIM by Halati et al. [8] distinguishes lane changes as mandatory (MLC) or discretionary (DLC). MLCs are performed when the driver must shift from the current lane to another (e.g. to use an off-ramp) in order to follow his route, whereas DLCs are performed when the driver changes lane just in order to improve his driving conditions (e.g. to overtake a slower vehicle). Ahmed [13] developed a lane changing model and an acceleration model to describe merging behavior under heavily congested traffic. Moreover, Hidas [4, 14] developed a massive multi-agent simulation system (SITRAS), in which a lane changing and merging algorithm was incorporated, based on Gipps' model and, additionally, taking into account the concepts of "forced merging" and "courtesy yielding".

Further developments include the addition of the cooperative behavior of vehicles, the so-called cooperative merging. As a matter of fact, automated merging systems has been a research topic as early as the 1960s, see [15] and the references therein. More recently, Wang et al. [2] proposed a framework for freeway merging, by combining an acceleration and a gap-acceptance model. They considered the cooperativeness of the vehicles on the main road introducing the cooperative lane changing (in order to allow vehicles to merge) and courtesy yielding (deceleration in order to create gaps). Additionally, Hidas [16] developed a merging model to examine the cooperative behavior of drivers, introducing explicit modeling of vehicle interactions by utilizing intelligent agent concepts. The definitions of free, forced and cooperative merges were included, whereby the cooperative merging depends on the decision of the lag driver rather than the merging driver. Choudhury et al. [10] presented a cooperative merging model, which has four levels of decision-making: normal gap acceptance, decision to initiate courtesy merging, decision to initiate forced merging, and gap acceptance for courtesy and forced merging. In the study of Loot [6], a new framework for modelling merging behaviour is proposed based on gap selection, instead of the usual gap acceptance theories, in combination with the cooperative behaviour of vehicles on the main road. Specifically, he pointed out that gap acceptance models, where the driver changes lane if the gap is large enough, do not respond to the actual merging behaviour. With the proposed model, every merging vehicle is able to find a suitable gap without being overtaken by many vehicles on the main road and without coming to a standstill at the end of the on ramp.

Undoubtedly, the most active areas of ITS studies are the advanced driver-assistance systems (ADASs), such as the adaptive cruise control (ACC) and cooperative adaptive cruise control (CACC) systems, and the automated highway systems (AHS), which offer the potential of substantial improvements in safety, efficiency, traffic flow volume and traffic stability [17, 18]. In recent years, the development of applications of that type by car manufacturers and the potential contribution of such systems to the merging behavior of the equipped vehicles, have further increased the interest of researchers on the subject. The case of automated vehicle merging into a platoon of automated vehicles at an on-ramp, in order to achieve a safe rather than highly efficient merging, was investigated in [19, 20, 21, 22] and [23]. Lu et al. [24] presented a real-time implementation of longitudinal control algorithm for vehicle merging for automated highway systems, proposing a concept of virtual platooning.

Furthermore, Ran et al. [18] developed a detailed model for simulating merging situations in one-lane AHS with dedicated through lane and entrance ramps, assuming that vehicles on this one-lane are operated fully automatically. The gaps are created on the main lane, following orders obtained by the infrastructure. Only if there is an available gap, the vehicle moves on; otherwise it stops on the ramp, waiting for the next available gap. Kesting et al. [25] investigated the impact of the expansion of ACC systems on the traffic dynamics, reporting that ACC-equipped vehicles can mitigate the traffic congestion, by simulating ramp vehicles merging to the main road. Davis [26] proposed a cooperative merging model for ACC vehicles to improve throughput (by 20%) and reduce travel times. In his strategy, an ACC equipped car on the main road decelerates in order to create a gap into which on-ramp cars can merge, adjusting its position according to the nearest front car.

This work reports the details of a cooperative merging system and its performance assessment, which was developed in order to ensure a near-optimal merging of vehicles entering the main flow. The microscopic traffic simulator of Aimsun provided the framework to develop and simulate the proposed methodology, by utilizing its SDK and API tools. The rest of the paper is organized as follows: In Section 2 the basic assumptions, regarding the proposed algorithms, are presented. In Section 3 two alternative merging algorithms are presented in detail. The car-following model is described in Section 4. The simulation framework and the simulation results are contained in Section 5, followed by Conclusions.

2. MODEL ASSUMPTIONS

The assumptions made about the vehicle systems, the infrastructure and vehicle communication should be clarified first. Since building new infrastructure or extending the existing one are actions that require excessive time, effort, and expenditure, the design of cooperative merging algorithms based solely on communication between vehicles, without the need of expensive infrastructure is a rational choice.

More specifically, the following assumptions have been adopted:

- All vehicles have up-to-date access to the exact geometry of the network (lengths of the on-ramps and acceleration lanes etc.).
- All vehicles are equipped with Vehicle-to-Vehicle (V2V) communication systems with a sufficiently large range. The specific kind of communication technologies or protocols, needed to enable such communication, is outside the scope of this work.
- Vehicles can exchange information regarding their current speeds, positions and neighboring vehicles (leader and/or follower) at a high sampling rate (for the performed simulations this was taken equal to 0.1s). Vehicles can communicate and exchange data with more than one vehicle quasi-simultaneously.
- The vehicle entering from the on-ramp (*merging vehicle*) makes automatically a choice regarding the targeted gap on the mainstream. The algorithm as well

as the criteria used to make this choice are discussed in the following section.

- After the targeted gap is selected, affected vehicles in the mainstream are informed about their new *"virtual leaders"* and adjust their speeds in order to modify their gaps or maintain the already existing ones.
- The whole cooperation process takes place in a pre-defined cooperation area (Fig. 1), which starts before the acceleration lane and ends in the merging point (MP).
- Vehicles are equipped with Adaptive Cruise Control systems (ACC).

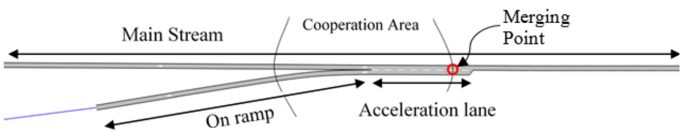

Figure 1. TOPOLOGY OF THE PROBLEM.

The previous assumptions are materialized in the proposed simulation methodology through the following procedures:

- The lengths and corresponding speed limits of all the segments of the network are stored in the beginning of the simulation and are available to all simulated vehicles.
- For every simulation step, all vehicle IDs, speeds and positions are stored in a database. This database can be accessed by all vehicles, therefore every vehicle has access to all vehicle data.

It should be pointed out that the proposed algorithm can be applied with minor modifications also in case only Vehicle-to-Infrastructure (V2I) communication is available.

3. DESCRIPTION OF THE MERGING ALGORITHMS

The proposed algorithms are initially designed for a simple network consisting of a single mainstream lane and a single on-ramp, leading to an acceleration lane, as depicted in Fig. 1. The reason for choosing such a network consisting of single lanes is to focus on the gap creation and gap selection methods rather than on lane changing, as a form of cooperation between vehicles. The development of a more comprehensive strategy, combining both actions in an optimal way, is part of the on-going work on the subject.

As indicated in Fig. 1, a cooperation area is defined, where communication and cooperation is taking place, and a merging point (MP) is fixed at a specific location on the acceleration lane, where the merging maneuver finally takes place. With these definitions in mind, the algorithms that were examined are the following:

- Algorithm 1: Vehicles will merge in the same sequence as they are entering the cooperation area.
- Algorithm 2: Vehicles will merge in a sequence which depends on the time they need to arrive to the merging point, assuming constant speeds inside the cooperation area.

Both algorithms specify and update (at each time step) a merging sequence (MS) (as in [5]), which is the order in which vehicles from both branches should merge; once a (on-ramp) merging vehicle has been placed in the MS, its position does not change in later updates of the MS. After execution of the merging algorithms, the speeds of the affected vehicles must be controlled in order to create new gaps or maintain the existing ones, for ensuring a safe and efficient merging process (Section 4).

Algorithm 1

This algorithm is based on the "First In First Out" (FIFO) or "First Come First Serve" concept for all arriving vehicles. Thus, any merging vehicle that enters the cooperation area, is inserted in the MS according to its absolute distance to the merging point. In case this distance is equal for two vehicles (a mainstream vehicle and a merging vehicle), the mainstream vehicle is selected to precede the merging vehicle. A pseudo-code of the merging algorithm is listed below:

For each simulation step:
Determine the IDs of the new vehicles that entered the cooperation area.
Measure their distance to Merging Point.
Sort them by distance ascending.
Place them with the same order at the end of the Merging Sequence (MS).
End

Algorithm 2

This second merging algorithm is based on a different concept, since it tries to determine the merging sequence in such a way that unnecessary decelerations (in the on-ramp and in the mainstream) are mitigated. To this end, the (projected) time to merging point is used to determine the insertion gaps for merging vehicles. The time to merging point is calculated (and updated) according to the current vehicle speed and its distance from the merging point.

More specifically, each new mainstream vehicle entering the cooperation area is placed at the end of the merging sequence since, for physical reasons, its time to MP will be longer than for preceding mainstream vehicles. On the other hand, for each merging vehicle entering the cooperation area, its net time to MP is calculated and is augmented by a typical time-headway to mitigate sharp merging maneuvers; eventually, the merging vehicle is inserted in the MS according to the updated times to MP of all MS vehicles behind the last merging vehicle (whose insertion gap cannot be modified).

For an illustrative example, we refer to Fig. 2. In the snapshot of Fig. 2, the current merging sequence includes vehicles 3, 2, 4, 5, and 6. Notice that the numbers used in Fig. 2 are completely random and serve only as IDs for the corresponding vehicles. A new merging vehicle (ID: 1) is entering the cooperation area and automatically retrieves the speeds and positions of all the vehicles inside this area (through V2V or V2I communication in reality). Then, the previous merging vehicle in the merging sequence is identified (in this case the vehicle with ID: 2).

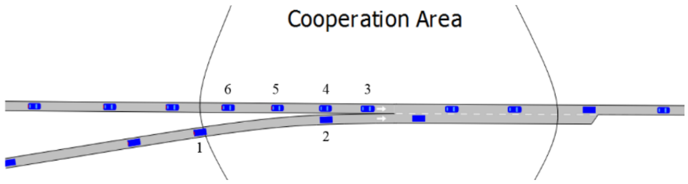

Figure 2. EXAMPLE OF GAP SELECTION FOR THE 2nd ALGORITHM.

Consequently, the possible gaps to enter are those formed behind this vehicle, namely the gaps between vehicles 2-4, 4-5, 5-6 and the gap behind vehicle 6. For all the vehicles involved in the formation of the previously mentioned gaps (vehicles 2, 4, 5, and 6), the algorithm, running in the entering vehicle (ID: 1), estimates their time to MP, based on their current speeds. These times are subsequently compared to its own (augmented) time to MP, and the new entering vehicle is placed in an accordingly updated MS. In the considered example let us assume that the following sequence of times to MP are computed: $t_2 < t_4 < t_1 < t_5 < t_6$. Therefore, the algorithm decides that vehicle 4 will be the new leader for vehicle 1, which will be the new leader for vehicle 5; the updated merging sequence results as 3, 2, 4, 1, 5, 6.

The outlined algorithmic logic guarantees the merging of vehicles from the on-ramp. However, in case of strong ramp demand, this may be to the detriment of the mainstream traffic, whose flow and mean speed may be accordingly lowered. If this situation is to be mitigated, additional constraints may be employed in the algorithm. For example, the merging vehicle should not be inserted in front of a mainstream vehicle with absolute distance to the merging point shorter than 45 m or speed lower than 10 m/s.

A pseudo-code of Algorithm 2 is listed below:

For each simulation step

> *Update the ID of the last merging vehicle (LMV) that has been inserted in the Merging Sequence (MS).*
>
> *Determine the IDs of the new vehicles that entered the cooperation area.*
>
> *Place all new mainstream vehicles at the end of the MS and compute their respective times to MP.*
>
> *For each new merging vehicle:*
>
> > *Compute the time to MP.*
> >
> > *Choose the potential followers from the MS. [Potential followers are all the mainstream vehicles in the MS which are placed behind the LMV, except those whose distance to merging point is less than 45 m or speed is less than 10 m/s.]*
> >
> > *Compare the time to MP with the current times to MP of all potential followers + 1.5 * Headway setting (1.5 factor is used for safety).*
> >
> > *Insert vehicle in MS, right in front of the first potential follower, and update the rest of the MS.*
>
> *End*

End

4. THE CAR FOLLOWING MODEL

Regardless which of the two algorithms is used to form the merging sequence (MS), we need to define how the decided MS is transformed into acceleration or deceleration commands for the vehicles. As it was stated earlier in Section 2, it is assumed that all vehicles in the network are equipped with Adaptive Cruise Control (ACC) systems and are enabled with V2V communication capabilities; the consideration of a penetration rate of equipped vehicles lower than 100% is the subject of on-going research. The control law for this system is a Constant Time Gap control law, described by the following equations [27]

$$\begin{aligned} \ddot{x}_{i,des} &= -\frac{1}{h_d}(\dot{\varepsilon}_i + \lambda\delta_i) \\ \varepsilon_i &= x_i - x_{i-1} \\ \delta_i &= x_i - x_{i-1} + h_d\dot{x}_i + L_{i-1} \end{aligned} \tag{1}$$

with

$$\begin{aligned} &max\_deceleration_i \le \ddot{x}_{i,des} \le max\_acceleration_i \\ &0 \le \dot{x}_i \le max\_speed_i \end{aligned} \tag{2}$$

where $\ddot{x}_{i,des}$ is the desired acceleration in m/s^2, h_d is the desired constant time gap in seconds, L_i is the length of the vehicle i, $\dot{x}_i$ is the speed of vehicle i, $max\_deceleration_i$ (which is a negative number) is the maximum admissible deceleration of vehicle i, $max\_acceleration_i$ is the maximum admissible acceleration of vehicle i, max_speed_i is the maximum desired speed of vehicle i and λ is a parameter set to 0.2 (Fig. 3).

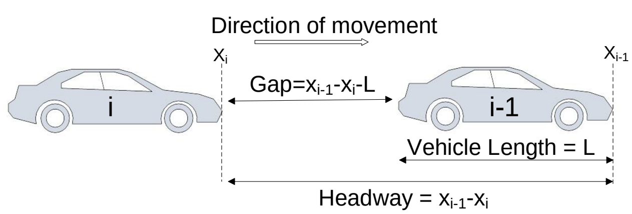

Figure 3. SCHEMATIC REPRESENTATION OF THE BASIC CAR-FOLLOWING PARAMETERS.

For the vehicle control in the merging context, we will use an approach similar to previous works (see, e.g., [15, 20, 28]). For the vehicles located inside the cooperation area, we will use the term *"actual leader"* to refer to the next downstream vehicle on the same lane, and the term *"virtual leader"* to refer to the vehicle which is registered ahead the current vehicle *i* in the merging sequence (Fig. 4).

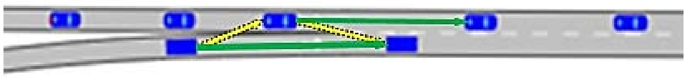

Figure 4. TWO-LEADERS ACC CAR-FOLLOWING MODEL. THE VIRTUAL LEADER OF EACH VEHICLE IS POINTED BY A DASHED ARROW (YELLOW); THE ACTUAL LEADER IS POINTED BY A SOLID ARROW (GREEN).

The application of the car-following model requires a (virtual) position for the virtual leader of each vehicle. Note that the virtual leader is different than the actual leader only if it is located on a different lane. Thus, the virtual leader is virtually moved to the lane of the vehicle in question; at a distance from the MP equal to its actual distance from the MP on its actual lane. We then have for each vehicle two possible cases:

1. Only one of the two leaders exists. The vehicle applies the equations above according to its actual or virtual leader.
2. Both leaders exist. The equations above are applied for both leaders; the most restrictive of both accelerations is selected.

Once the vehicle is out of the cooperation area, it will return to the standard ACC model, as previously described.

5. SIMULATION FRAMEWORK AND RESULTS

5.1. Simulation Environment

The microscopic simulator Aimsun was used to develop the proposed methodology and perform the corresponding simulations [29]. The Aimsun API module [30] offers the possibility to extend the functionalities of the basic Aimsun simulation environment by including user-defined applications in C++ or Python, which can exchange information dynamically with the Aimsun module. Additionally, the Aimsun microscopic simulator includes the MicroSDK tool [31], which, among others, allows for the modification or replacement of the incorporated car-following and lane-changing models. For the purposes of this work, we used all of the above mentioned tools, combined with an external database, to enable the required exchange of information between them. More specifically, the MicroSDK was used to implement the ACC car-following model as well as to store the vehicle speeds and positions in the database. The API was used to determine the Merging Sequence, as described in Section 3, by using the data stored in the database.

5.2. Simulation Setup

For the performed simulations we introduce two types of vehicles: *"Car"* and *"Merging Car"*, with identical attributes (maximum acceleration = 3.0 m/s^2, maximum desired deceleration = 4.0 m/s^2, maximum desired speed = 120 km/h, vehicle length = 4 m).

Simulations of 10 min duration (with additional 2 min warm-up period) have been performed, with various values for the on-ramp demand and different desired time-gap settings. The simulation step, for all the cases considered in this work, was set to 0.1 s.

For the performed simulations, the entering flow depends on the desired time-gap, which is the same for all vehicles in each simulation run. Several different on-ramp entering flows were used, spanning from 200 to 1300 veh/h. Four different values for the (constant for all vehicles) desired time-gap were considered, namely, 0.8 s, 1.0 s, 1.2 s, and 1.4 s. The distance from the entrance of the cooperation area to the start of the acceleration lane was set to 60 m, while the merging point MP was set 75 m further downstream.

5.3. Simulation Results

Successive snapshots of the cooperative merging procedure are presented in Figs. 5 and 6 for Algorithms 1 and 2, respectively. Figure 7 depicts the specific positions where detectors have been placed to collect measurements of the traffic flow values for evaluation purposes.

The 2nd algorithm is more general than the 1st one, which can be considered as a sub-case of algorithm 2. Both algorithms provided very similar results concerning the macroscopic values of flow and density, as it is depicted in Figs. 8 - 11. The videos from the performed simulations showed that both the developed algorithms provide smooth and rational merging, for all the different flow conditions that have been considered, without crashes, sharp maneuvers or other undesired situations.

Although the two algorithms have similar performances in terms of outflow, the second algorithm is closer to real driving behaviour, since it evaluates also forward gaps and avoids unnecessary decelerations. A careful look at Figures 5 and 6 may demonstrate the different gap selection behavior of the two algorithms. It is a subject of ongoing work to also study and measure the acceleration or deceleration efforts implied by each algorithm as it is one of the most significant factors regarding drivers' acceptance of such systems, as well as passengers comfort and safety.

For each specific desired time-gap, the increase of the on-ramp demand beyond a certain limit is seen in the figures to decrease the mainstream flow upstream of the ramp, while the mainstream outflow remains almost constant and independent of the on-ramp demand. This constant outflow corresponds to the highway capacity which depends on the time-headway of the vehicles. This observation remains the same for all different desired time-gaps, albeit at different levels depending on the employed desired time-gap. It is of high interest that the on-ramp demand is not affecting the capacity, which in turn means that the infrastructure is not underutilized.

The decrease in the main flow before the merging point is needed in order to create the required gaps for the merging of the on-ramp vehicles. This is more pronounced in the density graphs (Figs. 9, 11), where a substantial growth in density at the entrance of the main flow (detector 1) is observed with increasing on-ramp flow. This growth is alleviated by decreasing the desired time-gap, as shorter gaps are needed to be created for the merging vehicles. A similar observation can be made for the density at the entrance to the cooperation area from the on-ramp (detector 3).

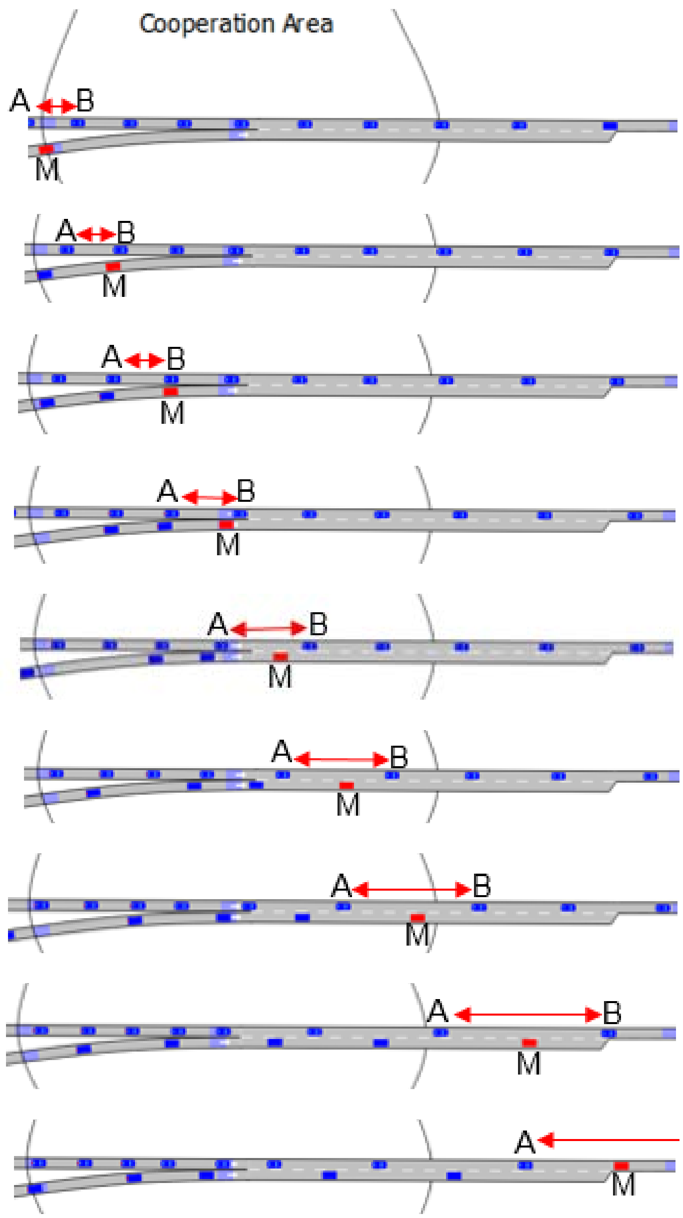

Figure 5. ALGORITHM 1: SUCCESSIVE SNAPSHOTS ILLUSTRATING THE COOPERATIVE MERGING.

At the exit of the main flow (detector 6), the density is independent of the on-ramp flow, as it exclusively depends on the desired time-gap. No comparison with manual vehicle merging situations are reported in this work; the reason is that many of the available simulation tools for manual vehicle merging may lead to unrealistic behaviors under certain flow conditions. Additionally, different lengths for the cooperation area have been studied with no significant impacts on the simulation results. However, the determination of the optimum cooperation area length necessitates a more thorough investigation; the maximum vehicle accelerations and decelerations should be considered as a criterion for this.

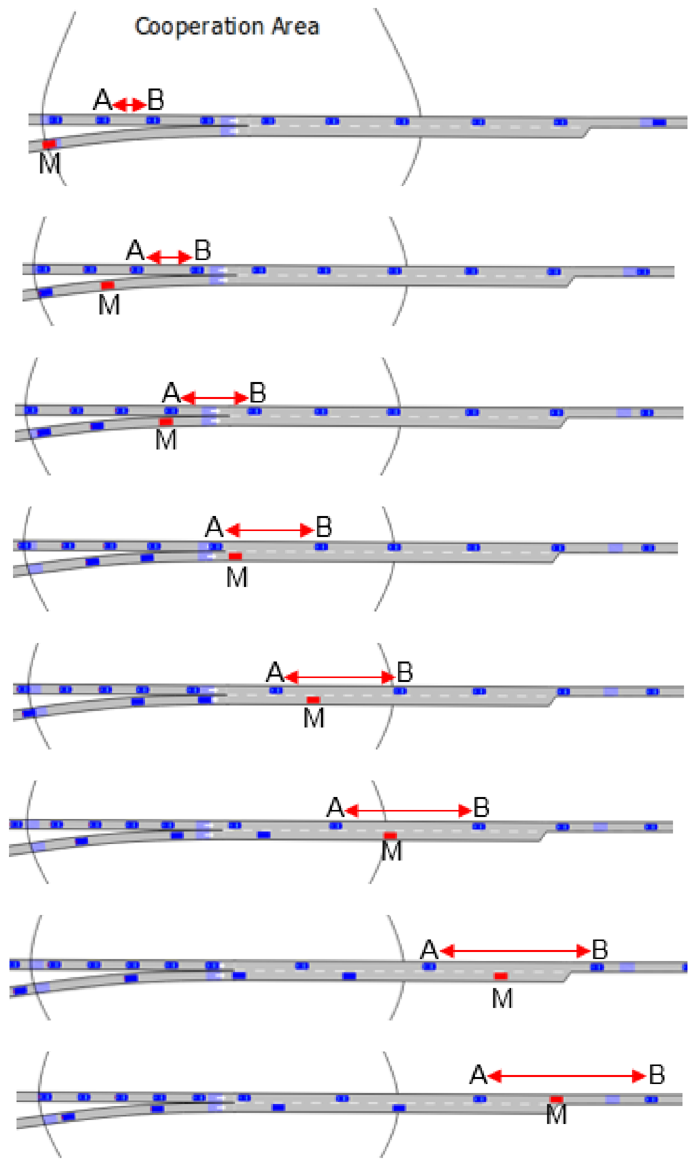

Figure 6. ALGORITHM 2: SUCCESSIVE SNAPSHOTS ILLUSTRATING THE COOPERATIVE MERGING.

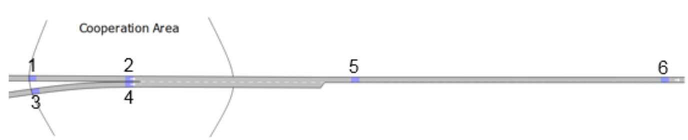

Figure 7. THE POSITION AND NUMBERING OF THE UTILIZED DETECTORS IN THE NETWORK.

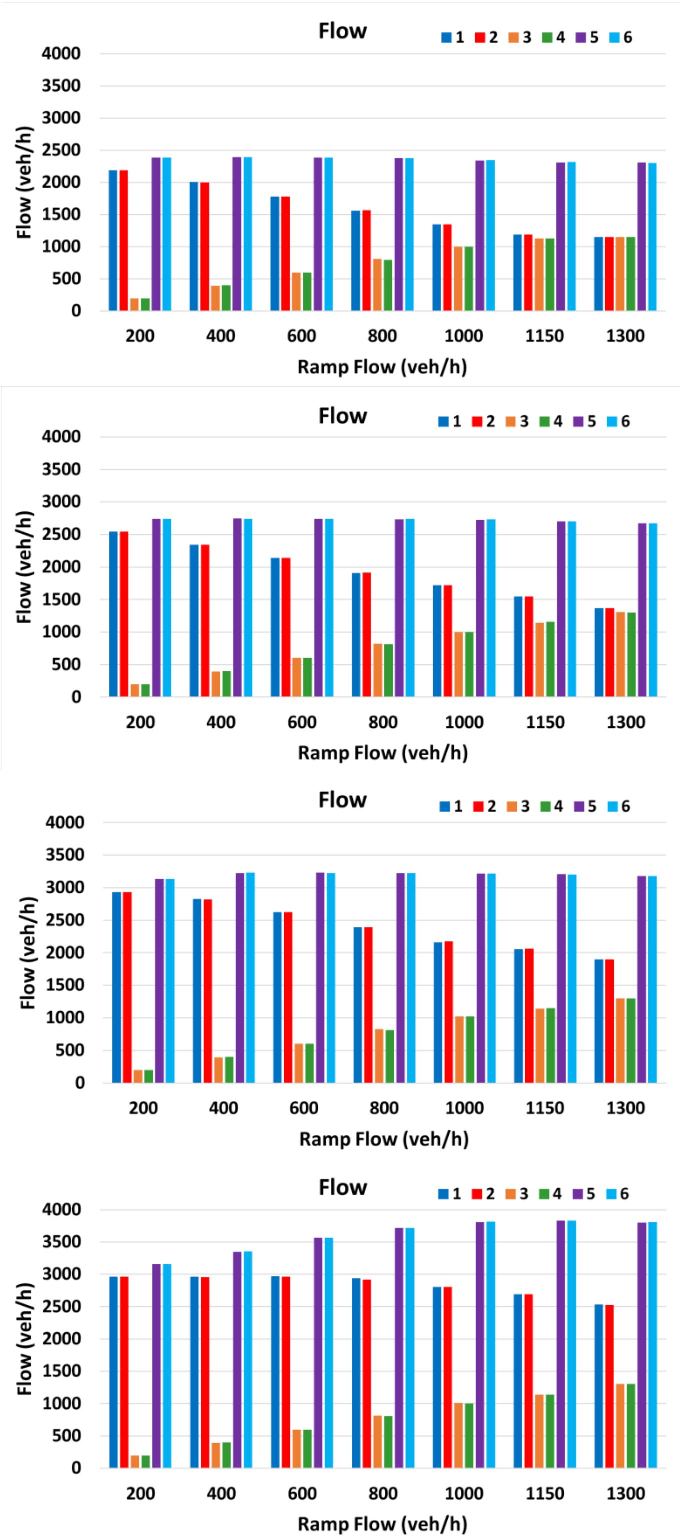

Figure 8. ALGORITHM 1. FLOW MEASUREMENTS AT THE 6 DETECTORS AS A FUNCTION OF ON-RAMP FLOW, FOR DIFFERENT DESIRED TIME-GAPS (TOP TO BOTTOM: 1.4s, 1.2s, 1.0s, 0.8s).

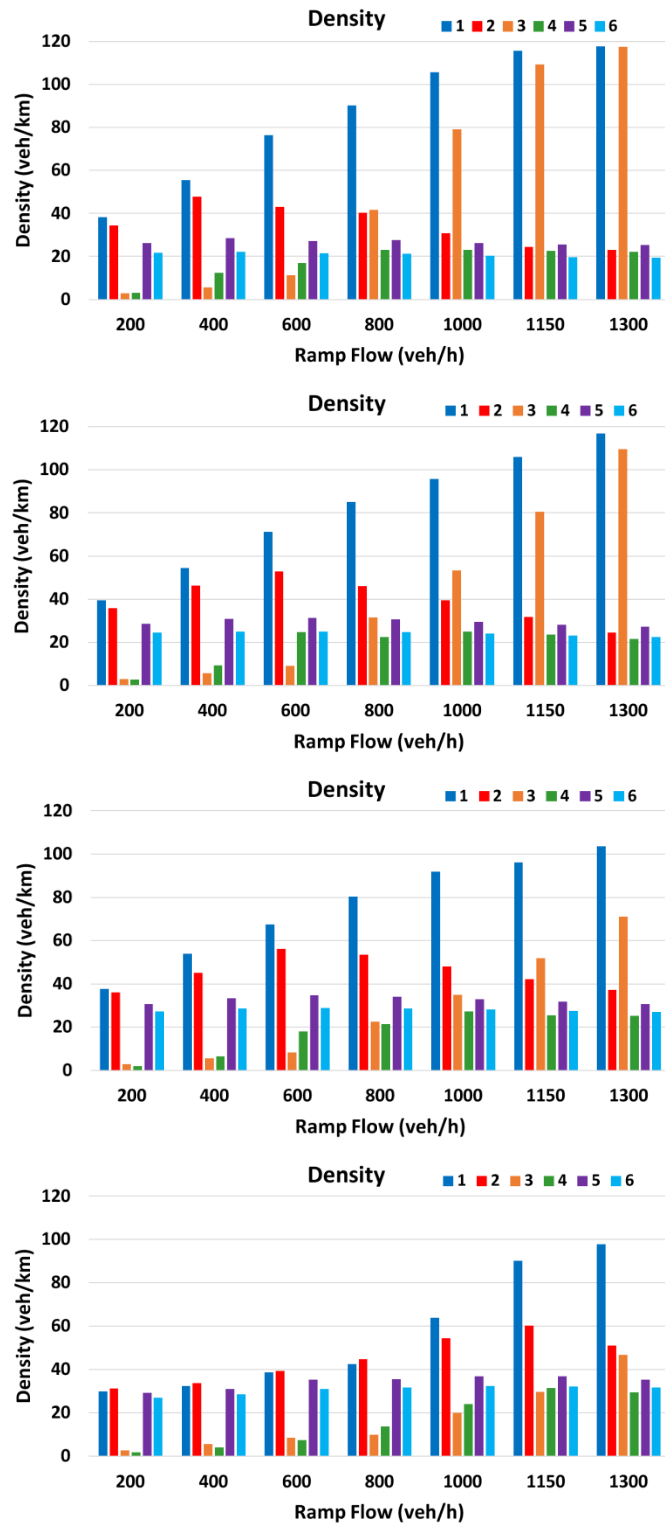

Figure 9. ALGORITHM 1. DENSITY AT THE 6 DETECTORS AS A FUNCTION OF ON-RAMP FLOW, FOR DIFFERENT DESIRED TIME-GAPS (TOP TO BOTTOM: 1.4s, 1.2s, 1.0s, 0.8s).

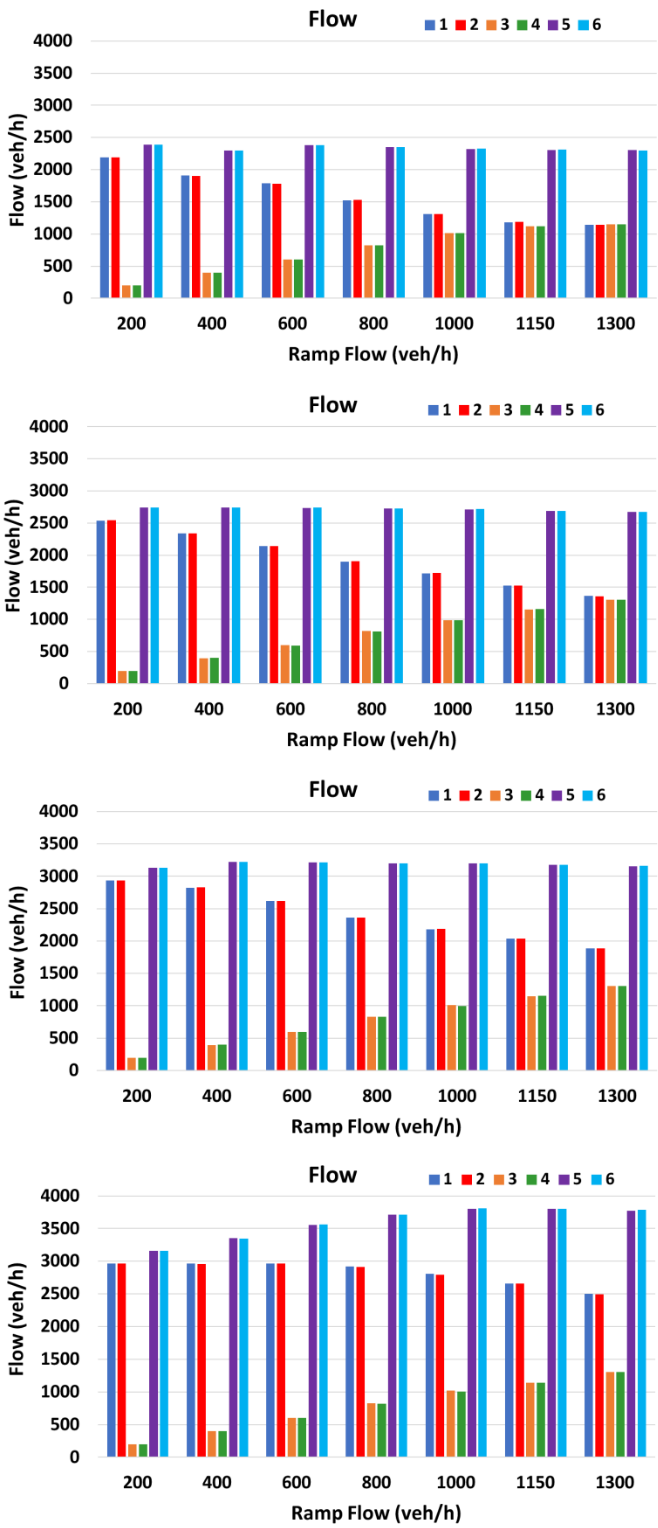

Figure 10. ALGORITHM 2. FLOW MEASUREMENTS AT THE 6 DETECTORS AS A FUNCTION OF ON-RAMP FLOW, FOR DIFFERENT DESIRED TIME-GAPS (TOP TO BOTTOM: 1.4s, 1.2s, 1.0s, 0.8s).

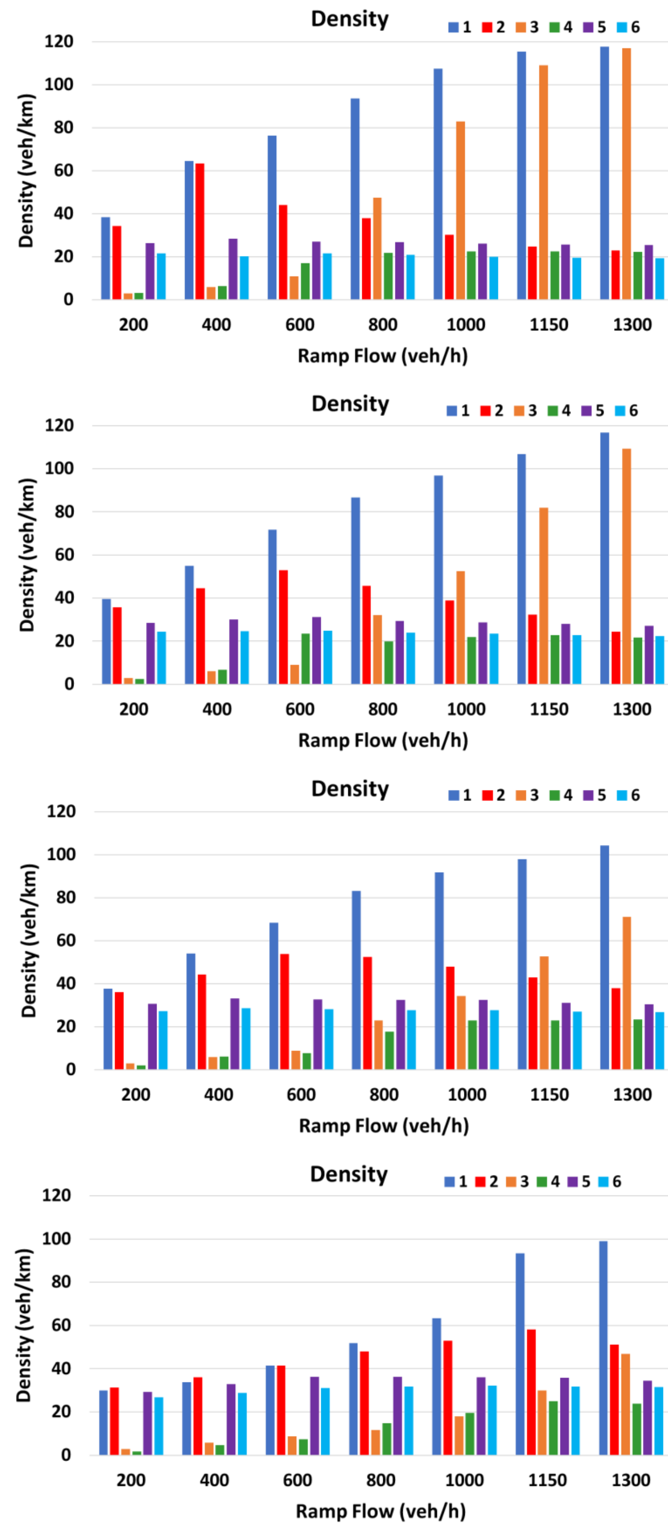

Figure 11. ALGORITHM 2. DENSITY AT THE 6 DETECTORS AS A FUNCTION OF ON-RAMP FLOW, FOR DIFFERENT DESIRED TIME-GAPS (TOP TO BOTTOM: 1.4s, 1.2s, 1.0s, 0.8s).

6. CONCLUSIONS

Two algorithms have been developed and evaluated in this work for the cooperative merging of vehicles in highways, within the framework of a microscopic traffic simulator; they were initially designed and tested for a simple network, consisting of a single mainstream lane and a single on-ramp, leading to an acceleration lane.

Aimsun was utilized as the simulation framework, and especially its SDK and API tools were used for the development of the corresponding subroutines. The proposed algorithms can simulate vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication, allowing for the cooperation of all vehicles traveling on the main motorway and those coming from the on-ramp. All vehicles were supposed to be equipped with ACC devices, obeying to a Constant Time-Gap control law. According to the 1st algorithm, vehicles are merging in the same sequence as they are entering the cooperation area, while for the 2nd algorithm the merging sequence depends on the time vehicles need to arrive to the merging point, assuming constant speeds inside the cooperation area.

Both algorithms were tested for a variety of on-ramp flows and desired time-gaps. The simulation results showed that the two algorithms have very similar performance, although different gaps may be selected by the entering cars, according to the algorithm used. However, the 2nd algorithm is closer to real merging, since it evaluates also forward gaps and mitigates unnecessary decelerations. For all the cases considered, the merging was very smooth, without unreasonable and irrational situations. Ongoing work includes the evaluation of other aspects of the two proposed algorithms, such as acceleration / deceleration efforts, speed reductions, fuel consumption, etc.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 321132, project TRAMAN21.

REFERENCES

[1] H. P. Kolen (2013), *Modelling Merging Behaviour on Freeway On-Ramps*, Master thesis, Civil Engineering and Geosciences, Delft University of Technology.

[2] J. Wang, R. Liu, and F. Montgomery (2005), “A Simulation Model for Motorway Merging Behaviour,” in *Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic Theory*, Institute for Transport Studies, University of Leeds, Leeds, pp. 281–301.

[3] C. C. Kou and R. Machemehl (1997), *Modeling Driver Behavior During Merge Maneuvers*, Technical Report, Southwest Region University Transportation Center, Austin, Texas.

[4] P. Hidas (2002), “Modelling lane changing and merging in microscopic traffic simulation,” *Transportation Research Part C Emerging Technologies*, 10, pp. 351–371.

[5] G. Raravi, V. Shingde, K. Ramamritham, and J. Bharadia (2007), “Merge algorithms for intelligent vehicles,” in *Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems*, Springer, pp. 51–65.

[6] M. Loot (2009), *Empirical analysis of merging behaviour*, Master thesis, Delft University of Technology.

[7] P. G. Gipps (1986), “A model for the structure of lane-changing decisions,” *Transportation Research Part B*, 20B(5), pp. 403–314.

[8] A. Halati, H. Lieu, and S. Walker (1997), “CORSIM-Corridor Traffic Simulation Model,” in *Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities*, pp. 570–576.

[9] J. Barceló and J. Casas (2005), “Dynamic network simulation with AIMSUN,” in *Simulation Approaches in Transportation Analysis*, *Recent Advances and Challenges*, Springer, pp. 57–98.

[10] C. F. Choudhury (2007), *Modeling Cooperative Lane Changing and Forced Merging Behavior*, Ph.D. thesis, Civil and Environmetal Engineering, Massachusetts Institute of Technology, Washington DC, United States.

[11] J. G. Hunt and S. Y. Yousif (1990), “Merging Behaviour at roadworks on Dual Carriageway Roads,” in *18th PTRC European, Transport, Highways and Planning Summer Annual Meeting*,University of Sussex, United Kingdom, pp. 55–68.

[12] Q. Yang and H. N. Koutsopoulos (1996), “A microscopic traffic simulator for evaluation of dynamic traffic management systems,” *Transportation Research Part C Emerging Technologies*, 4(3), pp. 113–129,

[13] K. I. Ahmed (199), *Modeling Drivers’ Acceleration and Lane changing Behavior*, Ph.D. thesis, Department of Civil and Environment Engineering, Massachusetts Institute of Technology.

[14] P. Hidas and K. Behbahanizadeh (1999), “Microscopic Simulation of Lane Changing under Incident Conditions,” in *14th International Symposium on the Theory of Traffic Flow and Transportation*, pp. 53–69.

[15] G. Schmidt, and B. Posch (2010), “Automatische Zusammmenführung zweier Fahrzeugströme - eine Rückblende,” *Automatisierungstechnik*, 58(6), pp. 317-321.

[16] P. Hidas (2005), “Modelling vehicle interactions in microscopic simulation of merging and weaving,” *Transportation Research Part C Emerging Technologies*, 13(1), pp. 37–62.

[17] V. Milanés, J. Villagrá, J. Godoy, J. Simó, J. Pérez, and E. Onieva (2012), “An Intelligent V2I-Based Traffic Management System,” *IEEE Transactions on Intelligent Transportation Systems*, 13(1), pp. 49–58.

[18] B. Ran, S. Leight, and B. Chang (1999), “A microscopic simulation model for merging control on a dedicated-lane automated highway system,” *Transportation Research Part C Emerging Technologies*, 7(6), pp. 369–388.

[19] P. Kachroo and Z. Li (1997), “Vehicle merging control design for an automated highway,” in *ITSC '97, IEEE Conference on Intelligent Transportation System*, pp. 224–229.

[20] A. Uno, T. Sakaguchi, and S. Tsugawa (1999), “A merging control algorithm based on inter-vehicle communication,” in *IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 783–787.
[21] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii (2002), “Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications,” *IEEE Transactions on Intelligent Transportation Systems*, 3(3), pp. 155–161,
[22] R. Sengupta (2003), “Simulation, analysis, and comparison of ACC and CACC in highway merging control,” *IEEE IV2003 Intelligent Vehicles Symposium Proceedings*, pp. 237–242.
[23] X. Y. Lu, H. S. Tan, S. E. Shladover, and K. J. Hedrick, (2010), “Automated Vehicle Merging Maneuver Implementation for AHS,” *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 41(2), pp. 85–107.
[24] X. Lu and K. J. Hedrick (2000), “Implementation of Longitudinal Control Algorithm for Vehicle Merging,” in *Proeeedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia*, pp. 450–455.
[25] A. Kesting, M. Treiber, M. Schonhof, F. Kranke, and D. Helbing (2007), “Jam-avoiding adaptive cruise control (ACC) and its impact on traffic dynamics,” in *Traffic and Granular Flow ’05, Springer*, pp. 633–643.
[26] L. C. Davis (2007), “Effect of adaptive cruise control systems on mixed traffic flow near an on-ramp,” *Physica A Statistical Mechanics and its Applications*, 379(1), pp. 274–290.
[27] R. Rajamani, D. Levinson, P. Michalopoulos, J. Wang, K. Santhanakrishnan, and X. Zou (2005), *Adaptive Cruise Control System Design And Its Impact on Traffic Flow*, Techical Report, Department of Civil Engineering, University of Minnesota.
[28] V. Milanes, J. Godoy, J. Villagra, and J. Perez (2011) “Automated On-Ramp Merging System for Congested Traffic Situations”, *IEEE Transactions on Intelligent Transportation Systems*, 12(2), pp. 500-508.
[29] Transport Simulation Systems (2013), Aimsun 8 Dynamic Simulators Manual.
[30] Transport Simulation Systems (2013), Aimsun 8 API Manual.
[31] Transport Simulation Systems (2013), Aimsun 8 MicroSDK Manual.