Validation of Perception and Decision-Making Systems for Autonomous Driving via Statistical Model Checking

Mathieu Barbier, Alessandro Renzaglia, Jean Quilbeuf, Lukas Rummelhard, Anshul Paigwar, Christian Laugier, Axel Legay, Javier Ibañez-Guzmán, Olivier Simonin

Validation of Perception and Decision-Making Systems for Autonomous Driving via Statistical Model Checking

Mathieu Barbier[1,2], Alessandro Renzaglia[1], Jean Quilbeuf[3], Lukas Rummelhard[1], Anshul Paigwar[1],
Christian Laugier[1], Axel Legay[4], Javier Ibañez-Guzmán[2] and Olivier Simonin[5]

***Abstract*— Automotive systems must undergo a strict process of validation before their release on commercial vehicles. With the increased use of probabilistic approaches in autonomous systems, standard validation methods are not applicable to this end. Furthermore, real life validation, when even possible, implies costs which can be obstructive. New methods for validation and testing are thus necessary. In this paper, we propose a generic method to evaluate complex probabilistic frameworks for autonomous driving. 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 studying the behavior of these metrics during a large number of simulations via our statistical model checker, we finally evaluate the probability for the system to meet the KPIs. We show how this method can be applied to two different subsystems of an autonomous vehicle: a perception system and a decision-making approach. An overview of these two systems is given to understand related validation challenges. Extensive validation results are then provided for the decision-making case.**

I. Introduction

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

- Vehicle-in-the-loop to test 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 level 3, as defined by the SAE, drivers will not be responsible of most driving decisions. A thorough validation of the concerned algorithms and subsystems is thus of fundamental importance in these contexts. However, as these systems will rely more and more on machine learning and probabilistic methods, conventional validation methods are not suitable and new solutions need to be adopted. Furthermore, the vehicles will eventually operate in a wide range of scenarios, including dangerous situations. As a result, a validation and verification process performed in simulations is preferable, since it allows increasing the coverage of system testing, while also reducing costs.

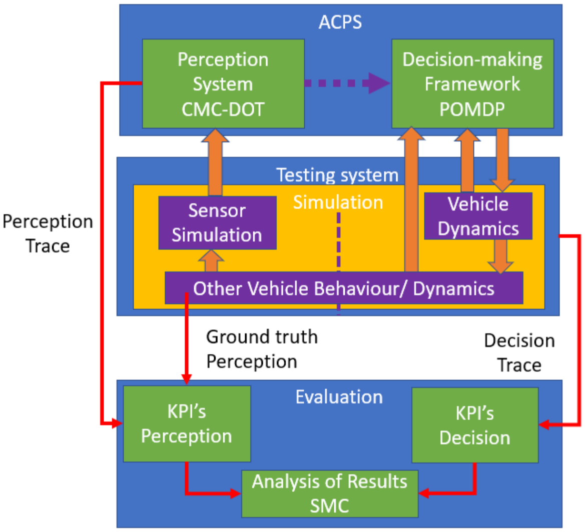

Fig. 1. Interactions between the different elements of the proposed validation pipeline. Dashed lines represents future developments to connect the decision and perception.

Two main challenges can be identified in the validation of algorithms for autonomous driving. First, the complexity and variety of scenarios that the vehicles can face is larger than in Advanced Driver Assistance Systems (ADAS). Second, the necessity of considering the constant possibility of interaction between multiple systems. In this study we focus on a use-case that highlights these two difficulties: road intersection crossing. Road intersections are among the most dangerous part of road networks with more than 8% of the total road fatalities in Europe [3]. From a perception point of view, this scenario is particularly challenging because of the limitations in the visibility field, resulting in only partially-observed vehicles. In terms of decision-making, the possibility of a wrong, unexpected behavior of other drivers makes the road intersections particularly complex to consider. For these reasons, this scenario has been identified as one of the main use-cases addressed in the Enable-S3 European project [4]. This industry-driven project aspires to propose new methods for validation and verification of Automated Cyber-Physical Systems (ACPS). The global architecture for validation and verification has been simplified to match our scenario and is illustrated in figure 1.

[1] Univ. Grenoble Alpes, Inria, Chroma, F-38000 Grenoble. Email: name.surname@inria.fr

[2] Renault S.A.S, 1 av. du Golf, 78288 Guyancourt, France. Email: name.surname@renault.com

[3] Univ Rennes, Inria, F-35000, RENNES, France. Email: name.surname@inria.fr

[4] Université Catholique de Louvain, Computer Science Department, B-1348 Louvain-la-Neuve, Belgium. Email: name.surname@uclouvain.be

[5] INSA Lyon, Inria, Chroma, CITI Lab., 6 avenue des Arts, 69680 Villeurbanne, France. Email: name.surname@inria.fr

Other examples of validation approaches for highly-autonomous systems can be found in literature, for instance in the aerospace domain, where formal methods are used to validate the behavior of a fleet of satellites [5]. 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 situations 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 be rather complex to do in uncertain environments.

The main contribution of this paper is to propose and demonstrate the use of a validation method, based on Statistical Model Checking (SMC), able to overcome the aforementioned limitations. In particular, we show its applicability to two different algorithms dealing with the perception and the decision-making problem respectively. The requirements for the testing in simulated environments are discussed for each system. Preliminary results for the decision-making system are presented as well as a discussion on the challenges generated by the perception system.

The rest of the paper is organized as follows. Section II presents the details of our validation approach based on statistical model checking. Section III describes the application of our approach to the perception system and the difficulties in finding applicable metrics for its validation. Then, Section IV presents a more complete application for the decision-making approach and results are provided and discussed.

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 algorithms 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 transitions 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.

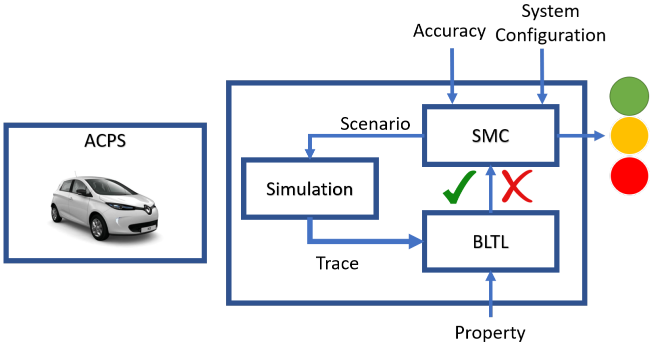

Fig. 2. An overview of SMC

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 satisfies the property.

We present in Figure 2 an overview of the approach. On the left we have a simulator that provides stochastic executions of our system. On the bottom we have the property 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 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 ω that a system satisfies a property Ω by checking Ω against a set of N random executions of the system. The estimation $\hat{\omega}$ is given by

$$\hat{\omega} = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \;\text{ where } f(x_i) = \begin{cases} 1 & \text{if} x_i \models \Omega \\ 0 & \text{otherwise} \end{cases}$$

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:

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

Assuming that ω is the value of the probability we want to evaluate and $\hat{\omega}$ is the estimation we compute, the formula means that the estimation error, i.e. the distance $|\omega - \hat{\omega}|$, is bounded by ϵ with a probability $1 - \delta$. In other words, the probability that the error in the estimation is greater than 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 some KPIs related to the system and scenario under test. 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 Bounded Linear Temporal Logic (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.

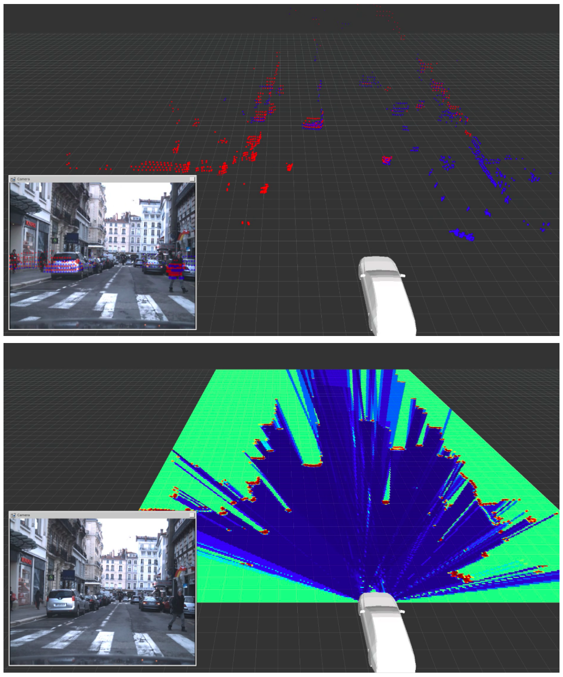

Fig. 3. Data fusion in an occupancy grid. Data from each of the 2 lidars are used to generate occupancy grids using sensor models, which are then combined by using Bayesian fusion.

III. A first validation application: CMCDOT Perception system

A. *Principle of the CMCDOT*

The Conditional Monte Carlo Dense Occupancy Tracker (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 [18].

This type of Bayesian formalism [19] 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 [20], 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 CMCDOT 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

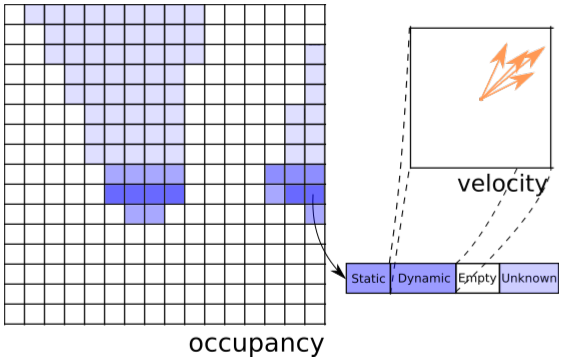

Fig. 4. Data representation in the CMCDOT formulation. The environment is divided into cells, to which are associated static, dynamic, empty and unknown coefficients. The dynamic part is allotted to weighted particles which sample the velocity space.

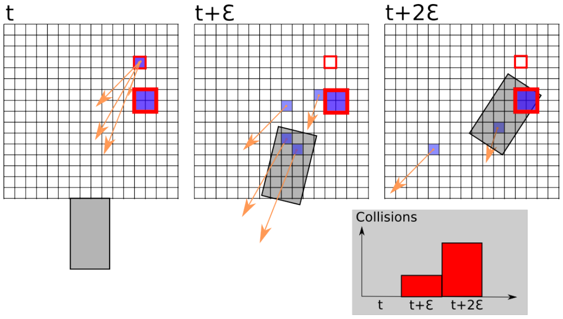

Fig. 5. Collision risk estimation over time for a specific cell. The cell position is predicted according to its velocity, along with the mobile robot. This risk profile is computed for every cell, and then used to integrate over time the global collision risk.

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 [18] instead directly computes estimations of the 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.

B. Method Application

1) Simulation for perception: In this project, the simulation relies on the use of two frameworks: CARLA, an open urban driving simulator [26], and Robot Operating System (ROS). CARLA simulation environment consists of complex urban layouts, buildings and vehicles rendered in high quality, allowing for a realistic representation of real-world scenarios. The ego vehicle and its sensors, as well as other moving vehicles, as depicted in Figure 6, can be configured in the simulation to match with the actual system. The provided CARLA-ROS bridge enables data acquisition from the simulation in native ROS message formats, where the data can be recorded, stored, and processed by the same code running on the actual vehicle.

In order to establish the ground truth, a grid indicating the position of all simulated objects is needed. This grid must reflect CMCDOT's occupancy grid in the following aspects: origin position, grid direction, cell size and velocities. The bounding box and velocity information provided by CARLA simulator is translated into occupancy grid at each time step to generate the ground truth.

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 designed 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 behavior of cars at a four-way crossroads and validating Time to Collision (TTC) estimated by CMCDOT. 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 a 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 Fig. 6) are stored alongside the parameter sets. The analysis of these datasets enables us to accurately measure the efficiency of our perception and estimation 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, the output of CMCDOT is not a direct list of detected objects, but a dynamic occupancy grid, i.e. a rich probabilistic representation of the entire surrounding space. Evaluating such occupancy grids, and especially dynamic occupancy

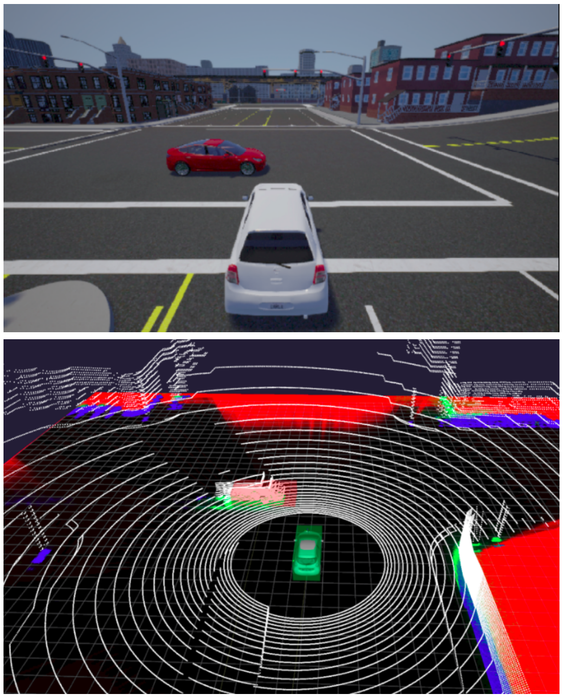

Fig. 6. Simulated scenario in Carla (top) and output of CMCDOT (bottom).

grids, incorporating at a cell level velocity field estimations, is still an important subject of research [27].

A first approach would be to define a global indicator based on the direct estimates of the grid, to be compared 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 would be 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.

We here propose a more straightforward approach taking into account a different output of the CMCDOT algorithm: the estimated risk of collision. In particular, the probabilities for collision in 1, 2 and 3 seconds can be considered.

In order to evaluate the correctness of this output, we can study the 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.

For each time interval, a different KPI parameterized by a threshold τ can be defined. 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 collected measurements as well as the uncertain consequences that its action will have on the situation.

Partially Observable Markov Decision Process (POMDP) is a mathematical model that can appropriately formalize this two kind of uncertainties and has been widely used for planning in stochastic environments [28].

With recent advancements on online-Pomdp solvers (e.g. [29], adopted in our work), complex problems such as road intersection crossing has been addressed in [30]. The key element of our approach [31], inspired by [32], is to take into account the difference between intention and expectation of drivers approaching an intersection to enable partial cooperation. The intention corresponds to the maneuver actually performed by the drivers and it can be classified by employing the approach presented in [33]. The expectation represents what the driver should do regarding the current situation and traffic rules. Situations where intention and expectation do not match could result in risky interactions. These two variables can be inferred from the physical state (velocity and distance approaching the intersection) of both vehicles. Our model is represented as a Bayesian network, as in Fig. 7 where the interaction between variables is shown.

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 maximizes 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 formally guaranteed. Thus, a large number of simulations is required to validate the model in order to ensure a safe behavior. The challenge is represented by the large dimension of the scenario space, due to the various regulations, different initial velocities and numerous possible behaviors. Then, the parameter space for the model, especially its reward function, is equally large and needs 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

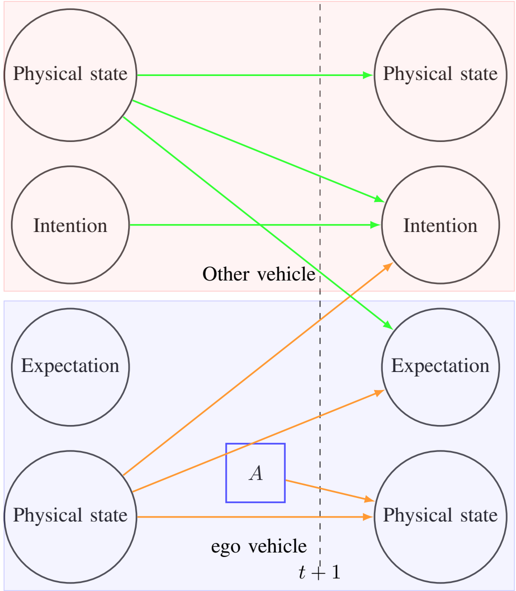

Fig. 7. The POMDP represented as a Bayesian network. The square node represents the action chosen by the framework.

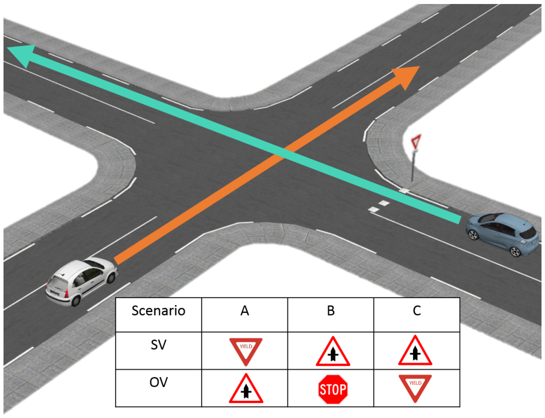

Fig. 8. Simulated scenario for the decision-making. The ego vehicle (blue) is controlled by the decision-making system and has to interact with the other vehicle (white) respecting the traffic rules.

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, is fundamental. 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 behavior. For the ego vehicle, the dynamic model of the vehicle does not need to have a high fidelity. However, since in the future we want 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. Fig. 8 represents the different scenarios that have to be tested (yield, stop controlled, or priority). The simulator must generate the appropriate behavior for each of the corresponding situations. Real-life scenarios could be also be imported to increase the validity of the reproduced situation. It would only require to import maps and perception data from other sources. Scaner [34], 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 an intersection. First, we define two areas in the intersection: a critical area, that corresponds to the actual intersection where stopped vehicles would block all branches of the intersection, and a non-critic area, that corresponds 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.

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 for which smaller values indicate 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 too-strong constraint. 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 considered metric.

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

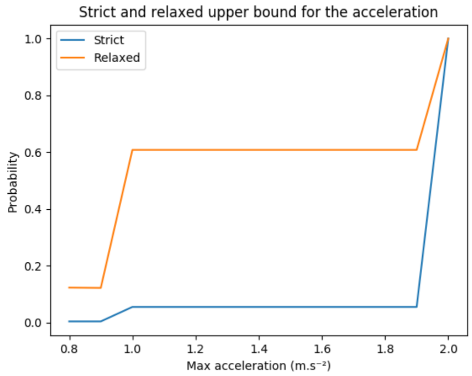

Fig. 9. Probability that the absolute value of the acceleration remains bounded, for the strict and the relaxed version.

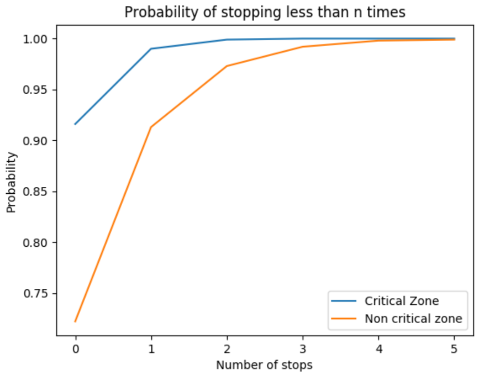

Fig. 10. Probability of bounded occurrences of stops, for critical and non-critical zones.

probability that the KPI is met for each bound. Fig. 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 zero probability that the acceleration stays below an absolute value of $0.8m/s^2$, and that it is always below $2m/s^2$. This corresponds to an acceptable range for human comfort and shows that in every scenario the decision-making system took actions to adapt the behavior.

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 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, the causes of the remaining 0.1 probability that the vehicle stops in the intersection need to be investigated to determine whether it corresponds to emergency maneuvers or a failure in the system. This could be done by introducing finer KPIs that would take into account the temporality of the problem.

In Fig, 12 we finally show the probability to cross the intersection in less than a given time.

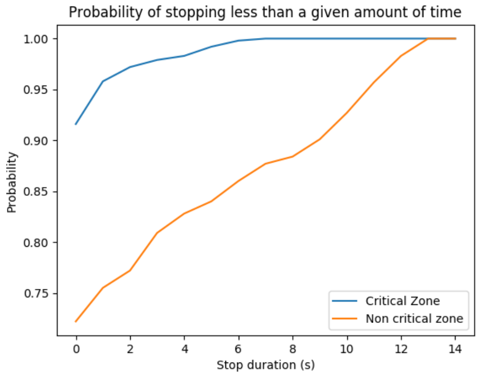

Fig. 11. Probability of a stop duration below a given bound, for critical and non-critical zones.

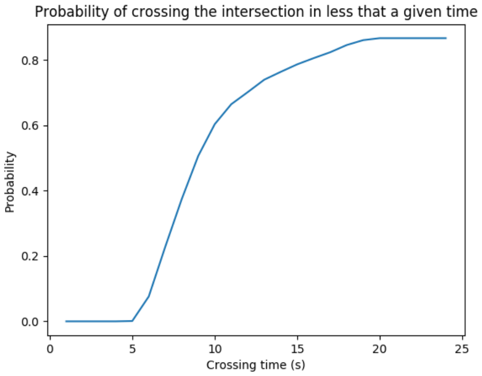

Fig. 12. Probability of crossing the intersection in less than a given time.

All this information can be then exploited to define what is the most likely behavior of the decision-making system in real scenarios, and it can also be useful to guide the improvement of these approaches, highlighting their weak aspects.

V. Conclusion

In this paper we presented and demonstrated a pipeline for the validation of different components of an 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 designers 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.

In the future, we intend to complete our analysis on the collision-risk estimation and propose meaningful grid-based metrics for stating more discriminating KPIs for 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. Additionally, more KPIs for the decision system could be considered to accurately pinpoint the causes 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*, 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*, 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*, 2010, pp. 192–197.

[4] Enable-S3, “Validation and testing of complex automated systems,” https://www.enable-s3.eu/.

[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*, 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)*, 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.

[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. 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. 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, 2001.

[14] A. Pnueli, “The temporal logic of programs,” in *Proc. of the 18th Annual Symposium on Foundations of Computer Science*. 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*, 2013.

[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] 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.

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

[20] 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, 2006.

[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, 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] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

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

[28] 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.

[29] D. Silver and J. Veness, “Monte-carlo planning in large pomdps,” in *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010.

[30] 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)*, 2015, pp. 1126–1133.

[31] M. Barbier, C. Laugier, O. Simonin, and J. Ibañez-Guzmán, “Probabilistic decision-making at road intersections: Formulation and quantitative evaluation,” in *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 795–802.

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

[33] 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)*, 2017, pp. 224–230.

[34] Oktal, “Scaner studio,” http://www.oktal.fr/en/automotive/range-of-simulators/software, 2017.