



**HAL**  
open science

# Combination of simulation and model-checking for the analysis of autonomous vehicles' behaviors: A case study

Johan Arcile,, Jérémy Sobieraj, Hanna Klaudel, Guillaume Hutzler

## ► To cite this version:

Johan Arcile,, Jérémy Sobieraj, Hanna Klaudel, Guillaume Hutzler. Combination of simulation and model-checking for the analysis of autonomous vehicles' behaviors: A case study. 15th European Conference on Multi-Agent Systems (EUMAS 2017) and the 5th International Conference on Agreement Technologies (AT 2017), Dec 2017, Evry, France. pp.292–304, 10.1007/978-3-030-01713-2\_21 . hal-01918518

**HAL Id: hal-01918518**

**<https://hal.science/hal-01918518v1>**

Submitted on 12 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Combination of simulation and model-checking for the analysis of autonomous vehicles’ behaviors : a case study

Johan Arcile, Jérémy Sobieraj, Hanna Klaudel, and Guillaume Hutzler

IBISC, Univ Evry, University of Paris-Saclay, 91025, Evry, France  
{johan.arcile, jeremy.sobieraj, hanna.klaudel, guillaume.hutzler}  
@univ-evry.fr  
<https://www.ibisc.fr>

**Abstract.** Autonomous vehicles’ behavioural analysis represents a major challenge in the automotive world. In order to ensure safety and fluidity of driving, various methods are available, in particular, simulation and formal verification. The analysis, however, has to cope with very complex environments depending on many parameters evolving in real time. In this context, none of the aforementioned approaches is fully satisfactory, which lead us to propose a combined methodology in order to point out suspicious behaviours more efficiently. We illustrate this approach by studying a non deterministic scenario involving a vehicle, which has to react to some perilous situation.

**Keywords:** Autonomous vehicles, simulation, verification

## 1 Introduction

Behavioural analysis of autonomous vehicles is a challenge for modellers for years [1–4]. It is often addressed by one of the following approaches:

- A road test consists in testing autonomous vehicles on existing roads or circuits in order to study their behaviour in the various situations they may have to deal with. Despite obvious advantages related to its realism, this method presents however serious limitations. Some countries simply do not allow the use of autonomous vehicles on existing roads, and even when it is legal, expensive prototypes are needed. Moreover, the time spent in testing is long, as it corresponds to the real time spent on the road. Finally, some scenarios cannot be studied with this approach, typically the dangerous ones, which potentially lead to crash. As a result, computer based approaches are generally preferred.
- More specifically, computer simulation enables to model vehicles’ behaviours in a chosen environment so that various kinds of scenarios may be studied in a comparatively shorter time. As compared to the former method however, this implies that the vehicles and their environment are considered at some

level of abstraction. Moreover, when vehicles present non-deterministic behaviours, simulation tools are generally not exhaustive since each simulation corresponds to a single path in the graph of all the possible behaviours;

- To improve the confidence in some model, the technique of model-checking, which is an exhaustive method, may be used. Model-checker analysers allow to automatically obtain binary answers (by yes or no) to questions about the dynamic behaviour of vehicles. To do so, specific models of vehicles in a given environment have to be provided. However, the use of the corresponding tools is generally limited, being quite prone to the well known state space explosion phenomenon.

We aim at proposing a method based on a combination of the latter two approaches in order to benefit from their respective comparative advantages when dealing with non-deterministic cases. The level of abstraction should therefore be carefully chosen so as to guarantee a convincing representation of the actual vehicles's behaviour in computer simulation. Because of the explosion of the state space, the same level of abstraction cannot be used in model checking. Since further abstractions are necessary to perform model checking analyses, it may lead to some gap as compared to computer simulation, so that we need to check that the obtained results are comparatively similar.

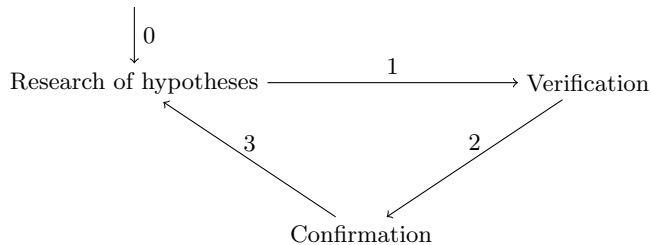
In order to define the desired levels of abstraction, it is necessary to determine the main properties of interest that are to be studied with these tools. Here, we focus on two main properties: the safety, which ensures that a vehicle always respects safety distances and minimises the risk of accidents; the fluidity of traffic, which consists in optimising the speed of each vehicle and in reducing the stop-and-go phenomenon (traffic jams) to lower atmospheric emissions and fuel consumption. The former can be addressed, for example, with *Time-To-Collision* (TTC) [5, 6], which computes the time before two vehicles on the same lane collide if they keep their current speeds. The latter can be addressed, for example, with *Travel Time* [7, 8], which computes the time required to travel a desired distance.

Our method is sketched in Figure 1. The first step (step 0 in the figure) consists in producing data with the help of computer simulation. On the basis of such data, a human expert makes hypotheses about the behaviour of autonomous vehicles for a given initial situation. It could be for example: all vehicles are safe at all times. This kind of query needs an exhaustive check in the case of a non-deterministic evolution of the system. Step 1 consists in expressing the query in temporal logics<sup>1</sup> and model checking it. If the model checker finds some execution leading to a counter example, the corresponding execution is then confirmed by simulation, to ensure that it is not a false positive due to abstractions (step 2). In the case it is confirmed that some counter example invalidates the hypothesis, the process can go on with a new one, thanks to the enrichment of data through the verification process (step 3). More precisely, the method allows us to add

---

<sup>1</sup> The translation of properties into temporal logics can be partly automatized using a predefined set of queries.

particular executions to the data that will give critical informations in order to understand the behaviour of the vehicle and its causes.



**Fig. 1.** Steps of the method.

To illustrate our method, we propose a case study, namely a car-following model in a given situation, which is rather straightforward for computer simulation but where the model checking has to face potential error accumulation due to the necessary discretisation of real number equations. A car following model is a mathematical model defining the vehicle's acceleration with respect to the dynamics of the vehicle that precedes it, such as Gipps [9] or IDM [10].

In particular, we are interested in the following questions:

- What conditions are needed for a vehicle in order to be in an unsafe situation?
- Are there executions where a vehicle is always safe?
- Is it possible to stop the car without overpassing a deceleration threshold?

The structure of the paper is as follows: Section 2 first presents the main elements of the computer simulation and model-checking approaches used in our case study on autonomous vehicles, as well as the corresponding tools. Section 3 is then devoted to the case study: first, a comparison between the simulation and model-checking tools is developed for more confidence in our results. Finally, our methodology is used on a non-deterministic scenario, where we aim at checking hypotheses on a vehicle's behaviour defined by a car-following model.

## 2 Presentation of the tools

### 2.1 Simulation

Simulation is a general method in which one first build a model to represent several aspects of a system (environment, behaviour, interactions, physical phenomena...). The simulation consists in the computational evaluation of the dynamics of the model over time. In the context of vehicles, it allows to reproduce their physics and interactions with the environment.

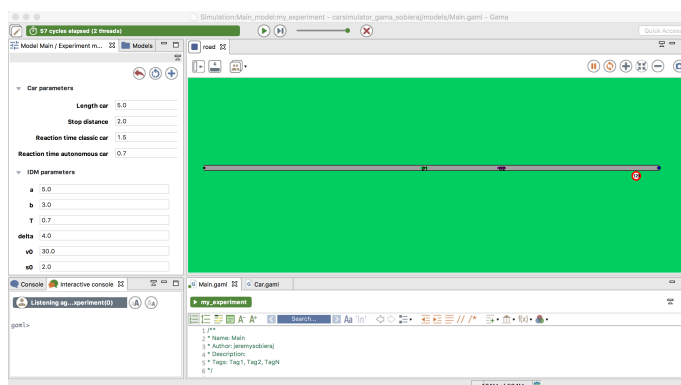
A level of abstraction should be defined in a way approaching the reality according to the desired observation. Various approaches have been developed

for the simulation of autonomous vehicles, among which we could distinguish three main families.

A first approach consists in reproducing realistically the behaviour of vehicles by reproducing perfectly the laws of physics together with their specific parameters (such as inertia, impulse or friction). The simulation can then assess various aspects such as the study of a precise trajectory [11] or, in the context of communicating vehicles, a study of the reliability and integrity of the information transmitted between the vehicles [12].

A second approach focuses on traffic, in general, and its evolution over time through three points of view [13]: macroscopic, microscopic and mesoscopic. In particular, in the case of microscopic studies, where vehicles are handled individually in a small area, this approach allows to represent the longitudinal component of vehicles' motion thanks to car-following models, the majority of which are guaranteed without collision (such as IDM), and the lateral component thanks to lane change models (such as MOBIL [14]).

Finally, the third approach is agent-oriented in the sense that a vehicle is assimilated to an agent, which reacts according to the perception of its environment [15]. Each agent has also the possibility to communicate with other agents or with its environment to exchange information or negotiate a future decision to make.



**Fig. 2.** Study of vehicles' behaviour with GAMA tool

The simulators allow to model and study various properties at different levels of realism and scale. We decided to illustrate our case study using the GAMA tool [16] (Figure 2), which is a multi-agent systems simulator using an agent-oriented programming language (GAML). Each vehicle is characterised by a position, speed and acceleration and is in a two-dimensional environment: the longitudinal dimension (road direction) and the lateral dimension (neighboring lanes). At each time step, each vehicle updates its position while respecting

the rectilinear motion with a uniform acceleration. It can perform the following actions: accelerate, slow down, keep the same speed or change lane in a road consisting in one or several lanes. The proposed abstraction is an effective way to observe complex phenomena between vehicles and is a first step before switching to a high level of realism.

## 2.2 Model-checking

Model checking is a formal method for solving complex decision problems. More specifically, it allows for behavioural properties of a given system to be verified and provides examples of behaviours, which either respect or violate the checked property. A model checking tool can be seen as an operator which uses a model (for example a set of possible behaviours of vehicles on a portion of road) and a property (for example the possibility of a collision between the vehicles) and gives a binary result (true or false). This generally requires to model the system as a finite-state machine along with the formalisation of a behavioural property in temporal logics. The result of the checking is then obtained through the automated inspection of all states of the model, meaning that all the possible futures from a given initial situation are considered in order to assess the property. The main asset of such method is that it handles non-determinism while guaranteeing exhaustivity. But as a drawback, getting an answer may be difficult due to the number of states of the model. Indeed, as a formalisation of the system's characteristics is needed, the resulting model is often composed of a very large number of states (often several billions).

For instance, in the case of modelling of vehicles on a road portion, one must take into consideration:

- The variables needed to express the state of a vehicle (position, speed, acceleration, direction, . . .);
- The cost of non-deterministic decision making;
- The number of vehicles, whose growth exponentially increases the number of states.

Using abstractions allows to deal with the state explosion problem, while impacting the reliability of the system due to the necessary discretisation. Actually, accumulation of errors due to discretisation and finite number computation may lead to a totally different behaviour than the expected one. Having said that, model checking may be interesting in solving complex problems, but the model must present a satisfactory compromise between realism and computability.

In our combined analysis, we use a slightly modified version (without communication or lateral movement) of an existing model initially devoted to assess robustness of autonomous vehicles [17]. It runs on the model-checker UPPAAL [18, 19], which allows the verification of properties expressed in a subset of CTL (computational tree logic) [20]. The expressivity of this query language is generally sufficient for the kind of information we want to obtain. The model considers a road section composed of several unidirectional lanes with several agents (vehicles) on it. This environment is represented by a data structure containing the

state of each vehicle along with a set of constraints on their possible actions. A vehicle's state is kept as a set of values including position, speed, direction, knowledge on the environment, etc. The current position is expressed using discrete values, but with enough precision to model the vehicle's progression without leading to an abnormal behaviour due to the loss of information. More precisely, the position of a vehicle is considered as a point on a two-dimensional orthogonal grid. At a given frequency, each vehicle uses the information at its disposal to make a decision on the immediate action to be performed (it acts on acceleration and direction). It performs an action at its own frequency, i.e., vehicles are seen as independent agents. Vehicles' speeds and positions are updated in a simultaneous way, which means that the observation of the system is independent from the vehicles' decision making process frequency.

In the case study, we assume that each vehicle knows the exact position and speed of the vehicle that precedes it anytime a decision is made, meaning that the information from sensors is considered perfect. However, the challenge regarding this case study is to express the continuous function of a car-following equation using a discrete setting of model checking tools, while reaching a satisfactory precision and a state space small enough to be analysed in a reasonable time. In order to do so we scaled variables before division, implemented rounding, power and square root functions (which are not supported by UPPAAL because usually not needed in model-checking), and wisely chose the granularity of the variables used in the model. We obtained a model satisfying precision requirements that may be analysed by a model checker in (at most) a couple of minutes for a few vehicles on a portion of road of a few hundred meters.

### 3 Study of a car-following model : Intelligent Driver Model (IDM)

In this section we first present the Intelligent Driver Model (IDM, a decision making algorithm for autonomous vehicles) [10] and the interesting indicators for behavioural analysis. First, for more confidence in our results, a comparison between the simulation and model-checking tools is developed in order to find indicators not impacted by the used abstractions. Then, we use our methodology to check properties of IDM on some non-deterministic scenario.

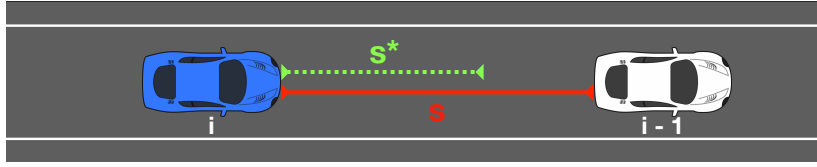
#### 3.1 Presentation

For our case study, we chose the Intelligent Driver Model (IDM) in order to observe suspicious situations by combining simulation and model-checking tools. This choice has several advantages:

- IDM allows to determine the acceleration of the follower vehicle for a given situation by observing the characteristics of the follower and leader vehicles;
- It can describe an autonomous vehicle using an Adaptive Cruise Control (ACC) system but it can also simulate the behavior of a human-driven vehicle;

- This variety of representations can be done through the ability to determine values for a set of initial parameters.

Let us consider two vehicles as depicted in Figure 3: the follower vehicle  $i$  (whose behaviour is determined by IDM) and the leader vehicle  $i - 1$  (whose behaviour is not necessarily IDM). At each time step, vehicle  $i$  updates its acceleration, which varies according to two main criteria: on the one hand, vehicle  $i$  tends to reach a maximum speed allowed on the road portion,  $v_0$  (cruise speed), and on the other hand, it must also respect a minimum safety distance  $s^*$  with the leading vehicle (which varies with the relative speed between the two vehicles).



**Fig. 3.** IDM operation: at any time, vehicle  $i$  adapts its speed with respect to vehicle  $i - 1$

The IDM equation computing the acceleration of the follower vehicle is as follows:

$$acc_i = \frac{dv_i}{dt} = a \left[ 1 - \left( \frac{v_i}{v_0} \right)^\delta - \left( \frac{s^*(v_i, \Delta v_i)}{s} \right)^2 \right] \quad (1)$$

where the desired bumper-to-bumper distance  $s^*(v_i, \Delta v_i)$  is:

$$s^*(v_i, \Delta v_i) = s_0 + \max \left[ 0, \left( v_i \times T + \frac{v_i \times \Delta v_i}{2\sqrt{ab}} \right) \right] \quad (2)$$

with  $a$  being the maximum acceleration,  $v_i$  the speed of vehicle  $i$ ,  $v_0$  the maximum allowed speed for vehicle  $i$ ,  $\delta$  the acceleration exponent ("aggressiveness" coefficient),  $\Delta v_i = v_i - v_{i-1}$  the relative speed,  $s_0$  the minimum bumper-to-bumper distance to the leading vehicle,  $T$  the estimated reaction time and  $b$  the desired deceleration.

The parameters, namely  $a$ ,  $v_0$ ,  $\delta$ ,  $s_0$ ,  $T$  and  $b$ , should be fixed at the initialisation. For all the vehicles controlled with IDM, we decided to use the following values corresponding to a recent autonomous vehicle on a highway offering a comfortable deceleration and with a slightly shorter reaction time than that of a human driver [10]:

- $a : 5.0m.s^{-2}$
- $v_0 : 30.0m.s^{-1}$  (less than  $110km.h^{-1}$ );



- $\delta : 4$ ;
- $s_0 : 2m$ ;
- $T : 0.7s$ ;
- $b : 3m.s^{-2}$ .

Note that in the IDM equation there is no limit on the maximum deceleration (only the desired deceleration  $b$  is fixed). This means that there are potentially situations where the value of deceleration exceeds  $b$ .

In what follows, we will use Time-to-Collision indicator TTC to study safety and therefore potential cases of abnormally high deceleration value involving unsafety or discomfort.  $TTC_i$  value depends on the speed and position of the two vehicles:

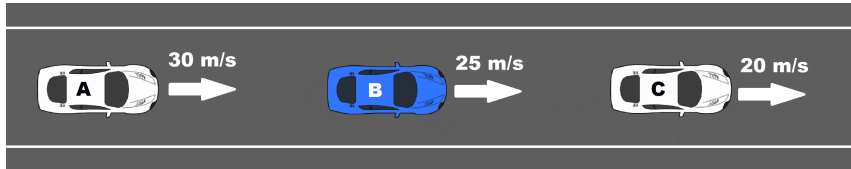
$$TTC_i = \frac{x_i(t) - x_{i-1}(t) - l(i)}{v_i(t) - v_{i-1}(t)} \quad \forall v_i(t) > v_{i-1}(t), \quad (3)$$

where  $i - 1$  represents the leader,  $i$  the follower,  $v_i$  the speed of  $i$ ,  $x_i$  the position of  $i$  and  $l_i$  the length of the vehicle  $i$ .

For all the scenarios presented in what follows, the observed portion of the road is 200 meters long, each vehicle's length is exactly 5 meters and their decision making process occurs every 100 milliseconds.

### 3.2 Comparison

We first want to check the difference in behaviour between computer simulation and model checking approaches by comparing the travel time at the end of the road portion and the position at a given time (three seconds after the beginning of the scenario). Scenario 1 features three vehicles controlled by IDM. Initially, vehicle A is at position  $0m$  and its speed is  $30m.s^{-1}$ , vehicle B is at position  $50m$  and its speed is  $25m.s^{-1}$ , and vehicle C is at position  $100m$  and its speed is  $20m.s^{-1}$ . Figure 4 illustrates this scenario.



**Fig. 4.** Initial situation of Scenario 1 with the respective speeds of each vehicle.

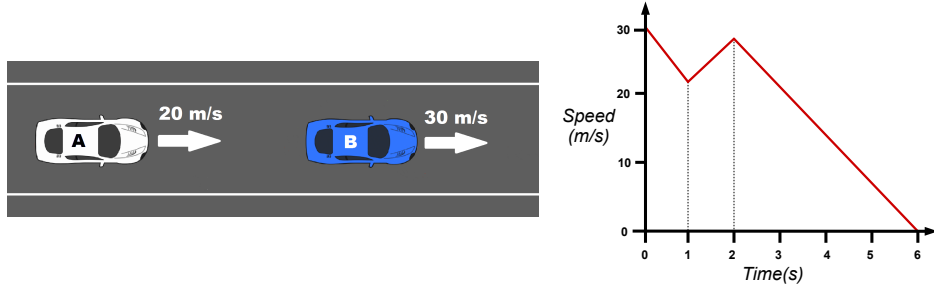
Here, the vehicles A and B have to adapt their speed to avoid collision, which seems to be a good example to detect a possible error accumulation. The results are presented in Table 1 and show that the position and travel time of the vehicles are very close, with the higher percentage of error being on the travel time of B

(0.52 %). Such results show that the average behaviour obtained for both tools is similar enough to use position and travel time as reliable indicators.

Criterion	Travel time			Position at 3 s		
	Car A	Car B	Car C	Car A	Car B	Car C
Simulation	7.38 s	5.68 s	3.92 s	80.51 m	125.69 m	174.03 m
Model checking	7.39 s	5.71 s	3.92 s	80.74 m	125.26 m	173.99 m

**Table 1.** Comparison of travel time and position at a given time for the three vehicles of Scenario 1.

Next, we check the behaviour regarding more sensitive indicators such as TTC or acceleration value. Scenario 2 features two vehicles: vehicle A controlled by IDM, initially at position  $0m$  with speed of  $20m.s^{-1}$ , and vehicle B initially at position  $50m$  with speed of  $30m.s^{-1}$ , controlled with the following rule: B starts by decelerating at  $-7m.s^{-2}$  for one second, then accelerates at  $5m.s^{-2}$  for one second, and finally decelerates again at  $-7m.s^{-2}$  until it stops, as depicted in Figure 5.



**Fig. 5.** Initial situation of Scenario 2 along with the evolution of vehicle B's speed.

One may notice that the initial situation is safe for the vehicle A (i.e., it respects the safety distance). On this scenario, we check the values and time of appearance of both the worst possible TTC value and the minimum acceleration value (i.e., the maximum deceleration) for vehicle A. From the same vehicle we also check the moment its acceleration value becomes negative.

Criterion	min TTC (value)	min TTC (time)	min acceleration (value)	min acceleration (time)	first deceleration (time)
Simulation	1.78 s	6.0 s	-7.36 m.s <sup>-2</sup>	6.0 s	2.70 s
Model checking	1.76 s	6.0 s	-9.40 m.s <sup>-2</sup>	6.0 s	2.70 s

**Table 2.** Comparison with respect to criteria based on acceleration and TTC of vehicle A's behaviour in Scenario 2.

The results are presented in Table 2 and show that for both tools, all the events occur in the same time units. Also, TTC values are very close with only 0.02 s of difference (1.13 % of error), meaning that we can use TTC as an indicator with a reasonable confidence. On the other hand, the value of minimum acceleration is quite different with more than  $2m.s^{-2}$  of difference giving 27.71 % of error. This is due to the propagation of errors of discrete value computation when making decision with the IDM controller. Due to a different level of abstraction, the acceleration computed at some step in the model checker is slightly different than the one obtained in simulation. If the acceleration computed by IDM equation with discrete domain of the model checker is greater than the real one, it implies that the speed at next step will be greater than needed and the time to collision lower than it should be. As a consequence, the next acceleration computed with IDM will be lower than the real one as it compensates this difference. One may therefore observe locally important differences on acceleration, even if the average values are close.

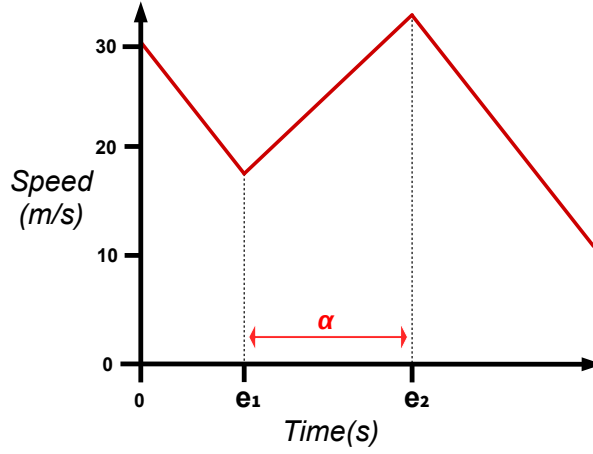
### 3.3 Application of the method

In this section, we create a non-deterministic example on which we apply our method. We define scenario 3 as a non deterministic variant of scenario 2, where vehicle B starts with a  $-7m.s^{-2}$  acceleration, at some time changes this value to  $5m.s^{-2}$ , and then at some time changes it back to  $-7m.s^{-2}$ . We call the time of the first event  $e_1$  and those of the second  $e_2$ , and define  $\alpha = e_2 - e_1$  as the duration of vehicle B's positive acceleration. This values are illustrated in Figure 6. Note that if  $e_2$  never happend before the vehicle is out of the observed portion of road, we consider that  $\alpha$  is infinite.

First, we run several random simulation executions and get both minimal TTC and acceleration values. On the basis of this data, we make three hypotheses:

1. There is a duration  $\gamma$  such that if  $e_1 \leq \gamma$ , TTC will never be under 1.7 s,
2. In order to have a TTC under 1.7 s,  $\alpha$  must be comprised between  $e_1 * 0.5$  and  $e_1$ ,
3. There is no possible execution where vehicle B stops and its acceleration value is never less than to two times the desired deceleration parameter  $b$  (i.e. never less than  $-6m.s^{-2}$ )

This leads to the following temporal logics queries:



**Fig. 6.** Values of  $e_1$ ,  $e_2$  and  $\alpha$  on a possible behaviour of vehicle B.

1. Not Exists Finally  $TTC < 1.7$  and  $e_1 \leq \gamma$
2. Always Globally ( $TTC < 1.7 \Rightarrow \alpha < e_1$  and  $\alpha > e_1/2$ )
3. Not Exists Globally  $acceleration(A) \geq -6$  and  $on\_the\_road(A)$ , where  $acceleration$  is the value of A's acceleration and  $on\_the\_road(A)$  is a Boolean variable, which is *true* if the vehicle A has not yet reached the end of the portion of the road.

The first hypothesis is easily confirmed, but it is more interesting to find a maximum for  $\gamma$ . Using model checker, we find by dichotomy a value of 2.3 s, however one must keep in mind that there may be a slight error in the computed TTC. Then, we explore the neighbourhood of this value by simulation in order to refine it. Simulation points out an execution with  $e_1 = 2.2s$ , where a TTC under 1.7 s is found, but none at  $e_1 = 2.1s$ . We can therefore assume that the actual maximum for  $\gamma$  is 2.1.

The second hypothesis appears to be wrong. Actually, the model checker finds an execution violating such property for both the upper bound (with  $e_1 = 2.4$  and  $e_1 = 4.9$ ) and the lower bound (with  $e_1 = 3.3$  and  $e_1 = 4.4$ ). We then refine the upper and lower bounds and finally get a result, where  $\alpha \in [e_1 * 0.3, e_1 * 1.1]$ . The exploration of these bounds by simulation does not show any counter example.

Finally, the third query gives a positive result, but we have to remember that, as we showed, the acceleration value may potentially be very different in the model checking and simulation. The verification of the same query with a value of  $-7m.s^{-2}$  instead of  $-6m.s^{-2}$  still gives a positive result adding some confidence to the result.

The verification process took less than 10 s for each query despite the complexity of the system due to the size of the variable ranges and a large number of positions in which each the agents may be. However, the system is still not

so complex in terms of non determinism, as we limited the number of possible actions the leader vehicle may perform at any given time. This choice was made for the purpose of the case study, as it was easier to check the consistency of our result on such a case. For this reason one might argue that it might have been possible to check all the possible behaviours with simulation alone in a relatively reasonable time, which in this particular case, might indeed work. However, in case of more complex behaviours, this does not seem to be a reasonable method. Also, despite the fact that the verification time also increases with the size and complexity of the system, it is still possible, up to some extent, to exhaustively check hypotheses on complex non deterministic systems in a reasonable time (i.e., less than a few hours). Note that performance mainly depends on the level of non determinism. For instance, adding other deterministic vehicles on a scenario will have a very low cost, whereas adding more possible behaviours may seriously affect the performance.

## 4 Conclusion

We addressed the challenge of modelling autonomous vehicles' behaviour with both computer simulation and model checking through well chosen abstractions and discretisation. Both modelling approaches were studied and gave a satisfactory representation of reality. First, we succeeded to model with a good accuracy the behaviour of vehicles whose description needed a large number of variables. Second, we experimented and compared both in regards to suitable indicators which showed that obtained values coincided. Motivated by the complementarity of these techniques we proposed a method combining both of them in order to increase confidence in the results. Finally, thanks to this modelling and such a methodology, we provided a case study showing it was possible to obtain efficiently useful information on autonomous vehicles' behaviour. In particular, the analysis of the car-following model IDM pointed out some non trivial behaviours.

The tools used in this paper support more complex environments than those used in our case study. These features may be used to deal with more realistic situations (several lanes, communication between vehicles, various decision algorithms, . . .). Also, to cope with the error due to discrete values' computation, it should be possible to use finer granularity for the variables but at a cost of increased verification time. In the future we plan to use the infrastructure presented here to study communications protocols between agents in order to improve the quality of decision making of autonomous vehicles.

## References

1. B. Y. Ekren and S. S. Heragu, "Simulation based performance analysis of an autonomous vehicle storage and retrieval system," *Simulation Modelling Practice and Theory*, vol. 19, no. 7, pp. 1640 – 1650, 2011.
2. H. Dia, "An agent-based approach to modelling driver route choice behaviour under the influence of real-time information," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5, pp. 331 – 349, 2002.

3. T. Shamir, "How should an autonomous vehicle overtake a slower moving vehicle: design and analysis of an optimal trajectory," *IEEE Transactions on Automatic Control*, vol. 49, pp. 607–610, April 2004.
4. "Hybrid-state driver/vehicle modelling, estimation and prediction," in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 806–811, Sept 2010.
5. K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 427 – 433, 2003.
6. M. M. Minderhoud and P. H. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89 – 97, 2001.
7. J. van Lint, S. Hoogendoorn, and H. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 347 – 369, 2005.
8. G.-L. Chang and H. S. Mahmassani, "Travel time prediction and departure time adjustment behavior dynamics in a congested traffic system," *Transportation Research Part B: Methodological*, vol. 22, no. 3, pp. 217 – 232, 1988.
9. P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B*, vol. 15, no. 2, pp. 105–111, 1981.
10. M. Treiber and A. Kesting, "Elementary Car-Following Models," in *Traffic Flow Dynamics*, pp. 157–180, 2013.
11. S. Zhang, W. Deng, Q. Zhao, H. Sun, and B. Litkouhi, "Dynamic trajectory planning for vehicle autonomous driving," in *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, pp. 4161–4166, 2013.
12. F. Bai and H. Krishnan, "Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications," *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 355–362, 2006.
13. M. Treiber and A. Kesting, *Traffic Flow Dynamics*. 2013.
14. A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models," *Transportation Research Record: Journal of Transportation Research Board*, vol. 1999, no. 1, pp. 86–94, 2007.
15. J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, vol. 222. 1999.
16. P. Taillandier, D. A. Vo, E. Amouroux, and A. Drogoul, "GAMA: A simulation platform that integrates geographical information data, agent-based modeling and multi-scale control," in *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7057 LNAI, pp. 242–258, 2012.
17. J. Arcile, R. R. Devillers, H. Klaudel, W. Klaudel, and B. Wozna-Szczesniak, "Modeling and checking robustness of communicating autonomous vehicles," in *Distributed Computing and Artificial Intelligence, 14th International Conference, DCAI 2017, Porto, Portugal, 21-23 June, 2017*, pp. 173–180, 2017.
18. "Uppaal." <http://www.uppaal.org/>.
19. K. G. Larsen, P. Pettersson, and W. Yi, "Uppaal in a nutshell," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 1, pp. 134–152, Oct 1997.
20. E. A. Emerson and J. Y. Halpern, "'sometimes' and 'not never' revisited: on branching versus linear time temporal logic," *J. ACM*, vol. 1, no. 33, pp. 151–178, 1986.