



HAL
open science

A simulation-based framework for functional testing of automated driving controllers

Adel Djoudi, Loic Coquelin, Rémi Regnier

► **To cite this version:**

Adel Djoudi, Loic Coquelin, Rémi Regnier. A simulation-based framework for functional testing of automated driving controllers. The 23rd IEEE International Conference on Intelligent Transportation Systems, Sep 2020, Virtual Conference, Greece. hal-02862369

HAL Id: hal-02862369

<https://hal.science/hal-02862369>

Submitted on 22 Jun 2020

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.

A simulation-based framework for functional testing of automated driving controllers

Adel Djoudi¹, Loic Coquelin^{1,2}, and Rémi Régnier^{1,2}

Abstract—Motion planning is a major component of any automated driving system. The safety assessment of such components requires a formal characterization of the perception and control mechanisms. This requires dedicated tools and models for the environment, sensors and vehicles that are highly representative of the real world. Simulation is a method to virtually investigate the behavior of systems under study. It has a key role to play in demonstrating the safety of autonomous vehicles. In this context, we consider a control module as a black-box and try to determine a reference which represents the 'right decision', if it exists. An optimization-based reference model is created for the control function. This model allows each scene in the environment to be mapped to the desired decision regardless of the black-box. The black-box and the reference model are run on several critical scenarios. In output, an assessment of decision making is performed along with systematic criticality characterization of targeted scenarios.

I. INTRODUCTION

Several Automated Driving Systems (ADS) aim to ensure driver safety and comfort. Testing is required to gain confidence that the ADS can operate effectively across the full range of situations it is likely to encounter. However, it is not realistic to physically test every combination of sensor input and driving situation. Currently, physical testing of Autonomous Vehicles (AVs) would require to accumulate an incalculable number of miles (likely billions of miles) without offering means to rigorously control the ADS evaluation, neither to repeat test cases on critical situations at reasonable cost. A secure test site is required, and changing the test setup and the subsequent repeated test drives in that site requires a lot of effort and time. On the other hand, simulation, modeling and testing has the potential to fill this gap and to enable rigorous, controlled and timely evaluation of ADS. The 3SA (Simulation for the Safety of Systems in Autonomous Vehicles) project, which is part of the French automotive platform PFA, aims to go further in the use of digital simulation to demonstrate the safety of ADS.

a) Safety evaluation: The topic of "safety" must have a paramount priority. In the case of ADS, human intervention may take place too late in a dangerous situation or (for systems with a high autonomy level) is not planned at all. Thus, a highly or fully automated driving system must have appropriate capabilities and strategies in order to

be functionally safe. It must identify potentially dangerous situations and then respond appropriately to the situations in order to avoid imminent danger or minimize damage as far as possible. The intended system functionality and the necessary safety functions cannot be implemented separately, but are two sides of the same coin.

b) Challenges: The major challenge here is to ensure that ADSs are designed in such a way that they can be approved to be (sufficiently) safe for use in public spaces (e.g. road traffic). With conventional, non-autonomous safety-critical systems, many standards use Safety Integrity Levels SIL as a measure of reliability and/or risk reduction. The associated SILs for automotive industry (of human-driven vehicles) are defined in the series of standards [1], [2]. A non-autonomous safety-critical system is designed in such a way that in the event of a technical fault, immediate danger is greatly reduced or eliminated by switching the system off or stopping its operation. This approach is not sufficient for automated driving systems! A self-driving car would increase the danger it poses instead of reducing it if it would stop "in the middle of the road" in the event of a sensor's failure. Autonomous systems should therefore have appropriate "fail-operational" capabilities [3], [4]. Consensus must be reached on suitable approval criteria. Existing approval procedures must be supplemented or new ones developed and adopted. Manufacturers will have to prove that their own products meet the approval criteria. The systematic testing of such products will play a major role in this context.

c) ADS architecture: Automated driving relies upon several sub-tasks which form the overall architecture of an autonomous vehicle. These sub-tasks can be formulated as test tasks [5]:

- Sensing: Capturing relevant environment data.
- Perceiving: Recognizing patterns or situations.
- Analyzing: Identifying options for appropriate action.
- Planning: Selecting appropriate or best action.
- Acting: Implementing chosen action correctly.

d) Functional testing: The purpose of functional tests is to test each function of the System Under Test (SUT), by providing appropriate input and verifying the output against the functional requirements. Functional testing mainly involves black-box testing and it is not concerned about the inner structure of the SUT. The testing can be done either manually or using automation. In order to functionally test an application, the following steps must be observed:

- Understand System Requirements & identify test input;
- Compute expected results with selected test input;

*This work is part of the Simulation for the Safety of Systems in Autonomous Vehicles (3SA) and is supported by Institute of Research and Technology SystemX, France

¹ Institute of Research and Technology SystemX, France
name.surname@irt-systemx.fr

² National laboratory for metrology and testing (LNE), France
name.surname@lne.fr

- Compute actual results by executing test cases;
- Compare actual and expected results.

Functionality-based testing approaches categorize the components of driving intelligence for autonomous vehicles into three parts: sensing/recognition functionality, decision functionality according to the recognized information, and action functionality with respect to the decision. The most important benefit of functionality-based testing is that we could quantitatively evaluate a part of driving intelligence through a reduced number of designed tests. In this work, we focus on functional testing of the "decision" functionality that encompasses "analyzing" and "Planning" functionalities.

e) Scenario-based testing: The systematic testing of ADS requires a catalogue of relevant situations that is as comprehensive as possible. These situations must be able to be varied in many parameters. It must also be possible to link the situations into scenarios (successive situations) [6] in order to bring about specific situations in a targeted manner. Hence, it is possible to examine the development over time from or through a specific situation and the timely, forward-looking behavior of the autonomous system. "Scenario-based Testing" refers to such testing of the behavior of a system in a sequence of situations. Relevant scenarios may be gathered in a central scenario catalogue for a neutral authority in order to create corresponding generally valid specifications, including any "acceptance tests". The standardization of formats for the exchange of such scenarios is being worked on. ASAM OpenSCENARIO defines a file format for the description of the dynamic content of driving and traffic simulators [7].

f) Testing strategy: Scenario-based testing requires the same test procedure to be repeated in a large number of variations of the driving environment. The systematic variation of the test environment is an essential and decisive part of the test strategy [8]. However, the combination of ADS and its mission complexity along with the environmental complexity lead to an astronomical number of potentially testable scenarios. Furthermore, it is difficult to reduce testing effort by prioritizing scenarios at the risk of omitting relevant ones. Testing only one such scenario in reality can require enormous effort (a secure test site is required, and changing the test setup and the subsequent repeated test drives in that site requires a lot of effort and time). A very large proportion of the necessary tests must and will therefore be carried out in the form of simulations. In order to be able to identify gaps in the test coverage and reduce redundancies, the test management must continuously monitor, evaluate and check the test data and results.

g) Contributions: Our work focuses on the evaluation of a control module of an automated driving system. Our main contributions are:

- Optimization-based reference behavior;
- Simulation framework for functional tests;
- Systematic classification of scenarios.

The rest of the paper is organized as follows: in section II we position our work in relation with existing approaches of simulation-based testing of automated vehicles

and optimization-based trajectory planification. Section III presents our optimization-based computation of a reference behavior for a control module of an automated driving system. For the sake of clarity, we focus on longitudinal control behavior. Section IV describes the simulation framework and results computed on an application example. Finally, conclusions and future works are given in section V.

II. RELATED WORK

Simulations provide an omniscient observer model, which means that the past and all future evolutions are known in every time step. Another relevant aspect is the ability to easily combine multiple simulation models and thus reuse existing work. [9] propose a complete framework for automated driving testing with different controllers and path planning algorithms in urban scenarios.

Motion planning techniques for autonomous driving can be classified into three categories: graph-search-based, sampling-based and optimization-based methods. The optimization-based method formulates motion planning as a mathematical optimization problem [10]. The planning is spatiotemporal which can deal with dynamic obstacles. It is performed in continuous space with infinitesimal resolution, which leads to better solutions than graph-search and sampling. Moreover, it is easier to formulate the constraints using a classic optimization framework. However, due to the complex non-convex constraints and the limitation of the existing numerical optimization techniques [11], the optimization-based motion planners are usually inefficient for real-time computation. In our case we propose to use offline optimization-based techniques to build a reference behavior of autonomous vehicles. Note that offline trajectory planification benefits from the knowledge of the scenario in advance (in this paper, a two-second advance knowledge of the scenario).

III. OPTIMIZATION-BASED REFERENCE BEHAVIOR

In order to automatically evaluate a control module of an ADS, it is necessary to systematically build the "right decision" if it exists. For this reason, we propose an optimization-based model of ADS control module (aka. test Oracle in functional testing literature) that is able to map each perceived environment situation into the expected decision. Once this baseline behavior computed, a numerical comparison with ADS control module behavior serves as a basis for the evaluation.

In the sequel, we show the application of our approach on an Advanced driver-assistance systems (ADAS) that automatically controls the longitudinal behavior of the equipped vehicle (hereafter called Ego vehicle):

- Adaptive Cruise Control (ACC): an intelligent form of cruise control that slows down and speeds up automatically to keep pace with the car in front (lead vehicle);

The evaluation process of control decision is depicted in Figure 1. The ADAS, while taking the control of the Ego vehicle (orange vehicle), tries to keep a safe distance w.r.t. the lead vehicle (blue vehicle). The expected behavior is

provided by our optimization-based model named Oracle (gray vehicle).

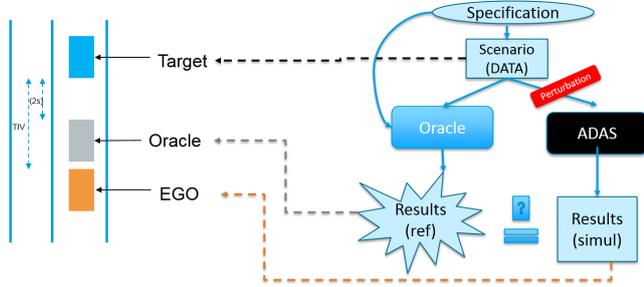


Fig. 1: Evaluation of ADS decision.

A. Requirements (Constraints)

Requirements resulting from recommendations of ISO/CD 22179 [12] for longitudinal control (ACC) are considered here to build our reference of longitudinal behavior. Requirements taken into consideration are:

- Under normal conditions, at speed V and given a distance of regulation expressed in terms of Time Inter-Vehicle constant TIV , the distance between the vehicle and the lead vehicle D must be greater than or equal to $\max(2m, TIV.V)$.
- The average deceleration over $2s$ shall not exceed $3m.s^{-2}$ when the speed is greater than or equal to $20m.s^{-1}$ and $5m.s^{-2}$ at a speed of $5m.s^{-1}$ or less.
- The average deceleration variation (jerk) over $1s$ must not exceed $2.5m.s^{-3}$ when the speed is greater than or equal to $20m.s^{-1}$ and $5m/s^{-3}$ at a speed of less than or equal to $5m.-s$.
- The average acceleration over $2s$ shall not exceed $2m/s^{-2}$ at a speed greater than or equal to $20m.s^{-1}$ and $4m/s^{-2}$ at a speed of less than or equal to $5m.s^{-1}$.
- The vehicle speed must comply with the speed command v_c defined by the driver.

These requirements are expressed as mathematical constraints for our optimization-based reference computation.

B. Optimization-based reference computation

In this section, we propose a reference behavior model formulation. The input of this model are data characterizing targeted scenario. Note the reference computation is performed offline. The position x_{lead} of the lead vehicle at each simulation step is known in advance. An acceleration reference γ^{ref} is computed in output. This acceleration minimizes the difference between current distance D behind the lead vehicle and the expected distance of regulation D^* expressed in terms of a Time Inter-Vehicle (TIV). We use a double integrator to model the vehicle dynamics but the same approach may be instantiated with other kinematic models [13], [14]. The discrete time model of the vehicle is given by the following set of equations:

$$\begin{aligned} x_{i+1} &= x_i + v_i \Delta t \\ v_{i+1} &= v_i + \gamma_i \Delta t \\ \gamma_i &= \gamma_i^{ref} \end{aligned} \quad (1)$$

Where x_i, v_i and γ_i are respectively the position, linear-velocity and acceleration of the vehicle at time t_i .

As some requirements of section III-A are expressed over periods of $1s$ and $2s$, we use vectors in \mathbb{R}^n manipulate sequences of n (n chosen so that $n = \frac{2}{\Delta t}$) values (windows of $2s$). The underlying optimization over sequences of n time-steps can be described as follows:

$$\arg \min_{\vec{\gamma}^{ref}} J = \|\vec{D} - \vec{D}^*\|^2 + P \quad (2)$$

$$P = \lambda_1 \|v_r(\vec{\gamma}^{ref})\|^2 + \lambda_2 \|\partial \vec{\gamma}^{ref}\|^2 + \lambda_3 \|\vec{\gamma}^{ref}\|^2 \quad (3)$$

$$\vec{D}^* = TIV.\vec{v} \quad (4)$$

$$\vec{D} = \vec{x}_{lead} - \vec{x} \quad (5)$$

$$(\vec{x}, \vec{v}) = \vec{f}(\vec{\gamma}^{ref}) \quad (6)$$

$$\vec{D} \geq \max(\vec{2}, TIV.\vec{v}) \quad (7)$$

$$\frac{1}{n} \sum_{i=0}^{n-1} \gamma_i \begin{cases} > -3 & \text{if } (v_0 > 20) \\ > -5 & \text{if } (v_0 < 5) \\ \leq \frac{2}{15}v_0 - \frac{17}{3} & \text{otherwise} \end{cases} \quad (8)$$

$$\frac{2}{n} \sum_{i=0}^{\frac{n}{2}-1} \frac{\gamma_{i+1} - \gamma_i}{\Delta t} \begin{cases} < 2.5 & \text{if } (v_0 > 20) \\ < 5 & \text{if } (v_0 < 5) \\ \leq \frac{1}{6}v_0 - \frac{35}{6} & \text{otherwise} \end{cases} \quad (9)$$

$$\frac{1}{n} \sum_{i=0}^{n-1} \gamma_i \begin{cases} < 2 & \text{if } (v_0 > 20) \\ < 4 & \text{if } (v_0 < 5) \\ \leq \frac{-2}{15}v_0 + \frac{14}{3} & \text{otherwise} \end{cases} \quad (10)$$

$$\vec{v} \leq \vec{v}_c \quad (11)$$

Where the function $\vec{f}(\cdot)$ is a compact representation of equation (1) extended to sequences of n values: $\vec{x} = (x_0, \dots, x_{n-1})$, $\vec{v} = (v_0, \dots, v_{n-1})$, $\vec{\gamma} = (\gamma_0, \dots, \gamma_{n-1})$ and $\vec{\gamma}^{ref} = (\gamma_0^{ref}, \dots, \gamma_{n-1}^{ref})$. The cost function at equation (2) ensures that the obtained position (trajectory) terminates as close as possible to the goal position at defined regulation distance defined in equation (4). In order to obtain a realistic control over the acceleration generated by the reference behavior model, the cost function is penalized with three terms: $\|v_r(\vec{\gamma}^{ref})\|^2$, $\|\partial \vec{\gamma}^{ref}\|^2$ and $\|\vec{\gamma}^{ref}\|^2$ which will minimize respectively: the relative speed w.r.t. the lead vehicle, the difference between two successive accelerations and the acceleration. Parameters λ_1 , λ_2 and λ_3 ensure the weighting of each of the components to be minimized. An empirical exploration resulted in the following estimates: $\lambda_1 = 10^{-2}$, $\lambda_2 = 2.10^{-1}$ et $\lambda_3 = 10^{-3}$. Equation (5) exploits the fact that the position of the lead vehicle $\vec{x}_{lead} = (v_{lead_0}, \dots, v_{lead_{n-1}})$ is known *in advance* at each simulation step. This allows to calculate the actual distance between the reference vehicle and the lead vehicle and compute the whole optimization offline. The rest of equations express the requirements described in section III-A.

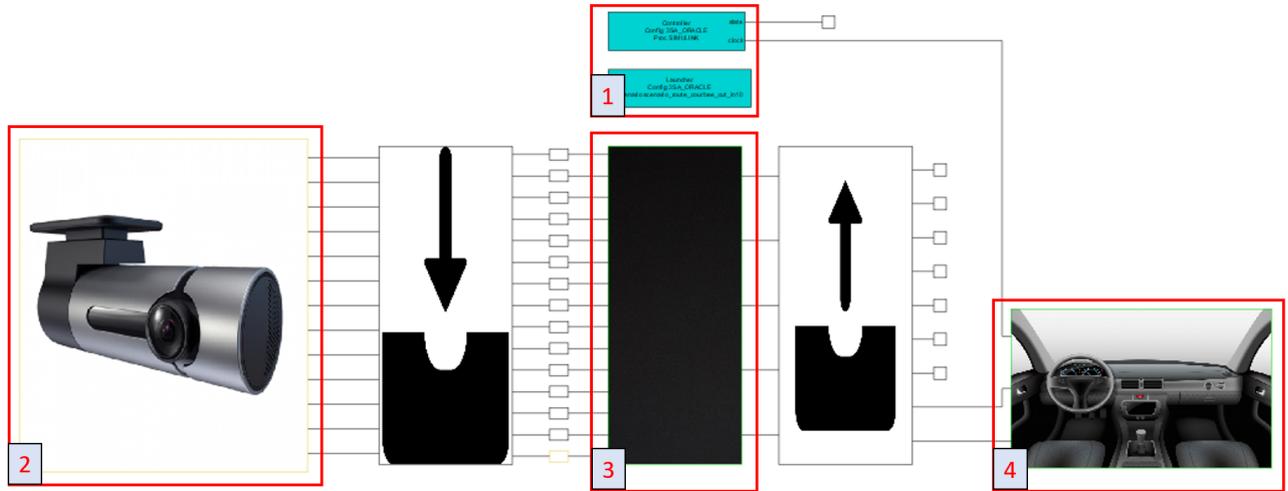


Fig. 2: Simulink model of a black-box control module and its interface with environment simulation.

IV. IMPLEMENTATION AND RESULTS

A. Simulation framework

Simulation relies on a virtual environment with virtual agents to generate knowledge about an ADS's behavior without the need for a physical vehicle and actual testing in the real world. The ADS and the virtual environment in which the ADS will be operating are modeled to the desired degree of fidelity. The higher the fidelity of these models, the more closely they represent the actual nature of the vehicle or environment, which results in more substantive data for analysis.

There are several simulators developed in the academic, industrial and commercial sectors. The main characteristics required in this kind of simulators are:

- Presence of tools and components to model the environment: road, lane markings, mobile objects, etc.;
- Wide range of sensor models: camera, radar, lidar, ultrasound, odometer, etc.;
- Interfaceability (compatibility) with third party software: Matlab/Simulink, etc.;
- Dynamic models of vehicles and ability to integrate models and external components;
- Accessibility to all data and settings of environment, sensors and vehicles.

In the following, we used SCANeR Studio [15] to implement our approach. The approach has a generic interface and may work as well with other simulators such as Pro SiVIC, PreScan, IPG CarMaker, etc.

Developed for automotive experts, SCANeR Studio is designed to meet the specific needs of dynamic simulation professionals. The use of the software was thought, initially, around the process of using driving simulators and was structured around five dedicated modules accessible from the graphical interface:

- *Terrain*: RoadXML road network edition in manual or automatic format via the import of different types of road databases (GPX, XML, OSM, SHP, etc.);

- *Vehicle*: Tool for debugging and studying dynamic models. It has a non-linear dynamic model and can import external models;
- *Scenario*: Driving simulation scenario editor that has its own functions that allow you to define complex simulation scenarios and also allows to import scenarios developed under python.
- *Simulation*: Simulation and simulator supervision tool, coupled with Matlab/Simulink for co-simulation between the two environments;
- *Analysis*: Graphical tool for fine analysis.

Given a Simulink Model of an ADS controller, we use the Simulink library of blocks provided by SCANeR studio to put our ADS control module in realistic driving conditions as depicted in Figure 2. Block 1 allows synchronization between Environment simulation and Simulink model simulation. Input data from environment simulation is provided by block 2. Input data for the ADS control module include identified characteristics of environment static/dynamic objects (e.g. road lanes, distances to surrounding vehicles, etc.) perceived with idealistic sensor models. The ADS control module (block 3) computes an acceleration and a steering commands to keep the equipped vehicle (Ego vehicle) in its lane at a safe distance behind the vehicles ahead in the same lane. These commands are sent back to the environment simulator to close the loop of simulation by updating the behavior of the Ego vehicle (block 4).

B. Testing architecture

The test architecture is illustrated in Figure 3. An iteration of three steps is performed in order to build our automatic testing framework.

a) *Step 1 (ADS simulation)*: Given a scenario specification (e.g. cut-in), the first step is to implement the scenario in the environment simulator. We use SCANeR explore tool to generate variability on parameters of implemented scenarios in SCANeR studio. As described in section IV-A, the ADS controller receives data from the environment simulator. The

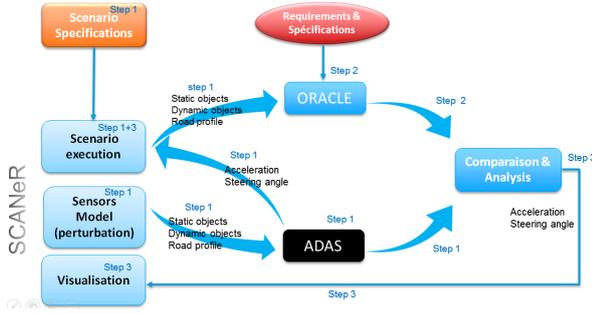


Fig. 3: Testing architecture.

controller makes decision and provides an acceleration and steering angle commands to control the Ego vehicle.

b) Step 2 (Reference computation): In this step we try to build an expected "right decision" if it exists. For this we use the optimization-based model (Oracle) introduced in section III-B. Each scenario is mapped, by two-second sequences, to a reference behavior. The proposed model is non-linear and provides dynamic solutions that simultaneously comply with comfort criteria (acceleration and jerk limit) and safety criteria (ie. collision avoidance).

c) Step 3 (Test verdict): The test verdict is the last but not least step in our test architecture. Unlike [9], we introduce the notion of reference behavior to guide our analyses. The reference behavior computed in step 2 (step-wise acceleration and steering angle) is used in this step to analyze the behavior of the Ego vehicle (equipped with the controller) with respect to the reference behavior.

C. Case study (Cut-in scenario)

In this section, we present a case study to illustrate the method of quantifying the risk for a cut-in scenario. We will first describe the cut-in scenario and the use case.

We want to characterize the risk for cut-in scenarios that are described as follows: while the Ego vehicle drives at a moderate to high speed while staying in its lane and following an initial lead vehicle at a safe distance, another vehicle cuts into the lane of the Ego vehicle, such that this vehicle becomes the Ego vehicle's lead vehicle. The Ego vehicle needs to brake to prevent a collision.

Figure 4 depicts the evolution over 45s of the distance between the Ego vehicle and the current lead vehicle as well as the distance between the reference vehicle (parameterized with TIV = 2s) and the current lead vehicle. Note that the distance to the lead vehicle drops instantly when the cut-in occurs. A simple comparison between the two curves shows that, in this case, the Ego vehicle leaves, in general, less distance behind the lead vehicle than the recommended TIV of 2s.

Figure 5 shows the visualization of this comparison in SCANeR studio. A simple visual comparison may give a first insight on whether the Ego vehicle behaves as expected (i.e. satisfying the requirements for the targeted scenario) by comparing its position to the position of the reference vehicle.

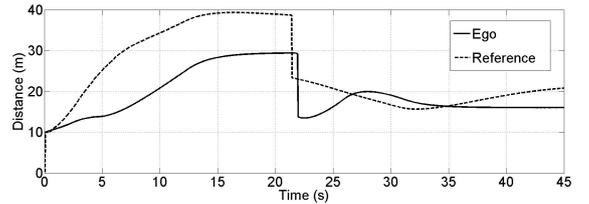


Fig. 4: Distance to lead vehicle in cut-in scenario for Ego vehicle and Reference vehicle at TIV=2s.



Fig. 5: Verdict visualization with SCANeR studio (left: before cut-in (at 3s), right: after cut-in (at 34s)): Ego (black vehicle), Reference at TIV = 2s (white vehicle), Initial lead vehicle (blue vehicle) and Cut-in vehicle (red vehicle).

D. Test case classification

The magnitude of the number of scenarios an ADS could encounter, along with the magnitude and variability of the parameters that make up a scenario (e.g. vehicle speeds, distance between vehicles, etc.), likely present an impractical set of test cases. Our computation of reference behavior offers a systematic approach to classify test cases derived from the variability of scenario parameters. For instance, given the cut-in scenario described in section IV-C, 30 test cases were generated with the SCANeR studio tool by varying the speed of vehicles around the Ego vehicle and the time of occurrence of the cut-in. Table I shows the resulting test cases and the corresponding behavior resulting from the Ego vehicle along with computed references at TIV = 1s, 2s and 3s respectively. A test case is characterized by the distance between the initial lead vehicle and the cut-in vehicle at the time of the cut-in (Δd). A vehicle behavior is characterized by the minimum and mean distance to lead vehicle resulting from vehicle's behavior. The results show that in average the behavior of Ego vehicle is close to the reference at TIV = 2s. This gives a hint that the Ego controller is parameterized with a TIV around 2s. In addition, the computed references are conservative and have a minimum distance to lead vehicle larger than the minimum distance resulting from Ego behavior. For each test case, the reference vehicle and the Ego vehicle have the same initial distance to the lead vehicle (10m). (i.e. the two vehicles are superposed initially).

Reference computation may fail when the optimization constraints can not be satisfied. Note that the bigger the distance between the initial lead vehicle and the cut-in vehicle at the time of the cut-in (Δd), the bigger safety distance is needed to compute a reference. This property

Cut-in $\Delta d(m)$	Ego(m)		[Ref:TIV=1s](m)		[Ref:TIV=2s](m)		[Ref:TIV=3s](m)	
	min	mean	min	mean	min	mean	min	mean
12	10.1	21.2	9.4	15.7	10	21.9	10	28.6
11.8	9.5	19.2	9	14.8	10	19.3	10	24.3
10.3	10.1	23	10	14.9	10	22.5	10	30.2
13.3	10.2	21.7	10	17	10	22.6	10	29.2
15.6	9.9	21.9	-	-	10	25.9	10	33.3
19.5	10	20.7	-	-	10	25.4	10	32.3
18.7	8.4	19.2	-	-	9.7	24.3	9.7	29.3
22.1	9.3	19.8	-	-	10	26	10	32.4
15.3	10.2	24.4	-	-	10	28.5	10	37.4
11.1	9.2	19	-	-	10	18.6	10	23.3
17.7	10.2	21.7	-	-	10	25.6	10	32.9
22.3	8.7	19.7	-	-	10	25.7	10	32.1
19.3	7.9	18.6	-	-	10	24.5	10	29.4
14.1	10	19.9	-	-	10	20.7	10	26.9
14.7	8.5	19.1	-	-	10	21	10	26.2
17.9	9.9	22.1	-	-	10	26.8	10	34.4
14	10	20.9	-	-	9.9	23	9.9	29.6
28.1	9.9	21.6	-	-	10	33.1	10	40.9
25.3	7.6	19.5	-	-	9.9	28	9.9	34.3
23.7	8.8	20.5	-	-	-	-	10	36.4
26.3	10	21.7	-	-	-	-	10	40.8
24.9	6.8	19.5	-	-	-	-	10	33
29.4	7.5	19.2	-	-	-	-	10	37.8
36.5	5.6	20.1	-	-	-	-	10	37.3
38.6	9.9	23.9	-	-	-	-	-	-
35.5	10	24.9	-	-	-	-	-	-
38.2	9.9	25.4	-	-	-	-	-	-
40.1	4.7	18.6	-	-	-	-	-	-
38.1	9.94	23.8	-	-	-	-	-	-
38.6	10	26	-	-	-	-	-	-

TABLE I: Cut-in scenario classification.

can be used to classify and sort the test cases according to their criticality from low to high as shown in the table II. The Ego vehicle regulates its speed according to the speed of the lead vehicle and the distance left behind this vehicle. A sudden drop in the latter at the time of the cut-in (change of lead) can put the vehicle in a situation all the more critical as this drop is significant. If no reference can be computed, no definite conclusion can be drawn but this likely means that no cut-in has occurred or that the vehicle, supposed to perform the cut-in, has changed lane behind the Ego vehicle (parameterized with TIV=2s).

Ref(TIV=1s)	Ref(TIV=2s)	Ref(TIV=3s)	Criticality level
✓	✓	✓	low
✗	✓	✓	medium
✗	✗	✓	high
✗	✗	✗	!

TABLE II: Scenario classification according to references.

V. CONCLUSION AND FUTURE WORKS

A central question is how functional safety of autonomous systems can be guaranteed and tested. The intended system functionality and the necessary safety functions cannot be implemented separately, but are two sides of the same coin. Accordingly, it is not possible to separate the aspects of functionality and safety during testing.

Manufacturers of autonomous systems need procedures and tools by means of which they can test the functionality and safety of such products seamlessly and efficiently and prove them to the approval authorities.

In our work, we propose a scenario-based testing approach of ADS controllers. Scenarios are used to model and describe usage situations. These scenarios are then used as test instructions for testing in simulations.

We described an application of our approach to testing an ACC controller in the case of cut-in scenarios considering

related safety requirements and goals. An optimization-based control model is devised according to these requirements. This allows an offline automatic computation of the expected outcome behaviors with the selected test input values.

We implemented our approach in a tool chain including an environment simulation tool, an automatic expected outcome reference generation tool and test analysis tool. This tool chain helps to classify scenario variants systematically and to evaluate scenarios and tests automatically with regard to safety relevance.

Currently, we are planning to use a more realistic vehicle model in our optimization-based reference computation and plug our tool chain with data-bases of a wider range of scenarios to retrieve input test data. This will allow to contribute to the overall effort of safety evaluation of ADS. Moreover, our current work aims to automatically synthesize ADS controller reference models. This will allow to provide reference behavior for more scenarios but require to find appropriate abstraction level to describe the safety requirements and goals associated to a wider range of scenarios.

REFERENCES

- [1] IEC 61508: 2010, "Functional safety of electrical/electronic/programmable electronic safety-related systems (part 1-7)."
- [2] ISO 26262:2018, "Road vehicles – functional safety."
- [3] ISO/PAS 21448:2019, "Road vehicles – safety of the intended functionality (sotif)."
- [4] M. Wood, "Safety first for automated driving," tech. rep., 2019.
- [5] T. Linz, "Testing Autonomous Systems," in *Futur. Softw. Qual. Assur.* (S. Goericke, ed.), pp. 61–75, Cham: Springer International Publishing, 2020.
- [6] S. Ulbrich, T. Menzel, A. Reschka, F. Schuld, and M. Maurer, "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving," in *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2015-October, pp. 982–988, IEEE, 2015.
- [7] ASAM OpenSCENARIO, 2020. <https://www.asam.net/standards/detail/openscenario/>.
- [8] W. Chen and L. Kloul, "Stochastic modelling of autonomous vehicles driving scenarios using PEPA," in *Model-Based Safety and Assessment - 6th International Symposium, IMBSA 2019, Thessaloniki, Greece, October 16-18, 2019, Proceedings*, vol. 11842 of *Lecture Notes in Computer Science*, pp. 317–331, Springer, 2019.
- [9] R. Lattarulo, J. Pérez, and M. Dendaluce, "A complete framework for developing and testing automated driving controllers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 258–263, 2017.
- [10] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, vol. 19. Siam, 2010.
- [11] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.
- [12] ISO 22179:2009, "Intelligent transport systems - full speed range adaptive cruise control (fsra) systems - performance requirements and test procedures."
- [13] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 812–818, IEEE, 2017.
- [14] M. Babu, R. R. Theerthala, A. K. Singh, B. Gopalakrishnan, K. M. Krishna, et al., "Model predictive control for autonomous driving considering actuator dynamics," *arXiv preprint arXiv:1803.03478*, 2018.
- [15] SCANer studio, 2020. <https://www.avsimulation.fr/solutions/#studio>.