INFORMAL RULES FOR AUTONOMOUS VEHICLES IN SCANeR™

Benoit Lacroix [1,2], Vincent Rouelle[1], Andras Kemeny [1], Philippe Mathieu [2], Nicolas Laurent [3], Guillaume Millet [3] and Gilles Gallée [3]

[1] RENAULT, Technical Center for Simulation
Technocentre
1 avenue du Golf
78288 Guyancourt, France

e-mail: {benoit.lacroix, vincent.rouelle, andras.kemeny}@renault.com

[2] LIFL CNRS UMR 8022
University of Lille
Cité Scientifique bat. M3
59655 Villeneuve d'Ascq, France

e-mail: philippe.mathieu@lifl.fr

[3] OKTAL
8, rue de la Ferme
92100 Boulogne, France
e-mail: {nicolas.laurent, guillaume.millet, gilles.gallee}@oktal.fr

Abstract

In driving simulators, the realism of road traffic takes an important part in users immersion. It relies upon different elements, and in particular on the decision model of autonomous vehicles and the variety and consistency of their behaviors. In this paper, we describe the new traffic module of SCANeR™, which includes important improvements. New behavioral parameters have been introduced to take into account informal rules in the vehicles decisions. In addition, the dynamic model of the autonomous vehicles was enhanced to realistically render additional properties. Finally, a specific model takes into account the variety and consistency of drivers behaviors, which are crucial elements of the realism. Its implementation in the software is described, and the improvements it introduces are demonstrated using experimental results.

Introduction

Road traffic realism is an essential issue in driving simulation. Both behavioral and physical aspects of autonomous vehicles have to be realistic to immerse the driver in the simulated environment, and assess the validity of the experimentations.

The realism relies upon different elements. One of them is the behavior of the autonomous vehicles, resulting from their decision model. The introduction of dynamic parameters is an interesting solution to improve it, as it allows simulating the evolution of driver's internal state. The trajectories of the vehicles on the road are also crucial for the immersion: the dynamic model has to be accurate and to reflect real drivers evolutions. In addition, to be credible, the behaviors have to be various and consistent. These elements are often not specifically taken into account, and most of the time the influence of the different models parameters is not totally controlled. The conception of simulations by scenario designers remains thus a complex work, as unbalanced parameters can lead to incoherencies.

In this paper, we first describe the traffic decision model used in the driving simulation software SCANeR™. Various improvements have been brought in by the use of new behavioral parameters in the decision model, as well as an enhanced dynamic model for autonomous vehicles. Then we introduce a behavioral differentiation model aiming at easily creating various and coherent behaviors. We present its implementation in traffic simulation, as well as experimental results to demonstrate the interest of the approach.

Traffic simulation in driving simulators

The simulation of road traffic in the context of driving simulators presents different specificities, as the main objective is to immerse the driver in a realistic environment and to confront him with specific situations. The traffic has to be autonomous, but also controllable using scenario rules.

In most of the driving simulation software, the approach used to simulate the traffic is based on multi-agents systems architectures. The vehicles are considered as autonomous agents, interacting in the road context. Their internal decision model can be based on driving psychologist studies (Espié et al., 1994): in real world situations, drivers tend to reduce certain categories of interactions with other drivers; this mechanism is used as decision model for the autonomous vehicles. In other approaches, drivers actions are reproduced using sub-models of the driving tasks, like car following, lane changing, merging... A finite state automaton often handles the switch between the different available actions. It is sometimes improved to introduce parallelism and hierarchy in the computation (Wang et al., 2005): parallelism allows different states to be active simultaneously, and hierarchy to use automata as the parent automaton state; it enhances modularity and behavior producing. To limit the computation cost of the traffic simulation, the vehicles level of details can also be reduced depending on their distance to the interactive vehicle: faraway vehicles are simulated using a mesoscopic model, instead of a microscopic one (Olstam, 2005).

The variety of the produced behaviors is crucial for the realism of the simulation. Wright, Ward and Cohn (2002) showed that providing the autonomous vehicles with virtual personalities increased the immersion feeling of simulators users. Indeed, psychological factors are involved in driving (Dewar, 2002), and drivers take into account various elements to determine their actions (Björklung & Aberg, 2005): formal rules (mostly the rules of the road), informal rules (practices and conventions applied while driving, which may not respect the formal rules), design of the road (which often leads to the creation of informal rules), and other drivers behavior (their current behavior and the behavior we think they will adopt). Each of these elements can be used in the simulation to improve the realism of autonomous vehicles behavior.

The SCANeR™ traffic module

The driving simulation software SCANeR™ uses a distributed architecture, allowing balancing the load by distributing the different functional modules on computers connected through the network. The traffic module is based on a multi-agent architecture, where each vehicle is an agent (Champion et al., 1999). The autonomous vehicles use a perception-decision-action architecture (Figure 1): during the perception phase, the vehicles acquire knowledge of the surrounding world; during the decision phase, they compute their next goal; and during the action phase, their next position is determined.

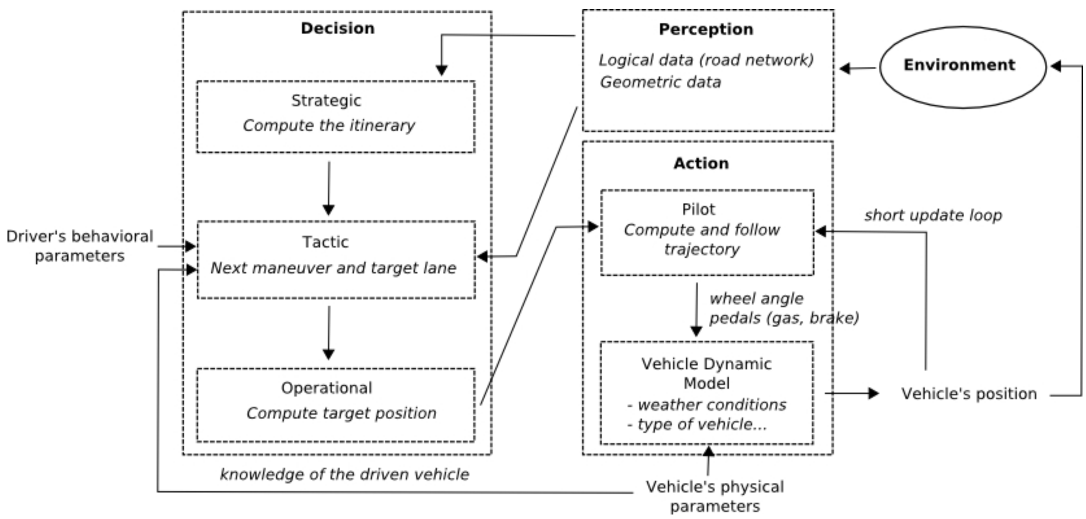

Figure 1: SCANeR™ vehicles architecture

Perception phase

The vehicles use two methods to get information on their environment. The first one is based on the logical data available from the road network description (Lacroix et al., 2007). For instance, road signs, next intersections or speed limits are retrieved this way. The logical description is also used to hierarchically organize the information about the surrounding vehicles: each vehicle is able to know which vehicle is driving on the same lane, on the same road, or approaching the same crossroad. The detection range is a function of the speed: a driver with a high speed looks farther than a slow one.

The second method is based on a geometric computation, used for emergency situations. Indeed, the logical data, relying on the description of the environment, might lead to miss vehicles being currently out of the road. Emergency are detected by computing the time to collision to close vehicles, according to their respective speed and direction. If a collision is imminent, the model switches to emergency status.

Decision phase

The decision phase is built on three levels: strategic, tactical and operational. The strategic level plans the itinerary: it is either defined by the users, or computed randomly at runtime.

The tactical level selects a short-term goal, combining the high-level goal with the environmental constrains. The objective is to optimize the displacement between intersections using lane changes, overtaking… A pre-existing finite state machine internally manages this level, allowing vehicles to decide which sub-models of the driving tasks they will use. The possible states are: “drive on” (the vehicle follows the car ahead, if any, or drives at its desired speed); “pre change lane” (the vehicle is preparing a lane change); “change lane” (the vehicle is changing lane); “pre overtake” (the vehicle is preparing to overtake); “overtake” (the vehicle is overtaking). A “blocked” state acts as fail state for all others. The transition between states is based on a scoring method.

At the operational level, the vehicles compute the acceleration and wheel angle needed to perform the maneuver determined at the tactical level. For instance, during the “drive on” state, the vehicle uses a car following model to compute its acceleration:

$$a = \alpha.(\Delta d - t.v) + \beta.\Delta v$$

where a is the resulting acceleration, Δd and Δv the distance and speed between current and ahead vehicle, $t.v$ the security distance (t a parameter of the model), and α, β adjustment control variables. A subsumption architecture is then added to select the final acceleration, checking that all rules are respected (emergency, road signs, maximal acceleration of the vehicle…). The wheel angle is computed to follow the road curvature, including two particularities: when the road turns, the vehicle cuts the curve, and a lane deviation factor is included to simulate the fact that vehicles do not always remain in the middle of their lane.

Different pseudo-psychological parameters are taken into account in this decision model. They are defined by the users, during the design of the simulation. The “maximal speed” parameter is the maximal speed the driver will use. The “safety time” describes the security margin it will adopt with the preceding vehicle, using its own instantaneous speed. The “overtaking risk” represents the risk a driver will accept to overtake, depending on the gaps with oncoming vehicles on other lanes. The “speed limit risk” allows it to bypass speed limits, and finally “observe priority” and “observe signs” are boolean rules regarding the respect of signalization and priorities.

Action phase

The action phase computes the new position of the vehicle. It is composed of two elements: a pilot and a dynamic model. The pilot uses the decision model outputs to provide the input values to the dynamic model. The trajectory to reach the goal point is computed, and a geometric algorithm is used to stay close to it. The instantaneous target is a point on the trajectory, where the vehicle should be in a few time steps. This algorithm regulates its own errors, using a short update loop on the position and speed.

The dynamic model uses simplified dynamic equations to lower the computation time and limit the number of parameters. However, each part of the car is modeled (engine, damper…), allowing a realistic dynamic behavior of the vehicle. For instance, animation during acceleration or brake phases has been improved, turn trajectories are more realistic and take into account the type of vehicle (buses, trucks and cars do not turn the same way), and local road depressions affect the vehicle.

Informal rules

Dynamic parameters have been added to the application to take into account the behavior of other drivers. First, an aggressiveness parameter is computed during the simulation. Using a short term memory structure, its level is increased if the driver did not reached its desired speed during a sufficient duration in the past. In contrary, when the driver is close to its desired speed (at least 80% of it), the level slowly decreases. A second element is taken into account in the computation: the waiting time at intersections. The aggressiveness increases with it, finally leading to break sign and priority rules

This aggressiveness level is used to dynamically influence the other parameters of the simulation: the maximal speed, the safety time, the speed coefficient... The decision model thus takes into account the behavior of others vehicles: slow vehicles or traffic jams cause aggressiveness increases, which results in changes in the behavior of the agent.

In addition, a specific handling of the road signs is added. As presented above, the signs were only considered using a boolean rule allowing either to respect or bypass them. However, this approach is limited: drivers do not adopt the same behavior when confronted to a red traffic light, a give way sign or a stop sign. These three elements have been distinguished, to increase the variety while producing realistic behaviors. Not only are the drivers allowed to break the rule (i.e. not respect the signalization), but the way they break it is specified: for a stop sign, respecting the rule leads to a full stop, followed by a few seconds wait aimed at observing the oncoming vehicles. Violating drivers use shorter waiting durations, the extreme behavior being to consider a stop sign as a give way one. For give way signs, the parameter influences the speed used to cross the priority road (emergency stops are easier at slow speeds), once a violation has been decided.

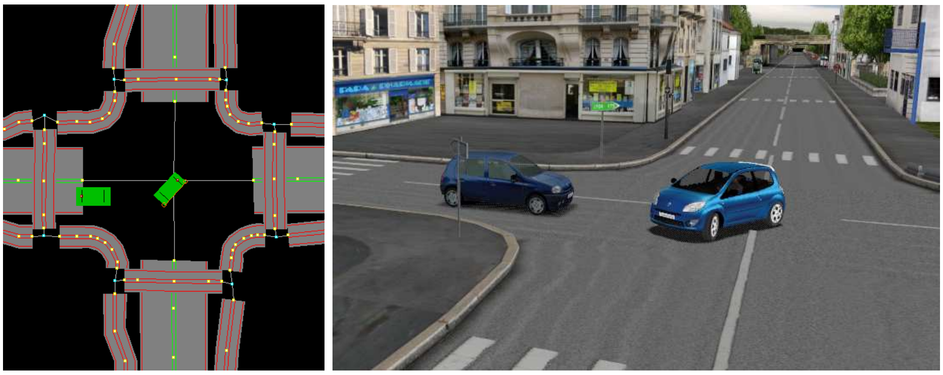

Figure 3: Introduction of informal rules in intersections: a vehicle takes the right of way

Behavioral differentiation

The variety and consistency of the behaviors are important factors for the simulation's realism, and depend on the drivers behavioral parameters presented in the previous section: they constitute the inputs of the traffic decision model. In simulations where the behaviors rely upon many parameters of different kinds (discrete, continuous…), controlling their values and their consistency is a complex issue. The notion of norm, which presents an intuitive description mean of the parameters sets, has been used to answer it (Lacroix, Mathieu & Kemeny, 2008).

In multi-agent based simulations, normative systems are usually used to add regulation possibilities to the environment, and to offer cooperation and coordination possibilities. For instance, Electronic Institutions (Esteva, Padget & Sierra, 2001; Castelfranchi at al., 1999) exploit them to regulate the interactions of the agents: the institution provides authority instances to control their behavior. The two main constitutive elements of the system are the Institution and the Norms. The Institution describes the sets of conventions that govern agents interactions. The norms are used to assess the consequences of their actions within the scope of the institution: for instance, regulative norms are used to associate punishment to certain agents actions. They are applied in various fields: disaster management, monitoring market mechanisms (Michael, Parkes & Pfeffer, 2004)… In the traffic simulation field, non-normative behaviors have been used to improve the behavior of vehicles in intersections (Doniec et al., 2006).

However, norms can be used in a descriptive way, rather than a prescriptive one. In that case, the Institution provides a fixed reference for the norms. It holds a finite set of parameters, associated to a set of definition domains. The link to the context is kept using institutional and environmental properties. A Norm describes a type of behavior. It holds a subset of the institutions parameters, associated to subsets of the definition domains. The norm handles a set of distance functions, which quantify the gap between a value and its domain. They are used to determine potential violations. Finally, a Behavior is the instantiation of a norm. Each parameter of the behavior has a value, taken from the norm's definition domain. If a behavior violates the norm, some of these values are taken outside this domain, but have to remain within the institution ones. Formally:

- an Institution is a tuple (P, D_P, P_i, P_e) where: P is a finite set of parameters; D_P is a set of definition domains; P_i is a set of institutional properties; and P_e is a set of environmental properties,
- a Norm is a tuple (I, P_n, D_{Pn}, Γ_{Pn}, P_{ni}, P_{ne}) where: I is the institution the norm refers to; P_n is the subset of parameters associated to the norm; D_{Pn} is the subset of definition domains; Γ_{Pn} is a set of distance functions; P_{ni} is a set of institutional properties; and P_{ne} is a set of environmental properties,
- a Behavior is a triple (N, P_b, V_{Pb}) where: N is a reference to the instantiated norm; P_b is a subset of the set of parameters defined in the instantiated norm; and V_{Pb} is the set of values associated to the parameters.

The instantiation from norm to behavior is realized using a specific generation engine. It allows controlling the determinism of the generation process: the users can choose between fully reproducible simulations, which are needed in some experiments, and non-reproducible ones, where the objective is to create unexpected or unusual behaviors and to study their influence on the simulation.

Using these elements, variety can be produced in two different ways. Firstly, as many norms as needed may be created by defining different sets of parameters and definitions domains. A wide range of behaviors can thus be described, and the consistency of the resulting behaviors is guarantied by their definitions. Secondly, behaviors violations can be introduced to increase the variety with unexpected ones. However, in that last case, their consistency can no more be guaranteed.

This model can be applied in various contexts: we will see below how it was used on traffic simulation, but its genericity allows using it on any simulation were many parameters have to be controlled at the same time.

Implementation

In SCANeR™, the model has been applied on the existing input parameters of the decision model: they influence the resulting behaviors of the vehicles, and can be easily modified by the users according to their own needs. The Institution is composed of the description of the set of parameters, and various norms can be defined (Table 1).

Different tools have been introduced in the software to manipulate the model. A graphical tool allows the users modifying existing scenarios, by selecting the norm they want to apply on vehicles. A generation tool was also added: traffic generators can be positioned in the database. The traffic flow and proportions of vehicles using each norm is defined for each generator, and the simulation is thus automatically populated with various kinds of vehicles during the execution.

Table 1: Definition domains of the Institution, and two examples of norms.

Parameter	Institution domain	Norm: highway normal driver	Norm: highway aggressive driver
maximal speed (km/h)	[0, 300]	[100, 140]	[140, 160]
safety time (s)	[0, 10]	[1, 3]	[0.1, 1.2]
overtaking risk	[-1, 2]	[-0.5, 0.5]	[1, 2]
speed limit risk	[0, 10]	[0, 1.1]	[1, 10]
observe signs	{true, false}	{true}	{true, false}
observe priority	{true, false}	{true}	{true, false}

Experimental results

In order to evaluate the approach, an experiment was done on a database representing a 11 km long highway. The vehicles were created at the beginning of the section, with a total traffic flow of 3000 veh/h. Each simulation run lasted 2h30. The data were recorded using detectors positioned at kilometer 2.2, 6 and 10.8. Three different sets of norms were used: *no norms*, where the differentiation mechanism was deactivated (the vehicles used only their defaults values); *normal only*, where only the highway normal driver norm was used. The third set of norms, *all norms*, used the normal and aggressive drivers, adding a highway cautious driver norm (slower, do not takes risks). The definition domains are truncated normal distributions using the limits presented in Table 1, the means being the mean of each interval $[L_1,L_2]$ and the variance $\Delta L/4$.

The results are presented in Figure 4. The first graph represents the distribution of speeds at kilometer 6. The measured speed variety increases with the use of norms. When no norm is used, the concentration of values around 80 and 130 km/h is explained by the fact that vehicles all try to reach a similar speed, but cannot take advantage of small changes in the traffic flow (we observe continuous flows on both lanes of the highway). As expected, the travel time decreases when the variety increases, because of the improvement in traffic dynamicity. However, with all the norms, the presence of cautious drivers limits the evolution of aggressive ones, which explains the stability of the results. Finally, the repartition of the vehicles in the different lanes correctly reflects the drivers population: aggressive drivers are mostly present in the left lane, and cautious ones in the right one. The results are similar on all detectors. The introduction of different norms thus improves highly the behaviors variety in the simulations, while guarantying their consistency within the norms definitions.

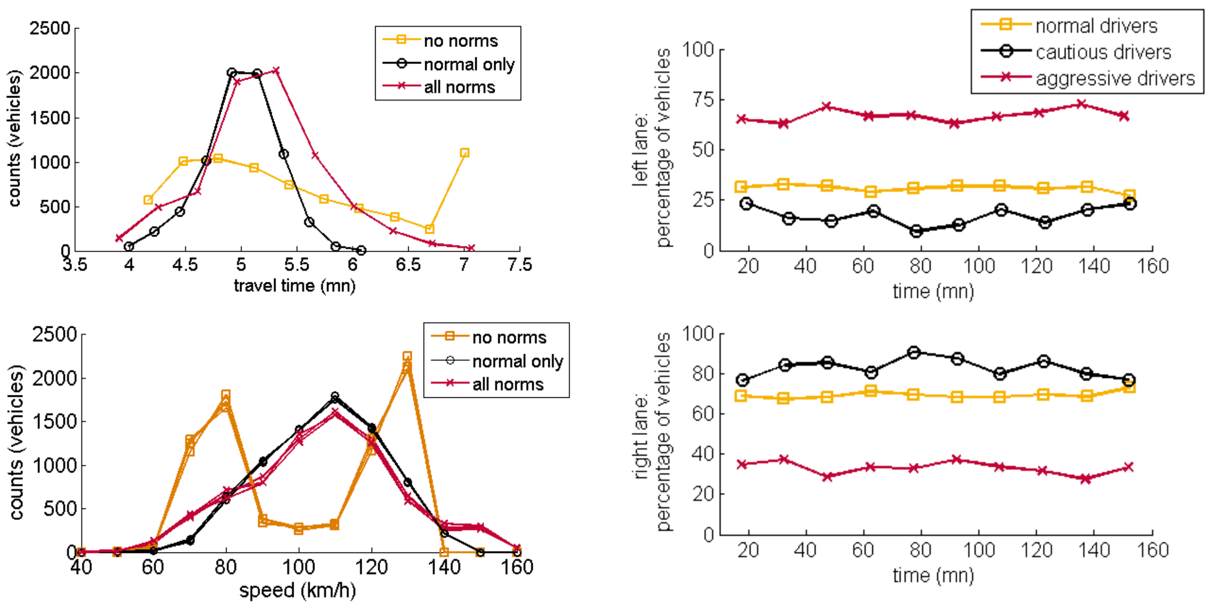

Figure 4: Total travel time, speed distribution and lane repartition at km 6

Different elements might be discussed on this experiment. First, the choice of the norms reflects the usual driving psychologists classification, but the values used have been chosen empirically. An important improvement would be to calibrate the model with real data, which is currently under work. Another point is that we did not exploit here the notion of violation. This will be done in further experiments, to introduce for instance drunk drivers and study their influence on the simulation.

Conclusion

In this paper, we presented the new traffic simulation module of SCANeR™. Dynamic parameters have been added, allowing taking into account the behaviors of other drivers: an aggressiveness parameter influences the models input parameters to act on drivers behavior. The dynamic model of the autonomous vehicles has been improved, and various features of the road are now supported, like road humps or local depressions. The different types of vehicles are specifically computed, including multi-axle vehicles. Finally, the variety and the consistency of agents behaviors were handled with a specific approach. Using a description of the behaviors based on a normative approach, the consistency of their generation can be controlled. The variety is introduced through the definition of the norms, or using violations.

The next steps of this work are to improve the informal rules, to formalize their use in the simulation, and to study the issue of scenario determinism and control when using automatically generated traffic flows. A mechanism allowing controlling the consistency of the behaviors will be added, based on unsupervised learning. It will permit to calibrate automatically the norms from real data.

References

Barcelò, J., Casas, J. (2002), Dynamic network simulation with AIMSUN. *Proceedings of the International Symposium on Transport Simulation*

Björklung, G., Aberg, L. (2005), Driver behaviour in intersections: Formal and informal traffic rules. *Transportation Research Part F*, pp. 239-253

Castelfranchi, C., Dignum, F., Jonker, C., and Treur, J. (1999), Deliberative normative agents: Principles and architecture. *Proceedings of Agent Theories Architectures and Languages*, pp. 206-220

Champion, A., Mandiau, R., Kolski, C., Heidet, A. and Kemeny, A. (1999). Traffic generation with the SCANeR II simulator: towards a multi-agent architecture. *Proceedings of the Driving Simulation Conference*, pp. 311-324

Dewar, R. E. (2002). Individual Differences. *Human Factors in Traffic Safety*, pp. 111-142

Doniec, A., Espié, S., Mandiau, R. and Piechowiak S. (2006), Non-normative behaviour in multi-agent system: Some experiments in traffic simulation. *Proceedings of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, pp. 30-36

Espié, S., Saad, F., Schnetzler, B., Bourlier, F. and Djemane, N. (1994). Microscopic traffic simulation and driver behaviour modelling: The Archisim project. *Proceedings of Road Safety in Europe and Strategic Highway Research Program*, pp. 22-31.

Esteva, M., Padget, J. & Sierra C. (2001), Formalizing a language for institutions and norms. In *Intelligent Agents VIII*, volume 2333 of LNCS, pp. 348-366, Springer.

Lacroix, B., Mathieu, P., Rouelle, V., Chaplier, J., Gallée, G., and Kemeny, A. (2007), Towards traffic generation with individual driver behavior model based vehicles. *Proceedings of the Driving Simulation Conference North America*, pp. 144-154

Lacroix, B., Mathieu, P., and Kemeny, A. (2008), A normative model for behavioral differentiation. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 96-99

Michael, L., Parkes, D., and Pfeffer, A. (2005), Specifying and Monitoring Market Mechanisms Using Rights and Obligations. *Proceedings of AMEC 2004*, LNAI 3435, pp. 188–201

Olstam, J. (2005). Simulation of Rural Road Traffic for Driving Simulators. *Proceedings of the 84th Annual meeting of the Transportation Research Board*

Wang, H., Kearney, J., Cremer, J., and Willemsen, P. (2005). Steering Behaviors for Autonomous Vehicles in Virtual Environments. *Proceedings of the IEEE Virtual Reality Conference*, pp. 155-162.

Wright, S., Ward, N. J. and Cohn, A. G. (2002), Enhanced presence in driving simulators using autonomous traffic with virtual personalities. *Presence*, vol. 11, pp. 578-590