Statistical Model Checking Applied on Perception and Decision-making Systems for Autonomous Driving.

Jean Quilbeuf[1], Mathieu Barbier[2,3], Lukas Rummelhard[4], Christian Laugier[2],
Axel Legay[1], Blanche Baudouin[2], Thomas Genevois[2], Javier Ibañez-Guzmán[3], and Olivier Simonin[2,5]

[1]Univ Rennes, Inria, F-35000, RENNES, France., Email: name.surname@inria.fr
[2]Univ. Grenoble Alpes, Inria, Chroma, F38000 Grenoble. Email: name.surname@inria.fr
[3]Renault S.A.S, 1 av. du Golf, 78288 Guyancourt, France. Email: name.surname@renault.com
[4]Univ. Grenoble Alpes, CEA, LETI, DSYS, LSOSP, F38000 Grenoble.
[5]INSA Lyon, CITI Lab., 6 avenue des Arts, 69680 Villeurbanne, France. Email: name.surname@inria.fr

***Abstract*—Automotive systems must undergo a strict process of validation before their release on commercial vehicles. The currently-used methods are not adapted to latest autonomous systems, which increasingly use probabilistic approaches. Furthermore, real life validation, when even possible, often imply costs which can be obstructive. New methods for validation and testing are necessary.**

In this paper, we propose a generic method to evaluate complex automotive-oriented systems for automation (perception, decision-making, etc.). The method is based on Statistical Model Checking (SMC), using specifically defined Key Performance Indicators (KPIs), as temporal properties depending on a set of identified metrics. By feeding the values of these metrics during a large number of simulations, and the properties representing the KPIs to our statistical model checker, we evaluate the probability to meet the KPIs. We applied this method to two different subsystems of an autonomous vehicles: a perception system (CMCDOT framework) and a decision-making system. An overview of the two system is given to understand related validation challenges. We show that the methodology is suited to efficiently evaluate some critical properties of automotive systems, but also their limitations.

I. Introduction

In the automotive industry the development and testing of human centric-systems must follow the guidelines of the ISO26262. In the automotive industry, this kind of testing can be divided in two:

- Vehicle-in-the-loop platform tests interactions between a human and the system in dangerous situation [1].
- Hardware-in-the-loop to test interactions between an embedded system, such as the Active Brake Control Systems [2], and the physics of a vehicle.

For autonomous functionality higher than the level 3 as define by the SAE, drivers will not be responsible of most of driving decisions. As these systems will rely on machine learning and probabilistic methods, conventional methods for validation are not adapted. The vehicle shall operate in a various range of scenarios as well as dangerous situations, the validation and verification operations must be carried on simulations perform. It allows to reduce cost and increase the coverage of system testing.

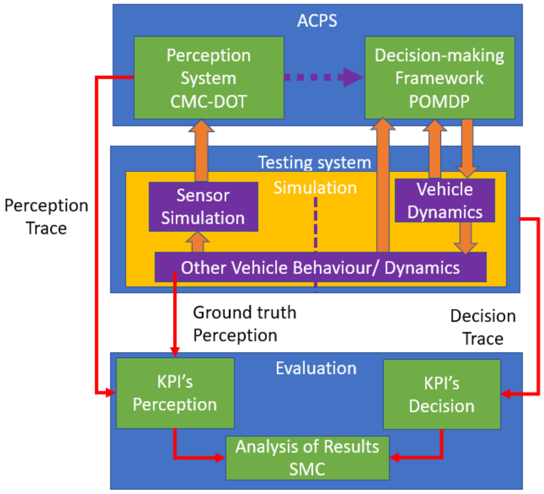

Figure 1. Interactions between the different elements of the proposed Validation pipeline. Dashed line represents future developments to connect the decision-and perception

The difficulty for the validation of an autonomous system are two-fold. First, the complexity and variety scenarios that autonomous vehicle will face is larger than in Advanced driver assistance systems (ADAS). Second, multiple systems will be in constant interaction. In this study we focus on a use-case that highlights these two difficulties: road intersection crossing. It is one of the most dangerous part of the road network with more than 8% of the total road fatalities in Europe [3]. Furthermore, there exists many variations of each scenario (number of vehicles, initial velocities, etc.). It is challenging for the perception because of the limited view range, partially-observed vehicles and because of the presence of multiple vulnerable users that could be potentially at risk. For the decision-making, the interactions between road users are complex to consider because of wrong behaviour of other drivers. Road intersection crossing has been identified as one use-case addressed in the Enable-3 European project [4]. This industry-driven project aspire to propose methods for validation and verification of automated cyber-physical systems (ACPS). The global architecture for validation and verification has been simplified to match our thematic and is illustrated in figure 1.

Examples of validation for highly-autonomous systems can be found in the aerospace domain [5], where formal methods are used to validate the behavior of a fleet of satellites. In the robotic domain, benchmarks allow researchers to compare their results in the same conditions [6], [7]. However benchmarks are often tailored for one specific kind of problem and are not representative enough of the variety of situation that an autonomous system may encounter to actually validate such a system. Waymo was recently confident enough in their system to remove the safety drivers for some tests. This was possible with an effort of 1 billion kilometers driven in a simulated environment [8]. Another way is to use formal methods to ensure the safety of the vehicle [9] but it would rather complex to do in uncertain environment.

The purpose of the paper is to demonstrate the use of a validation method (SMC) on two different systems that have been previously developed, namely Perception and Decision. The requirements for the testing in simulated environment are discussed for each system. Preliminary results for the decision-making system are presented as well as discussions on the challenges caused by the perception system.

Section II presents our validation approach based on statistical model checking. Section III describes the application of our approach on the perception system and the difficulty to find applicable metrics for its validation. Then Section IV shows a more complete application and interprets the results for the decision-making system.

II. Statistical Model Checking

In the context of ACPS, it is not possible to afford validation through exhaustive techniques, that is by stating a property and checking that it holds in all reachable states. Indeed, this would require to model and traverse all the reachable states of the ACPS. Such a modelling is possible at a very abstract level, but requires a huge effort to be brought at a more detailed level. Furthermore, even if a very detailed model of the ACPS were provided, exploring all its reachable states would not be possible due to the very large state space. Stochastic algorithm are complex to validate with conventional methods, thus it is interesting to use probabilistic methods to evaluate them [10].

Statistical Model Checking (SMC) [11], [12] provides an intermediate between test and exhaustive verification by relying on statistics. In order to perform SMC, one needs an executable model and a property to check. The executable model is expected to be stochastic, that is, to have some of its transition governed by probabilistic choices. Note that most ACPS simulations are already modelled as stochastic processes, because variations in the scenario are defined by probability distributions. The property to check must be decidable on a finite trace.

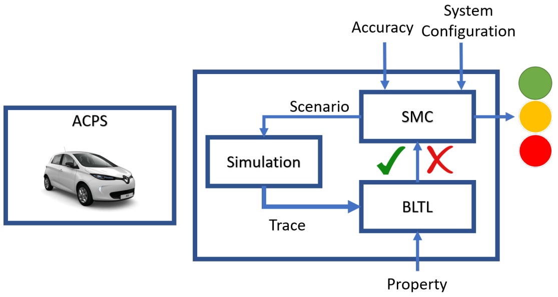

Figure 2. An overview of SMC

The execution being stochastic, some traces will satisfy the property to check and some other will not. Therefore, we can define the probability that a trace satisfies a property. The main goal of SMC is to evaluate that probability. Note that a probability of satisfying a formula gives actually more information than a yes-or-no answer. Indeed, if the model does not satisfy the formula, there is an evaluation of how well it performs.

In order to perform SMC, one needs to be able to

- Generate traces of the execution of the system to validate. These traces have to be generated according to the probabilities in the model.
- Write the property to check as a formula that can be decided on a finite trace, and a procedure for deciding whether a trace satisfy the property.

We present in Figure 2 an overview of the approach. On the left we have a simulator that provides stochastic execution of our system. On the bottom we have the property φ to check. On the top, we have some configuration for the SMC algorithm, such as the required accuracy. The SMC algorithm requires some simulations to the simulator. In turn the simulator provides a trace σ that is fed to the property checker. Finally the property checker returns its verdict to the SMC algorithm. At this point, if the SMC algorithm has enough information to return a result that meets the required accuracy, it does so. Otherwise, it asks for an additional simulation and the loop is run again. We give an intuition of SMC by illustrating it with the Monte-Carlo Algorithm. This algorithm estimates the probability p that a system satisfies a property P by checking P against a set of N random executions of the system. The estimation $\hat{p}$ is given by

$$\hat{p} = \frac{1}{N}\sum_{1}^{N} f(ex_i) \quad where f(ex_i) = \begin{cases} 1 & \text{if} ex_i \models P \\ 0 & \text{otherwise} \end{cases}$$

Using the formal semantics of the property language, the property is checked against each execution trace. The trace must be long enough to decide whether the property holds.

Of course, the larger is the set of simulations, the more precise is the result. The confidence bounds of the estimation are set by two positive real parameters ϵ and δ. The confidence is defined by the Chernoff bound that is stated as:

$$Pr(|p - \hat{p}| \leq \epsilon) \geq 1 - \delta$$

Assuming that p is value of the probability we want to evaluate and $\hat{p}$ is the estimation we compute, the formula means that the estimation error, i.e. the distance $|p - \hat{p}|$, is bound by ϵ with a probability $1 - \delta$. In other words, the probability that the error in the estimation is greater that ϵ is δ. Once δ and ϵ have been set, we can compute the number of simulations N necessary to enforce the above formula. The quality of the approximation is high (and thus N is high as well) when ϵ and δ are close to 0. When ϵ and δ increase, the estimation is more approximate but requires less simulations to be computed.

A. *Defining KPIs*

In order to define and evaluate KPIs based on a set of simulations, we proceed as follows. We first identify with peoples in charge of developing the system some KPIs related to system under test and scenarios. We then express the KPIs as temporal formulas involving the identified metrics. Temporal formulas allow a finer formulation of KPIs by taking into account the evolution of the metrics during time. Let us consider acceleration as a metric. A rough formulation of a KPI concerning acceleration might be that the acceleration should be bounded, i.e. to guarantee the comfort of the passengers [13]. A finer formulation could be that the acceleration should generally be bounded, but the bound can be exceeded for a short period of time.

In order to express such formulas, we rely on BLTL, a bounded version of LTL [14]. The syntax of BLTL is as follows: $\phi \ ::= p \mid \phi \vee \phi \mid \neg\phi \mid \phi \, U_{\leq t} \, \phi \mid X_{\leq t} \, \phi$. A BLTL formula is expressed with respect to a trace. In our case a state is a sequence of states, one for each simulation step. Each state contains the value of each of the metrics at that current state. The symbol p represents a predicate expressed on the current state, for instance a comparison between a metric and a bound. The disjunction ($\vee$) and the negation ($\neg$) defined as usual. Finally, the temporal operators until (U) and next (X) define properties about the time. Since we need to be able to decide whether a property holds on a finite trace, these operators are parameterized by a time bound $t \in \mathbb{R}$. The formula $X_{\leq t}\phi$ is true if ϕ is true in the state reached after t units of time from the current state. The formula $\phi_1 \, U_{\leq t} \, \phi_2$ is true if 1) the formula ϕ_2 becomes true before t units of time from the current state and 2) the formula ϕ_1 remains true in every state before the one where ϕ_2 becomes true. For a formal definition of BLTL semantics, see [15].

In practice, we often use the *always* (G) and *eventually* (F) operators. Eventually is defined as $F_{\leq t}\phi = \texttt{true} \, U_{\leq t} \, \phi$ and means that the formula ϕ should become true before t units of time happen. Always is defined as $G_{\leq t}\phi = \neg F_{\leq t} \neg\phi$ and means that ϕ must always hold for the next t units of time.

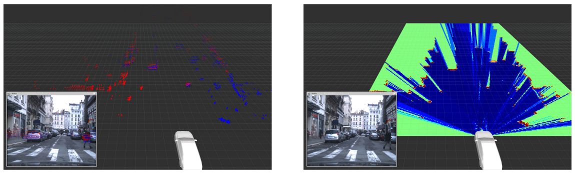

Figure 3. Data fusion in an occupancy grid. Data from each of the 2 LiDARs are used to generate occupancy grids using sensor models, which are then fused by Bayesian fusion.

III. A first validation application: CMCDOT Perception system

A. *Principle of the CMCDOT*

The CMCDOT Framework is a perception system, based on environment representation through probabilistic occupancy grids, a dense and generic representation [16], [17], and Bayesian fusion, filtering and inference.

This type of Bayesian formalism [18] allows proper confidence estimation and combination, particularly important features when confronted with incomplete or even contradictory data coming from different sensors. A major feature of the system is its highly-parallelized design: from data fusion, to grid filtering, velocity inference and collision risk assessment, the methods have been designed to allow massive parallelization of computations, and so benefit from parallel-computing devices [19], allowing real-time performances on embedded devices.

Sensor data is converted to occupancy estimation using specific sensor model, sensor occupancy estimates are then combined by Bayesian fusion in every grid cell (Fig. 3). The Conditional Monte Carlo Dense Occupancy Tracker (CMCDOT) [20] itself is a generic spatial occupancy tracker, which then infers dynamics of the scene through a hybrid representation of the environment consisting of static and dynamic occupancy, empty spaces and unknown areas(Fig. 4). This differentiation enables the use of state-specific models (classic occupancy grids for motionless components and sets of moving particles for dynamic occupancy), as well as relevant confidence estimation and management of data-less areas. The approach leads to a compact model that dramatically improves the accuracy of the results and the global efficiency in comparison to previous approaches.

This method is particularly suitable for heterogeneous sensor data fusion (camera, lidars, radars etc...). The occupancy of each cell over time can be estimated from various sensors data whose specific uncertainty (noise, measurement errors) are taken into consideration. Filtered cell estimates are thus much more robust, leading to a more reliable global occupancy of the environment, reducing false detections.

While most of risk estimation methods consist in detecting and tracking dynamic objects in the scene [21], [22], the risk being then estimated through a Time to Collision (TTC) approach by projecting object trajectories to the future [23], [24], the grid-based approach used in the CMCDOT framework[20] instead directly computes estimations of the

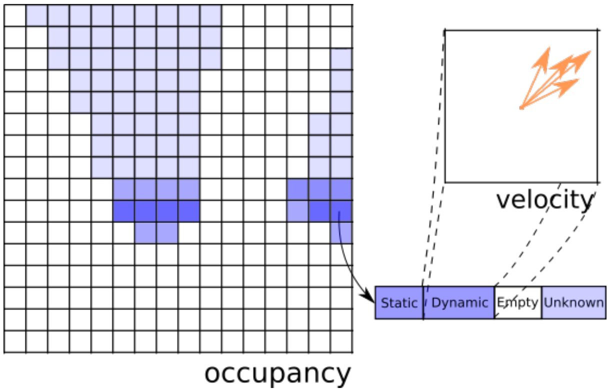

Figure 4. Data representation in the CMCDOT formulation. The environment is divided into cells, to which are associated static, dynamic, empty and unknown coefficients. The dynamic part is allotted to weighted particles which sample the velocity space

position in the near future of every static and dynamic part of the grid, as well as the trajectory of the vehicle. These estimations are iteratively computed over short time periods, until a potential collision is detected, in which case a TTC is associated to the cell from which the colliding element came from (Fig. 5). In every cell, the associated TTCs are cumulated over different time periods (1, 2, 3 seconds for example) to estimate a cell-specific collision risk profile. Risk grids, and global aggregated risks, are thus generated, and later used to generate response impulses for the control system. This strategy[25] avoids solving the complex problem of multi-object detection and tracking, while integrating the totality of the available information. It provides a probabilistic estimation of the risk associated to each part of the scene.

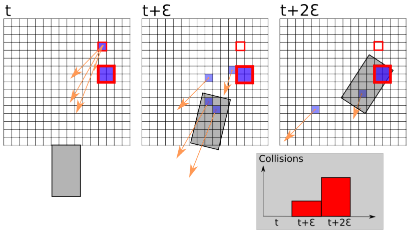

Figure 5. Collision risk estimation over time for a specific cell. The cell position is predicted according to its velocity, along with the mobile robot. This risk profile is computed for every cell, and then used to integrate over time the global collision risk.

B. Method Application

1) Simulation for perception: In this project, the simulation relies on the use of two frameworks: Gazebo and ROS. Gazebo allows for the representation and simulation of the environment, the ego vehicle and its sensors, as depicted in Figure 6. Each item in these three categories is matched with a visual representation and physical characteristics (dimensions, weight, friction, etc). The data acquisition and processing part of the simulation is carried out in ROS, where the data can be recorded, stored, and processed by the same code running on the actual vehicle. The communication between the ROS and Gazebo modules is carried out seamlessly thanks to the native use of ROS messages. In order for our simulation approach to be precise and fully exploitable, the simulation framework must provide the following elements:

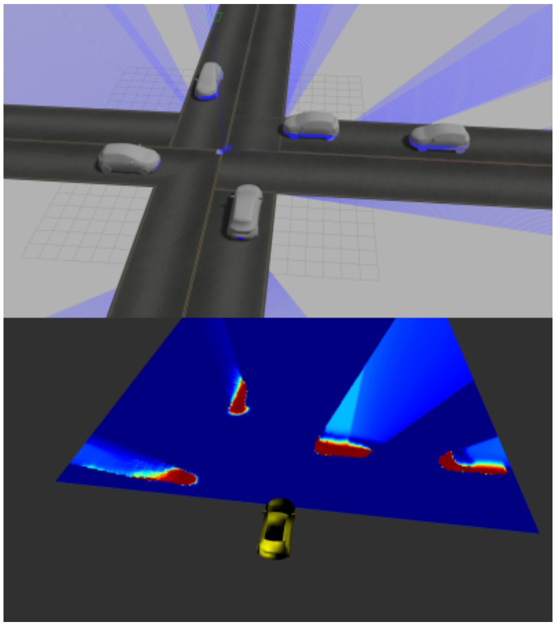

Figure 6. Simulated scenario for the CMCDOT algorithm (top),Output of CMCDOT (bottom)

- precise volume and shape of each vehicle, and surface reflectivity.
- atmospheric conditions which might impact the vehicles' trajectory (wind gusts) or lidar detection (heavy rain or snow).
- in order to establish the ground truth, a grid indicating the position of all simulated objects. This grid must reflect CMCDOT's occupation grid in the following aspects: origin position, grid direction, cell size.

Currently, each lidar is simulated with the appropriate position on the ego vehicle, the same sampling frequency and the same data format as the physical sensor. To match the sensing uncertainty, a Gaussian noise can be added.

In order to be able to efficiently generate a large number of simulated environments, we have perfected a parameter-based approach which streamlines the process through which the dimensions and initial position and velocity of non-ego vehicles are specified.

Our simulation scenario aims at checking the behaviour of cars at a four-way crossroads. The rule governing this crossroad is that at any given moment in time, a maximum of one simulated vehicle is present on the crossroad. To simulate the different cases, we rely on the random generation of parameter sets (non-ego vehicle class, initial position and initial speed). The test cases are then run, and their results (perception results as in Figure 6) are stored alongside the parameter sets. The analysis of these datasets enables us to accurately measure the efficiency of our perception and control solution.

The strong advantage of this approach is the ease with which a large number of simulated scenarios can be generated, ran, and analyzed.

2) KPI definition: Contrary to most perception systems, outputs of CMCDOT are not a direct list of detected objects, but dynamic occupancy grid, a rich probabilistic representation of the entire surrounding space. While object detector metrics are already not perfectly defined, the topic of evaluation of occupancy grids (furthermore dynamic occupancy grids, incorporating at a cell level velocity field estimations) is an important subject [26].

A first approach is to define a global indicator based on the direct estimates of the grid, in comparison to the ground truth. But if by qualitative analysis of results it is quite simple to evaluate if an occupancy grid is correct or not, an objective quantification of this quality is particularly complicated, each metrics focusing on a specific aspect, ignoring others (for example occupied / free space factor, cell by cell comparison, convolution-based metrics, etc.).

Another approach is to focus on specific applications of the method: the validation of the whole system itself is performed by statistical validation of its usages. In the case of the CMCDOT framework, a direct application of the perception system is an automatic braking system, based on aggregated risk estimates of the system. By comparing the difference in response of the system and expected behavior according to the ground truth, a partial evaluation of the system can be accessed.

In order to assess the correctness of the CMCDOT algorithm, we compare the output of the algorithm to the actual context of the car in the simulation. We focus on the risk of collision at 1, 2 and 3 seconds.

In order to evaluate the correctness of this output, we extract a traces of the simulation containing the following metrics: $\texttt{cmcdot\_risk}_i$ and $\texttt{real\_coll}_i$ for $1 \leq i \leq 3$. The metric $\texttt{cmcdot\_risk}_i$ indicate the probability of a collision in i s according to the CMCDOT algorithm. The metric $\texttt{real\_coll}_i$ is a Boolean indicating whether a collision will occur if object continue to move with their current speed, according to their speed and position in the simulation.

We define one KPI for each time interval, parameterized by a threshold τ. We formalize our KPI through the property $G_{\leq t}(\texttt{real\_coll}_i \Rightarrow (1 - \texttt{cmcdot\_risk}) < \tau) \wedge (\neg\texttt{real\_coll}_i \Rightarrow \texttt{cmcdot\_risk}) < \tau)$. This property states that if there is a risk of collision, the probability returned by CMCDOT must be high enough. Conversely, if there is no risk of collision, the probability returned by CMCDOT must be small enough.

IV. A SECOND VALIDATION APPLICATION: A DECISION-MAKING SYSTEM

A. Principle of the POMDP based decision-making

The decision-making system is a key component of an autonomous vehicle. Its task is to plan the movement of the vehicle taking into account the uncertainty in the situation measurement as well as the uncertain consequences of its action will have on the situation.

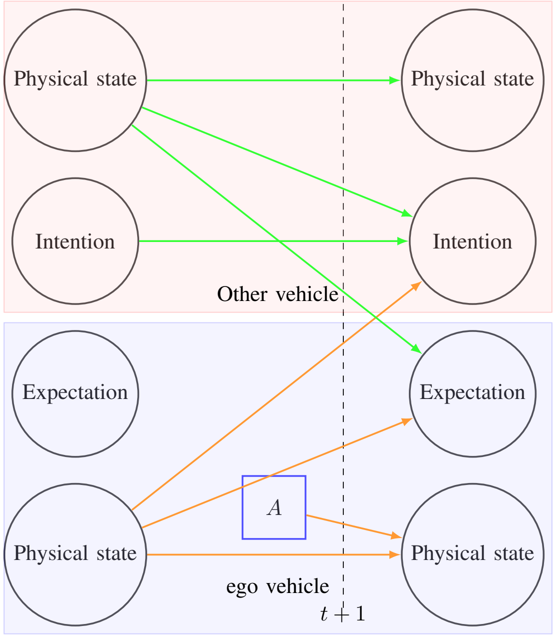

Figure 7. The POMDP represented as Bayesian network. The square node represent the action chosen by the framework

Partially observable Markov decision process (POMDP) is a mathematical model that formalizes this two kind of uncertainties and has been used for planning in stochastic environment [27].

With recent advancement on online-Pomdp solver [28](used in our work), complex problems such as road intersection crossing has been addressed in [29]. The key element of our approach [30] is to take into account the difference between intention and expectation of drivers approaching an intersection (inspired from [31]) to enable partial cooperation. The intention corresponds to the manoeuvre actually performed by the drivers and could observed with the approach developed in [32]. The expectation represent what the driver should do regarding the current situation and traffic rules. Situation where intention and expectation does not match could result in risky interactions. These two variables can be inferred from the physical state (Velocity and distance towards the intersection) of both vehicles. Our model is represented as a Bayesian network in Figure 7 that shows the interaction between variables. The reward function of the model is constructed to take into account: comfort, velocity, time to collision, traffic rules and differences between intention and expectation. The system interacts with the environment by selecting an acceleration that maximize the current estimations of the sum of future expected rewards. Because of the stochastic aspect of the model and its solvers a safe intersection crossing cannot be guaranteed. Thus, a large number of simulations is required to validate the model in order to ensure the safety. The two problems is that the scenario space is large because of the different regulations, initial speeds or different behaviours. Then, the parameter space for the model, especially its reward function, is as large and need to be correctly explored in order to find the functional range of the system.

B. Method Application

1) Dedicated simulator development: The decision-making system interacts with the simulation trough observations that can be made on the situation and selected actions that have to be realized in the simulated environment. Thus the fidelity, that is how closely the simulator can generate environmental data and model the system that are not under test, is important. In our scenario, the micro-traffic simulation (vehicle state and interactions between vehicles) is more important than the macro-simulation (simulation of traffic as a group of vehicles). As our system selects actions, it expects the other vehicle to change its behaviour. For the ego vehicle, the dynamic model of the vehicle does not need to have an high fidelity but as we want, in the future, to compare results obtained against field operational testing, the possibility of having high fidelity model is a plus. The decision could be of different forms (trajectory, goal points, control input), so the communication between the system under test and the simulation models must be adaptable. Figure 8 represents the different scenarios that have to be tested (yield, stop controlled, or priority). Thus the simulator must generate the appropriate behaviour for each of the corresponding situations. Real life scenarios could be also be imported to increase the validity of the reproduced situation. It would require the importation of maps and perception data from other sources. Scaner [33], an automotive grade simulator, has been chosen to test the decision-making systems. It has been mostly used for vehicle in the loop testing. However, most of the features previously described are available, at various levels of maturity. It has simple but interactive models for road intersection crossing and map generation. Scaner features a batch testing function, that we found too complex to interface with the SMC.

2) KPI definition: In order to evaluate the quality of the decision algorithm, we define some Key Performance Indicators regarding the crossing of a intersection. First, we define two areas in the intersection: a critical area, that correspond to the actual intersection where stopped vehicles would block all branches of the intersection and a non-critic area, that correspond to the entry of the intersection where cars usually stop before crossing the other road. We count the number and total duration of stops in each area, a smaller number indicates a better quality of the algorithm. We also measure the total time needed to cross the intersection, where again a smaller number indicates a better quality. We measure the acceleration to evaluate the comfort of the passenger, where again a smaller number indicates a better quality.

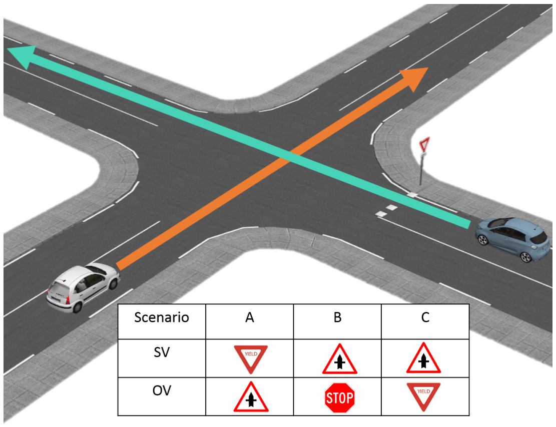

Figure 8. Simulated scenario for the decision-making. The ego vehicle (blue) is controlled by the decision-making system and have to interact with the other vehicle (white) with respect to the traffic rules

Table I
LIST OF VARIABLES EXTRACTED FROM THE SIMULATIONS.

Name	Description	Unit
t	Timestamp or time elapsed	s
nc_stops	Number of stops in the non-critical area	
c_stops	Number of stops in the critical area	
t_nc_stops	Duration of stops in non-critical area	s
t_c_stops	Duration of stops in critical area	s
acc	Acceleration	ms^{-2}
crossed	True if intersection is crossed	

For all metrics m whose smaller value indicates a better performance, we check whether m is bounded by a bound b. The formula $G_{\leq t} m \leq b$, with t corresponding to the time needed to cross the intersection, states that m is always smaller than b. Stating that the acceleration must always be smaller than a bound might be a constraint too strong. We thus propose a relaxed version of this KPI where the acceleration is allowed to be above the bound for a short period of time (1s). This is stated by the formula $G_{\leq t} F_{\leq 1} acc \leq b$. The previous formula can be read as follows: at any point during the simulation, m will be smaller than b in less than 1s. In other words, it is not possible that $m > b$ for more than 1s. The value of the bound b is defined w.r.t. the metric considered.

Finally, to evaluate whether the intersection is crossed quickly enough, we set a maximum duration d for crossing the intersection and require that the intersection is crossed in less than d seconds, stated by $F_{\leq d} crossed$.

3) SMC application: In order to obtain results, we selected for each metric some adequate bounds and plot the probability that the KPI is met for each bound. The Figure 9 represents the probability that the acceleration/deceleration remains below a certain bound when crossing the intersection, both for the strict (i.e. the bound is never exceeded) and the relaxed version (the bound is never exceeded for more than 1s). We see that there is a probability 0 that the acceleration stays below an absolute value of $0.8 m.s^{-2}$, and that it is always below $2 m.s^{-2}$. It corresponds to an acceptable range for human comfort and shows that in every scenario the decision-making system took actions to adapt the behaviour.

Figures 10 and 11 present the probability of respectively having a bounded number of stops and having a bounded total stop duration. We see that there is a probability 0.9 that the

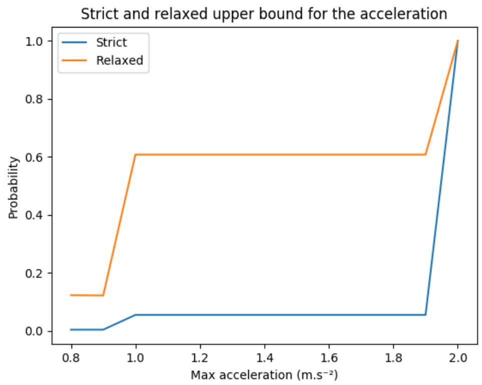

Figure 9. Probability that the absolute value of the acceleration remains bounded, for the strict and the relaxed version.

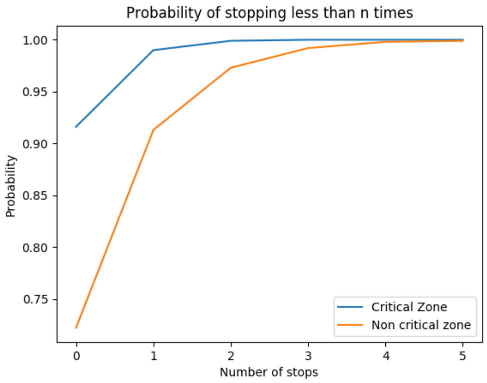

Figure 10. Probability of bounded occurences of stops, for critical and non-critical zones.

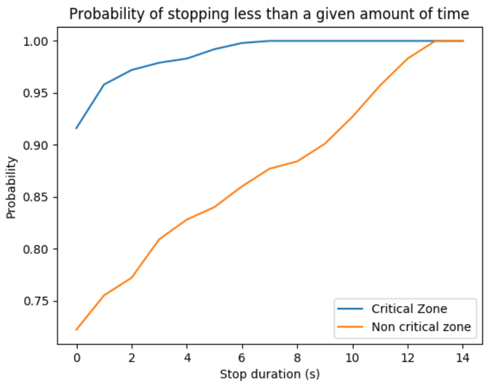

Figure 11. Probability of a stop duration below a given bound, for critical and non-critical zones.

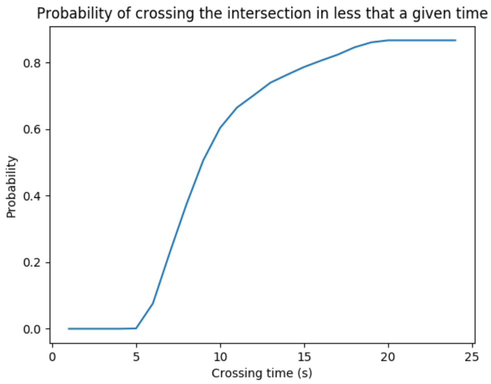

Figure 12. Probability of crossing the intersection in less than a given time.

car does not stop in the critical zone. With that measure it can be said that most likely the ego vehicle will comply with the traffic law. However for the 0.1 probability that the vehicle stop whitin the intersection, causes for the subject to come to a stop must be investigate in order to find if it correspond to an emergency manoeuvre or a failure of the system. This could be done by introducing finer KPIs that would take into account the temporality of the problem.

In Figure 12 we show the probability to cross the intersection in less that a given duration. All this new information tells people in charge of the validation what is the most likely behaviour of the decision-making system. It also helps people working on designing the decision-making to find area of improvement in their systems.

V. Conclusion

In this paper we presented and demonstrated a pipeline for the validation of different ACPS on two different automotive use-cases. The application of our approach based on Statistical Model Checking to the decision-making system provides useful information to the designer of the system and to the people in charge of the validation. This valuable information is formulated through probability for our system to stay in a certain range of KPIs.

Future works include the definition of meaningful grid-based metrics for stating more discriminating KPIs about the perception system. We also plan to compare results obtained in the simulated environment with tests on proving ground to ensure the validity of our approach. Also more KPIs for the decision and perception could be introduced to accurately pinpoint the cause of identified failures.

Acknowledgment

This work has been developed within the scope of ”Institut de Recherche Technologique” NanoElec, founded by the French program ”Investissements d’Avenir” ANR-10-AIRT-05.

This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL joint undertaking under grant agreement NO 692455. This joint undertaking receives support from the European Union’s Horizon 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

REFERENCES

[1] T. Bokc, M. Maurer, and G. Farber, "Validation of the vehicle in the loop (vil); a milestone for the simulation of driver assistance systems," in *2007 IEEE Intelligent Vehicles Symposium*, June 2007, pp. 612–617.

[2] T. Hwang, J. Roh, K. Park, J. Hwang, K. H. Lee, K. Lee, S. j. Lee, and Y. j. Kim, "Development of hils systems for active brake control systems," in *2006 SICE-ICASE International Joint Conference*, Oct 2006, pp. 4404–4408.

[3] J. Ibanez-Guzman, S. Lefevre, A. Mokkadem, and S. Rodhaim, "Vehicle to vehicle communications applied to road intersection safety, field results," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sept 2010, pp. 192–197.

[4] Enable-S3, "Validation and testing of complex automated systems," https://www.enable-s3.eu/, 2016, last accessed 29/06/2018.

[5] M. G. Hinchey, J. L. Rash, and C. A. Rouff, "Verification and validation of autonomous systems," in *Proceedings 26th Annual NASA Goddard Software Engineering Workshop*, 2001, pp. 136–144.

[6] S. Ulbrich, D. Kappler, T. Asfour, N. Vahrenkamp, A. Bierbaum, M. Przybylski, and R. Dillmann, "The opengrasp benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 1761–1767.

[7] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 719–726.

[8] Waymo, "Waymo s safety report: how we are building a safer driver," https://medium.com/waymo/waymos-safety-report-how-we-re-building-a-safer-driver-ce5f1b0d4c25, 2017, last accessed on 22/05/18.

[9] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[10] E. T. Jaynes, *Probability theory: the logic of science*. Cambridge university press, 2003.

[11] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet, "Approximate probabilistic model checking," in *Proceedings of the 5th International Conference on Verification, Model Checking, and Abstract Implementations*, ser. Lecture Notes in Computer Science, B. Steffen and G. Levi, Eds. Germany: Springer Berlin Heidelberg, 2004, vol. 2937, pp. 73–84.

[12] K. Sen, M. Viswanathan, and G. Agha, "On statistical model checking of stochastic systems," in *Proceedings of the 17th International Conference on Computer Aided Verification*, ser. Lecture Notes in Computer Science, K. Etessami and S. K. Rajamani, Eds. Germany: Springer Berlin Heidelberg, 2005, vol. 3576, pp. 266–280.

[13] K. Yi and J. Chung, "Nonlinear brake control for vehicle cw/ca systems," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 1, pp. 17–25, Mar 2001.

[14] A. Pnueli, "The temporal logic of programs," in *Proc. of the 18th Annual Symposium on Foundations of Computer Science*, ser. SFCS '77. Washington, DC, USA: IEEE Computer Society, 1977, pp. 46–57.

[15] P. Zuliani, A. Platzer, and E. M. Clarke, "Bayesian statistical model checking with application to stateflow/simulink verification," *Formal Methods in System Design*, vol. 43, no. 2, pp. 338–367, Oct 2013. [Online]. Available: https://doi.org/10.1007/s10703-013-0195-3

[16] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[17] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, no. 2, p. 61, 1988.

[18] P. Bessière, E. Mazer, J. Ahuactzin-Larios, and K. Mekhnacha, *Bayesian Programming*. CRC Press, Dec. 2013.

[19] M. Yguel, O. Aycard, and C. Laugier, "Efficient gpu-based construction of occupancy grids using several laser range-finders," in *International Journal of Vehicle Autonomous Systems*, vol. 6, 10 2006, pp. 105–110.

[20] L. Rummelhard, A. Négre, and C. Laugier, "Conditional monte carlo dense occupancy tracker," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2485–2490.

[21] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, vol. 19. IEEE, 1980, pp. 807–812.

[22] Z. Khan, T. Balch, and F. Dellaert, "An mcmc-based particle filter for tracking multiple interacting targets," in *Computer Vision-ECCV 2004*. Springer, 2004, pp. 279–290.

[23] R. Labayrade, C. Royere, and D. Aubert, "Experimental assessment of the rescue collision-mitigation system," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 1, pp. 89–102, Jan 2007.

[24] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, Jan 2009.

[25] L. Rummelhard, A. Nègre, M. Perrollaz, and C. Laugier, "Probabilistic grid-based collision risk prediction for driving application," in *International Synposium on Experimental Robotics*, Springer, Ed., Marrakech, Marocco, 2014.

[26] R. Grewe, M. Komar, A. Hohm, S. Lüke, and H. Winner, "Evaluation method and results for the accuracy of an automotive occupancy grid," *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pp. 19–24, 2012.

[27] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[28] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 2164–2172. [Online]. Available: http://papers.nips.cc/paper/4031-monte-carlo-planning-in-large-pomdps.pdf

[29] W. Liu, S. W. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online pomdp," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1126–1133.

[30] M. Barbier, C. Laugier, O. Simonin, and J. Ibañez-Guzmán, "A pomdp-based intention-expectation decision-making and key performance indicators for road intersections crossing," 2018, submitted to IEEE Intelligent Vehicles Symposium (IV) 2018, under review.

[31] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Risk assessment at road intersections: Comparing intention and expectation," in *2012 IEEE Intelligent Vehicles Symposium*, June 2012, pp. 165–171.

[32] M. Barbier, C. Laugier, O. Simonin, and J. Ibañez-Guzmán, "Classification of drivers manoeuvre for road intersection crossing with synthethic and real data," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 224–230.

[33] Oktal, "Scaner studio," http://www.oktal.fr/en/automotive/range-of-simulators/software, 2017, last accessed 05/02/2018.