IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 July 2023, accepted 14 August 2023, date of publication 18 August 2023, date of current version 24 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3306602

RESEARCH ARTICLE

A Two-Layered Approach for the Validation of an Operational Autonomous Shuttle

MOHSEN MALAYJERDI[1,2], QUENTIN A. GOSS[3], (Graduate Student Member, IEEE), MUSTAFA İLHAN AKBAŞ[3], (Member, IEEE), RAIVO SELL[1,2], (Member, IEEE), AND MAURO BELLONE[2], (Member, IEEE)

[1]Department of Mechanical and Industrial Engineering, Tallinn University of Technology, 19086 Tallinn, Estonia
[2]FinEst Smart City Centre of Excellence, Tallinn University of Technology, 19086 Tallinn, Estonia
[3]Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

Corresponding author: Mohsen Malayjerdi (momala@taltech.ee)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under Grant 856602, in part by the European Regional Development Fund, and in part by the Estonian Ministry of Education and Research under Grant 2014-2020.4.01.20-0289.

ABSTRACT To embrace safety while bringing autonomous vehicles (AVs) to public roads, AV manufacturers need to validate and verify the functionality and reliability of the control software. Real-road testing is time-consuming, tedious, costly, and unsafe for validation. Hence, simulation testing has been playing an important role in the market as a viable solution. This paper presents an approach that exploits both methods to find edge case scenarios and evaluates the software reliability of an existing AV shuttle, iseAuto, currently operating at the Tallinn University of Technology campus. To show the method's effectiveness, a range of scenarios are generated and simulated for avoidance maneuvers by means of a low-fidelity simulator. Then, the scenarios that are found to be jeopardizing the AV are filtered and simulated by a high-fidelity simulator with the AV control software in the loop. Finally, to investigate the methodology and simulation reliability, a real study case is proposed using the AV shuttle. Results of the study suggest that the proposed toolchain is capable of tuning simulation models for automated driving development as well as validating safe AV operations.

INDEX TERMS Autonomous vehicles, scenario testing, safety validation, SiL testing, simulation.

I. INTRODUCTION

Autonomous vehicles (AVs) are expected to reduce traffic jams, boost mobility, and produce more sustainable and safer transportation. Despite the studies on considerable uncertainties towards AVs adoption [1], [2], various novel technologies have been in development for AVs in recent years to ensure safety and gain public trust [3], [4], [5]. However, the methods and tools for evaluating and validating such evolution still need more attention. Studying all incidents in which AVs are involved [6], [7], including the death of a pedestrian in 2018 [8], reminds us that the testing in different conditions cannot be ignored if the goal is the pervasive deployment of AVs on public roads [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez.

One of the best examples of AVs in the public transportation sector is autonomous shuttles that have been in operation in restricted areas for the past few years. These shuttles are effective and clean mobility solutions. However, researchers and engineers are trying to find and eliminate these vehicles' vulnerabilities in operations and maneuvers by putting them under the test. Using an innovative and effective validation and development toolchain, this research evaluates the safe passing maneuver of an operational autonomous shuttle, iseAuto, developed by the AV research group at the Tallinn University of Technology (TalTech), Estonia (see Fig. 1). The iseAuto project's objective is to build an open-source AV shuttle and establish a smart city testbed [10], [11], [12] in the TalTech campus so that different types of projects on the future of urban mobility can be conducted in this environment. Currently, this SAE level 4 and 5 shuttle is operating

on the campus for experimental and study purposes [13]. The passing maenuver is the basis of overtaking; one of the challenging operations that low-speed shuttles face [14]. To demonstrate the effectiveness of the proposed validation regime, this maneuver was chosen as a use case and a sample for implementing other testing scenarios.

FIGURE 1. TalTech iseAuto - an AV shuttle.

Open-road, closed-track testing, and simulation are the three main strategies for examining AVs. However, the first two are considerably costly, slow, labor-intensive to organize [15], [16] and are far too broad to test comprehensively [17]. Furthermore, in real-world testing, the conditions are not always easy to repeat and, in some cases, the safety of the involved actors can be jeopardized. On the other hand, the simulation strategy accelerates experimentation and enables us to test highly regulated scenarios without any safety concerns. It is fast, repeatable, and scalable [18], [19], [20]. For safety assessments and ''stress testing'' of autonomous algorithms, simulations can generate comprehensive databases and achieve the statistical power required [17]. Despite the advantages of simulations, Open-road and closed-track remain indispensable before deployment. To utilize these advantages of the simulation and digital testing, iseAuto and the testing environment are connected to its digital twin, which enables running all developed features first in simulation. The simulation environments, interfaces, and concepts are described in detail in [21] and [22].

There are various simulators available for AVs including commercial and open-source tools [18] that can execute low and high-fidelity simulations. Low-fidelity simulation, which imitates the actual scenario but leaves out detailed factors, is useful for the primary evaluation for quick processing. In contrast, high-fidelity simulation attempts to be realistic for chosen characteristics of a validation scenario and includes many features suitable for software-in-the-loop (SiL) testing. The scope of this paper lies in utilizing both methods in series as a comprehensive validation toolchain.

Microscopic simulators have been designed and developed to model traffic and handle large networks with an optimal speed [23]. These open-source platforms enable us to create various scenarios including actors configured with different properties for low-fidelity simulation purposes. However, they suffer from a lack of abilities that would make them eligible to be used as a standalone AV validation platform. There are well-known and powerful end-to-end simulators based on game engines among the open-source platforms including SVL by LG and CARLA [24], [25]. Highly detailed 3D environments, various virtual sensor types, and realistic vehicle dynamics allow these tools to be used in reliable validations. Still, there are some basic challenges to be overcome, including defining precise validation metrics for the AV evaluation and developing efficient tools to generate test scenarios [26]. The proposed method in this study creates a platform to generate scenarios in a passing mission and then evaluates the AV control algorithms' performance in that mission by utilizing a state-of-the-art simulator.

The contributions of this paper are as follows:

- The integration of the scenario-based low and high-fidelity simulation into the overall field of safety assessment.
- A scalable and efficient methodology to identify low-priority and impractical scenarios before performing time-consuming simulations (high-fidelity).
- A software-in-the-loop (SiL) demonstration of the methodology featuring TalTech's IseAuto AV shuttle.
- Implementing the proposed methodology in a highly safety-critical maneuver to investigate the performance.
- Testing the fidelity of the proposed methodology with a real-world experiment involving the highly autonomous shuttle.

The remainder of the paper is organized as follows. Related work is presented in Section II, then our approach is described in detail in Section III. In Section IV, the methodology is demonstrated by a simulation study, and results are provided. Following this, we present an experimental case study in Section V. We also discuss the results, limitations, and future work of the study in Section VI and conclude in Section VII.

II. RELATED WORK

The complexities of AVs as a Cyber-Physical System (CPS) render them crash-prone and vulnerable [27]. However, validation and verification of AI-controlled AVs is a critical challenge, and considerable effort has been directed towards providing safe autonomous systems [28], [29]. Thus far, mainly real-life experiments and simulations have been utilized to find safety flaws and performance limitations [30], [31]. It is important to note that despite the advantages of simulation, it is not feasible to conduct all tests purely in a virtual environment. For instance, virtual sensor technology still needs to be developed and has not matured [32]. AI-based driving algorithms constitute a core area of development for autonomous driving, for additional information about the topic the reader might refer to [33] and [34].

Kalra and Paddock [9] suggested that over 11 billion miles will have to be driven by AVs to verify that they are safer than human drivers. Test miles are not, by themselves, a good measure of AV's safety. In the future, it may also be necessary to repeat these driving miles due to software changes. Instead, the types of tests that they undergo during testing are determinant. There are currently several safety standards for the

automotive industry including ISO 26262 [35] and ISO/PAS 21448 ''Safety of the Intended Functionality'' (SOTIF) [36]. As of yet, there is neither a consensus nor a standard procedure for testing and evaluating AVs [37]. Koopman et al. [38] introduced a safety standard approach for highly autonomous vehicles based on setting scope requirements for a safety case. Furthermore, Koopman and Fratrik [39] listed factors that should be addressed in the area of operational design domain (ODD, e.g. scenarios) and vehicle maneuvers to validate the system. These papers, along with many others [40], [41] in the field of AVs, underscore the importance of rigorous testing, simulation, and real-world validation to ensure the safety and reliability of AVs before they can be deployed on public roads.

In [42], authors implemented Hazard Based Testing (HBT) by exploiting Systems Theoretic Process Analysis (STPA) to create test scenarios for the Unsafe Control Actions (UCA) of an automated driving system. Although they did not test or simulate the resulting 3000 test scenarios to investigate the failures and flaws of the system, they argued that their systematic STPA approach is more effective in finding test scenarios that would reveal actual weaknesses or flaws in the system compared to the random scenario generation method. In [43], Gelder et al. stated that employing only real-world road traffic scenarios for the AV examination is not adequate. Instead, they suggested a technique for determining the parameters that characterize the real-life originated scenarios to a sufficient extent reliable for evaluation, at the same time relying less on strong assumptions on the parameters that characterize the scenarios in the first place.

Hallerbach et al. [44] introduced a generic simulation-based toolchain to determine and verify critical scenarios for AVs. They utilized a traffic simulator coupled with a vehicle dynamics simulator to flag safety-critical cases and exploit the test results for automation functions development of an SAE level 3 car. However, their method finds cases randomly to evaluate the criticality, and this can be inefficient in the case of high-fidelity simulators. Similarly, 17 industrial and academic partners worked together in the PEGASUS project to find new standards and validation methods for the highly self-driving functions [32]. The project partners developed a scenario generation regime that produces scenarios in different levels of abstraction. Then, these scenarios were tested in the simulation (SiL and HiL), and verified and validated on test grounds and in field tests. By deploying naturalistic driving data and introducing adversarial behavior into NPCs, Feng et al. [37] presented a novel testing methodology. They concluded that this initiative would accelerate the evaluation process significantly. The authors, however, did not consider any other criteria reflecting the performance of the AV algorithms, settling only on crash-based critical violations as a measure of criticality. Further, no real-world tests were conducted to determine the validity of their proposed method. In a review study [45], Rosique et al. explored perception systems and their simulations. They described different types of simulators including model-based, game engine-based, robotics field-oriented, and ones designed specifically for AVs.

This study [6] extracted the specific features of traffic accidents with AVs. Even though their sample of traffic accidents was limited, the summarized report should be taken into consideration, especially when creating scenarios to prevent such failures in the future. In [46], authors proposed a method to generate concrete AV validation scenarios based on historical fatal accident data. First, they filtered and removed the redundant scenario components, and then the pruned cases were prioritized by severity levels according to the fatality ratio. As a continuation, they improved the validation effort efficiency by significantly reducing the sample space of the utilized datasets [5], [47]. Also, in [48], they exploited the current AV crash records and formulated them into modular and measurable scenario units by employing the Measurable Scenario Description Language (M-SDL). The proposed technique produces modular scenario units with coverage analysis and identifies edge scenarios using AV evaluation metrics.

Overall, our approach differs from previous efforts by introducing a state-of-the-art toolchain that evaluates a real AV shuttle's safety in desired ODD and maneuver scenarios. While most of the earlier work focused on using a single low-fidelity simulation method or multiple methods separately as their evaluation tools, we take the initiative in exploiting both low and high-fidelity simulation platforms and coupling them in a progressive approach to increase efficiency and reliability.

III. DEVELOPMENT OF THE VALIDATION REGIME

Our proposed approach can be summarized in three main hierarchical steps as shown in Figure 2. It starts with a scenario description block (step-A) to prepare concrete scenarios for testing. In the next step, those scenarios defined in a JSON format are simulated within a low-fidelity traffic simulator (SUMO, step-B). During step B, the device under test (DUT) is controlled according to the rules defined in the scenario setup without considering other non-player characters (NPCs). The simulated scenarios are then analyzed, filtered, and translated into a CSV format for an end-to-end high-fidelity simulation (SVL, step-C). In the last step, the DUT is tested within a naturalistic simulated driving environment while being controlled by the exact software (Autoware.ai) used on a real operational autonomous shuttle. A more detailed description of the method can be found in Figure 3. The ''AV Black Box'' block inside step C is

FIGURE 2. Three main steps of the proposed validation method. The format of each signal passing among these steps is annotated.

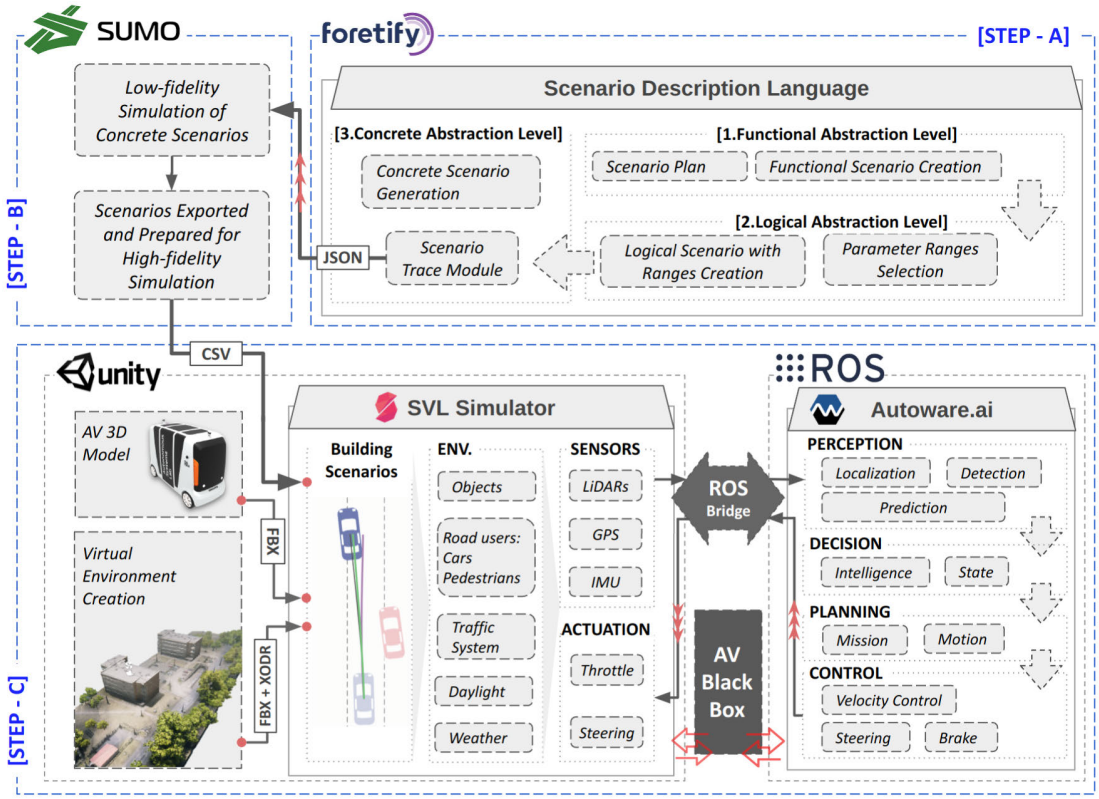

FIGURE 3. High-level architecture of the scenario generation, simulator, and AV control system. The low-fidelity simulation is represented by the SUMO block, while the SVL simulator and the Autoware control system can be considered as one high-fidelity simulation block. Please refer to Section III for further details.

designed to record all the necessary data for validation later based on desired metrics.

A. SCENARIO PLANNING AND FORMALIZATION

The cycle of an AV validation scenario begins with scenario planning. The plan is typically described using a scenario description language such as SCENIC [49] or MSDL [50]. With the help of the scenario description language, desirable ODDs, such as scenarios, maneuvers, and road and weather conditions, can be formally defined.

Its description is transformed from functional to logical and then to a more concrete abstraction level with the minimum required parameters to describe the actions of each actor in the scenario (see Figure 3, step-A, scenario description language box).

To start, the plan is constructed without considering the limitations or quirks of simulation in a human-readable form. The goal of the scenario and its requirements are discussed between the low-fidelity simulation group and the high-fidelity simulation group which results in the creation of a functional description. In our implementation, we use MSDL and Foretify™ [51] to describe scenarios. Once the functional description of the scenario is created, both groups begin preparing their simulators and the desired parameters, and then the safety evaluation metrics are determined. Based on these metrics, parameter ranges are selected and a logical scenario with parameter ranges is produced.

The generated logical scenario is used as the template in which concrete values are selected from each parameter range. Selecting unique combinations of parameters produces unique concrete scenarios.

Scenario description languages allow AV validation scenarios to be formalized in a way which is reproducible and shareable. These formalized scenario descriptions are ideal for storing and sharing abstract and logical scenarios. A scenario description language may also share configurations of concrete scenarios, but the structure of the testing environment and results must be tailored to the application. For this approach, the set of concrete scenarios, their configurations, and testing results are formalized in equation notation. This is an adaptation of the scenario formalization used in the survey from Mullins et al. [52], which correlates a scenario configuration→result of black box unmanned underwater vehicle tests in order to visualize boundaries and objectives of the physical scenario space. In the architecture of this paper, AV black box testing is performed at two levels of low and high-fidelity simulation. In order to evaluate and compare the low and high-fidelity simulations, the scenario limits and input–state→score relation are described formally.

The **scenario configuration space** $\mathcal{X}^n = [\mathcal{X}_1, \ldots, \mathcal{X}_n]$ is composed of n elements. Each element in the state space

vector represents a parameter with a range in the plan of the scenario. For example, if a logical scenario contains five (5) parameter ranges, then $\mathcal{X}$ is composed of five (5) elements which are the limits of each parameter range.

The **scenario input state** is defined as a vector $X = [x_1, \ldots, x_n]$ where $\forall i \in n : x_i \in \mathcal{X}_i$. Each input state is one scenario configuration where concrete values are sampled from the parameter ranges in the plan. A sample set of N states is defined as $X^N = [X_1, \ldots, X_N]$. Each element of X is a concrete value and the size of X is the same as $\mathcal{X}$.

The **score space** $\mathcal{Y}^m$ has m parameters where each output score is defined as the vector $Y = [y_1, \ldots, y_m]$. Each element in the score vector is a metric where the system is evaluated. A sample set of N states is defined as $Y^N = [Y_1, \ldots, Y_N]$. For example, if two (2) metrics are used to evaluate a scenario, then $\mathcal{Y}$ is comprised of two (2) elements. Y is the same size as $\mathcal{Y}$.

A **DUT function** for the iseAuto is $\mathcal{F}(X^N) = Y^N$ which accepts a set of N input states X^N and returns a sample set of N score vectors Y^N. For example, if 100 scenario tests are performed then $N = 100$, and both X and Y would contain 100 elements ($[X_1, \ldots, X_{100}]$ and $[Y_1, \ldots, Y_{100}]$) which correspond to the 100 tests.

Table 1 reports an example of a scenario's required parameters at the functional level. Dx and Dy, respectively, are the longitudinal and lateral initial relative distances between DUT and NPC in each scenario. The requirements are selected based on our testbed limitations.

TABLE 1. Target scenarios definition.

Actor	Speed	$[D_x, D_y]$	Goal
DUT	[1-15]km/h	[0,0] m	To overtake the NPC safely
NPC	0	[5-50,Dy] m	To stay immobile

Each row of the table indicates an actor playing in the scene, the TalTech AV shuttle which is the DUT, and a passenger car, i.e. an NPC. The speed range for the DUT is 1-15 kilometers per hour (km/h). The NPC is parked and immobile at the front of the DUT. At the start of the scenario, the NPC is between 5 to 50 m far from the DUT along the road. In addition, there is a small lateral shift, Dy, which is defined by the scenario generator. The goal of the DUT is to safely maneuver around the parked NPC and continue along the road. A simulation is successful when the DUT safely passes the parked NPC and is back in the original lane.

B. LOW FIDELITY SIMULATION

The first level of abstraction in our approach is the low-fidelity simulation. The concrete scenarios are run in these simulations, while information about the scenarios is collected at runtime and consolidated after scenario completion. In this step, we use the SUMO traffic simulator to run concrete scenarios (see Fig 3, step-B). SUMO is selected as the low-fidelity simulator for the following reasons:

- The street network-based approach allows for high performance, even at a very large scale [53].
- The default "no-collision" vehicle control requires no configuration.
- The ability to define a new vehicle control logic, i.e. one controlled by foretify™.
- An optional and minimal graphical user interface that is useful for debugging and presentation.
- Actors obey the network rules.

The low-fidelity simulations are fast and scalable. One important purpose of them is to identify, and filter out, the scenarios which are obvious failures. This results in more efficient utilization of the high-fidelity simulation, which is more computationally demanding compared to abstract simulation.

The scenario is described at a logical abstraction level with parameter ranges for the relative position of the DUT from the NPC at the start and end of the scene, and the speed of the DUT. Figure 4 describes the relative positional measurements of the scenarios.

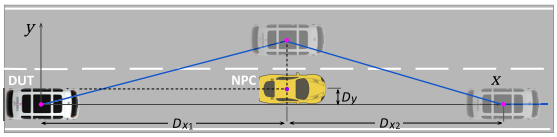

FIGURE 4. Two relative positions are given, Dx_1 and Dx_2, define the NPC position in each scenario. Dy is determined by Foretify.

The distances Dx_1 and Dx_2 are provided as parameter ranges. There is a warm-up period of the simulation, as such some measurements are taken after simulation time 0 such as measurement Dy, which is calculated when the scenario begins. Parameter ranges and descriptions are listed in Table 2.

TABLE 2. Low-fidelity scenarios parameters.

Parameters	Range	Description
s	[1-15]km/h	The speed of the DUT during the scenario
D_{x1}	[5-50](m)	The DUT starts distance behind the NPC
D_{x2}	[5-50](m)	The DUT finish distance ahead of the NPC
D_y	[-0.4-0.4](m)	The small lateral shift for the NPC

A suitable straight path for the mission is selected in the street network map, shown in Figure 5, when the concrete scenario is generated.

Then values are selected from the parameter ranges (see Table 1) to generate the concrete scenarios. The low-fidelity tests are defined as follows: A scenario input state for a single test is $X = [s, Dx_1, Dx_2]$. The score vector for a configuration is $Y = [$ Pose(x, y), Collision$]$, which includes two metrics:

- The position of the DUT during the simulation.
- Whether or not a collision is observed.

Once all abstract simulations are complete, the result is exported and analyzed. The low-fidelity simulations provide rapid testing and debugging, allowing for quick turnaround when the scenario is edited at the functional or logical

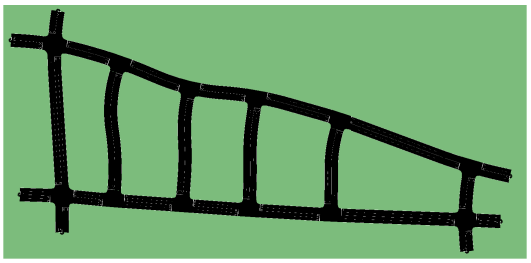

FIGURE 5. The street network.

abstraction stage as AV testing requirements change. When actors behave as intended in the low-fidelity simulations and the decisions of the actors are logical, they can be exported and converted for the iseAuto simulations with more diversity and complexity involved in a realistic SiL simulation.

C. HIGH-FIDELITY SIMULATION

Understanding the environment that AVs operate in has been one of the biggest challenges of their development and deployment [54]. In this context, end-to-end simulations provide a platform to investigate these challenges in detail. In this step, we deployed a high-fidelity simulator to analyze the DUT behavior in the pre-simulated scenarios while the AV software controls the DUT (see Fig. 3, step-C, SVL simulator box).

Selected scenarios assessed by the previous step are the primary input imported into the SVL scenario builder. Figure 6 shows the parameters needed to define the position of the NPC relative to the DUT, resulting in different scenarios in the high-fidelity platform. Dx and Dy represent the longitudinal and transverse distances relative to the DUT, which are Dx_1 and Dy in the low-fidelity scenario configuration.

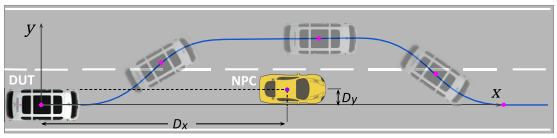

FIGURE 6. Two relative coordinates, *Dx* and *Dy*, define the NPC position in each scenario.

A Python script reads the scenarios list, then imports them to the simulator and executes them one after another. To increase the fidelity of the simulation, we deploy the digital copy of both the real AV and a similar environment to the area of the operations. We created a virtual copy of the iseAuto and defined the same sensor configuration as shown in Figure 7. The shuttle kinematics and dynamics are mimicked inside the simulation for more accurate and reliable evaluation results. It is worth mentioning that iseAuto utilizes a LiDAR-based perception. Two Velodyne LiDARs are installed at the top front (VLP-32) and back (VLP-16) of the vehicle, in addition to two Robosense RS-Bpearl at both sides (left and right), to decrease the sensor blind zone around the car. Furthermore, one RS-LiDAR-16 is installed at the front bumper to detect small objects in front of the vehicle that

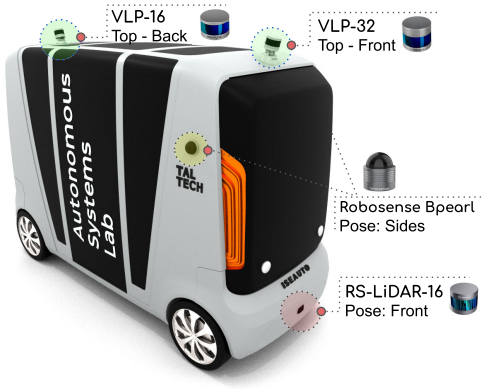

FIGURE 7. iseAuto simulated model with different LiDARs installed.

is not in the other LiDARs' field of view. Processes such as calibration, filtration, and concatenation are performed on the LiDARs' point cloud for a better and optimized perception.

A realistic virtual environment containing urban details and vegetation is one of the required elements for a high-fidelity and accurate evaluation. We demonstrated how to build the 3D virtual environment for the AV simulator by utilizing aerial drone images in [21]. In this study, we use a similar virtual environment to the real world where the AV operates.

Once these elements are initialized, the test platform is ready to run a simulation. This provides virtual sensor data to the perception algorithms and, conversely, receives control commands from the control algorithms (see Fig. 3, step-C, ROS Bridge). The high-level software architecture of the shuttle is based on the Robot Operating System (ROS). Perception, detection, and planning are performed by Autoware.ai [55] (Fig. 3 ROS box), an open-source ROS-based stack for autonomous driving, in which many advanced algorithms are present, including, but not limited to, lane tracking, obstacle avoidance, traffic light detection, and lane detection. All virtual sensor data is transmitted to the software side via a ROS bridge connection. In the perception algorithms, the data is processed, and after the result is processed by planning algorithms, control commands are issued and sent back to the simulator for actuation. The path planning algorithm used in this work is a modified sigmoid planner developed in [14].

Another important element on this platform is the recorder. During each run, the information needed for later analysis is recorded, e.g., speed, position, orientation of the actors, etc. This also allows us to monitor and verify the performance of each algorithm in the control software, e.g., localization and detection. We then review this data against the safety criteria to find safety breaches.

The high-fidelity tests are defined as follows: A scenario input state for a single test is $X = [Dx, Dy]$. The score vector for a configuration is $Y = [\text{EgoSpeed, Brake intensity, DTC, NDTscore, Collision}]$. These metrics are explained in table 3.

Algorithm 1 illustrates the process of importing, running, and recording the required data. A list of desired scores, Y_i,

TABLE 3. Safety and Performance metrics utilized to evaluate the maneuver.

Safety Metric	Description
Collision	Collision between DUT and other objects
DTC	Distance from the ego to the NPC (collision)
Performance Met.	
Brake intensity	Normalized braking magnitude during the mission
NDT score	The localization matching score during the mission
EgoSpeed	The DUT speed
Travel distance	The distance that DUT traveled in the mission
Steering angle	The DUT steering command during the mission

was recorded while using the SVL simulator to run scenarios, X_i, selected after the low-fidelity simulations. This vector contains both safety and performance metrics including collision occurrence, distance to the NPC, normalized brake intensity, localization score, ego speed, DUT traveling distance, and DUT steering command (see Table 3). Distance to collision (DTC) is the minimum distance in meters between actors' bodies at any point in the scenario. The normalized braking magnitude in each mission expresses the driving comfort. Hard brakes result in discomfort for passengers and increase the likelihood of an accident during an operation. The NDT score is a result of a 3-dimensional normal distribution function, implemented in the Point Cloud Library, calculating alignment error between the input laser scan and the reference point cloud map [56]. In terms of performance, the travel distance is an indicator of how far the DUT has progressed in its mission. Finally, the steering command reflects the smoothness of the navigation and steering equipment.

Algorithm 1 Importing and Running the Filtered Scenarios

1: **input:** Selected Scenarios Input $X^N = [X_1, \ldots, X_N]$
2: **output:** Score Vector $Y^N = [Y_1, \ldots, Y_N]$
3: **procedure** RunScenarios(X^N) **do**
4: **for** $\forall i \in n : X_i \in X^N$ **run SiL Simulation**($\mathcal{F}$)
5: $\mathcal{F}(X_i) = Y_i$ = [EgoSpeed, Brake%, DTC, NDT-score, Collision]
6: **end**
7: **return** Y^N
8: **end procedure**

Figure 8 shows the SVL simulation (top images), while the AV software data including the map, trajectory, and perceived point cloud are displayed in the RViz visualization software (bottom images).

IV. SIMULATION RESULTS

In this section, we present a quantitative analysis by applying the proposed platform to the validation of the AV shuttle in a passing maneuver. First, we present the result of the low-fidelity simulation performed with the scenarios proposed by Fortify. We then nominate some of the scenarios for the high-fidelity simulation to evaluate the AV software's behavior by monitoring the proposed metrics.

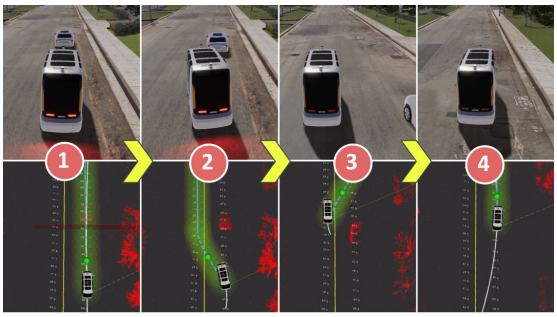

FIGURE 8. One simulated scenario shown in different frames. The top frames represent the SVL simulator and the below one displays the RViz visualization software receiving simulated data.

A. LOW FIDELITY

The platform generated 120 unique scenarios represented by the NPC location in Figure 9 which displays each simulation result in the "Failure" and "Success" groups. The scenarios are divided into three ranges by their longitudinal NPC position to show the probability of failure in different areas.

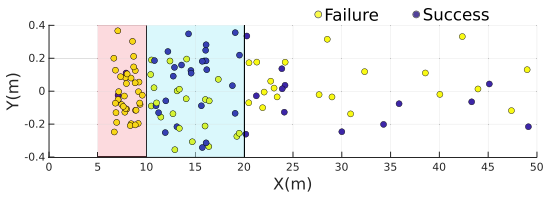

FIGURE 9. Points representing all initial relative NPC locations in 120 scenarios that are marked based on their simulation result.

Table 4 summarizes the number of scenarios in two main groups in the subdivided areas. According to the table, almost 95% of the scenarios generated in the [5-10] m region failed. The failure likelihood decreased to near 46% for the [10-20] m interval. In addition, 47% and 28% of all failures occurred in [5-10] m and [10-20] m, respectively.

TABLE 4. Number of Failure and Success scenarios in different D_x distance.

	[5-10] m	[10-20] m	[20-50] m	sum
Success	2	26	13	41
Failure	37	22	20	79
All	39	48	33	120

At this point, we select 87 scenarios in the range of [5-20] m for further investigation. The reasons for this are: first, more failures are observed before 20 m, and second, it is impractical for the shuttle to begin the passing operation over 20 m distance from the NPC.

Figure 10 shows the paths of the DUT in the low-fidelity simulations. The simulations are separated into two groups:

1) Simulations where a collision between actors occurs, causing the simulation to end.
2) Simulations where the passing scenario completes successfully.

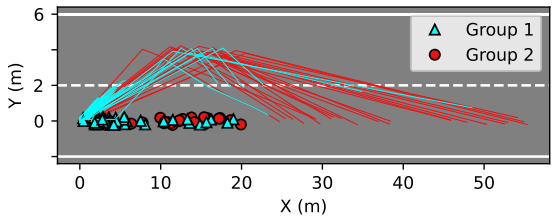

FIGURE 10. All routes traveled in the Foretify simulation.

The collisions occur during lane-change maneuvers where there is an insufficient distance for the DUT to safely traverse around the NPC. The two simulations in group 1 with a distance $x > 30\,m$, observe collisions where the DUT travels some distance while decelerating after the collision.

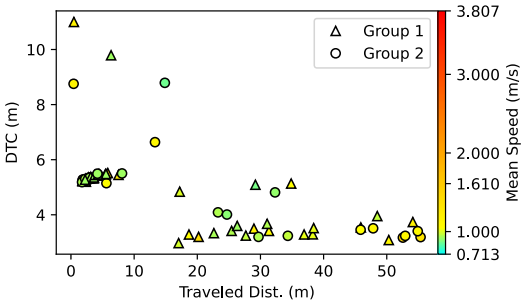

FIGURE 11. Results of the low-fidelity simulations.

Figure 11 uses the same simulation grouping as in Figure 10. The color of the marker shows the mean speed of the DUT. The minimum mean speed is $0.713\frac{m}{s}$ or 1 kilometer per hour. The average mean speed in the scenarios is $1.610\frac{m}{s}$. DTC near 3.00 m means that the actors are side-by-side in adjacent lanes.

TABLE 5. Summary over 87 runs in a low-fidelity simulator.

	duration (sec)	Dy (m)	Dx (m)	$\max(\bar{s})$ (m/s)	$\min(\bar{s})$ (m/s)	DTC (m)
mean	36.21	0.15	8.19	1.80	1.61	4.88
std	35.69	0.09	5.82	0.82	0.62	1.49
min	2.72	0.01	0.41	1.01	0.71	2.97
0.25%	64.40	0.08	2.93	1.17	1.12	3.51
0.50%	14.12	0.13	5.61	1.57	1.52	5.25
0.75%	76.34	0.20	13.43	2.06	1.93	5.40
max	98.48	0.37	19.92	4.16	3.81	11.00

Table 5 gives a summary of the results for the 87 low-fidelity simulations. These include the duration of simulations in seconds, the difference in lateral distance and longitudinal distance of the NPC to the DUT on the road in meters, the average speed of the DUT in meters per second, and the closest distance between actors at any point in the simulation.

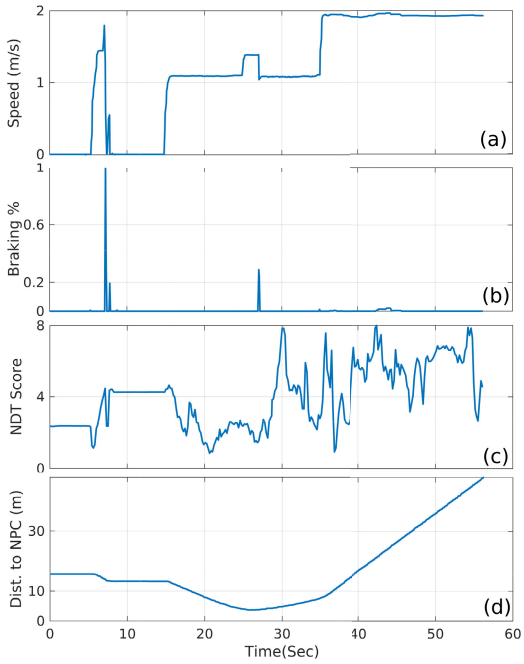

FIGURE 12. The result of a filtered scenario simulation; (a) Speed of the DUT, (b) normalized braking pressure, (c) NDT score for localization performance, and (d) distance measured from Ego to NPC.

B. HIGH-FIDELITY

In this step, we simulate the 87 scenarios in the SiL high-fidelity platform. During the process, we observe all the corresponding data of the evaluation metrics and store them in a rosbag file in addition to a general tabular report. Figures 12 and 13 represent the values of the metrics (see Table 3) recorded during the simulation of an example scenario. Fig. 12 shows (a) the DUT speed, (b) normalized braking intensity, (c) localization score (NDT-score), and (d) the closest distance to the NPC from Ego during the simulation.

The speed chart in the inset of Fig. 12(a) explains how fast the shuttle traveled the route and where it stopped, accelerated, or decelerated. In this case, Ego had an average speed of 1.15 m/s and reached a 1.96 m/s maximum speed. The normalized braking intensity displayed in the inset of Fig. 12(b) shows the moment that the DUT took intense brake during the mission. In the 8th second of the operation, the DUT took an intense brake that made it stop. The effect of the brake can be clearly seen in the changes in speed. The inset of Fig. 12(c) displays the NDT matching score during the mission. This number indicates the accuracy of localization during driving. The higher the numbers, the less accurate the localization is. Loss of vehicle localization may result in unpredictable behavior. Finally, the distance to collision (DTC), which is one of the main parameters monitored during the simulations (see the inset of the Fig. 12(d)), shows how close the Ego vehicle was to the NPC (body to body) in the mission. Among all scenarios, the nearest distance was 0.36 m, while the farthest one was 6.02 m measured from the DUT body to the NPC body.

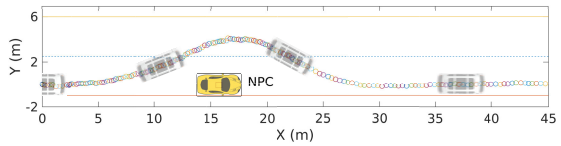

FIGURE 13. Simple representation of the Ego traveling route and the NPC position during the scenario.

It was also necessary to record the trajectory followed by the DUT in each scenario to study the performance of the passing operation and to track the behavior of the DUT in case of a violation of the safety metrics. To this end, we created a track graph for each simulation, as shown in Fig. 13. In this figure, the circle markers represent the track that the DUT follows on the road, and the rectangle shows the position of the NPC.

In the next figure (Fig. 14), a spaghetti diagram shows all trajectories traveled by the DUT (curves) next to the location of the NPC (squares) in each scenario. Depending on the progress of the mission, the results were divided into three groups as follows:

- Not started missions (group 1): The scenarios in which the DUT could not start the passing maneuver, and stayed behind the NPC.
- Completed missions (group 2): The missions are finished by the DUT as expected in scenarios.
- Aborted missions (group 3): scenarios that the DUT has started the maneuver but could not finish it. For instance, losing localization can cause uncontrolled movements that fail the mission.

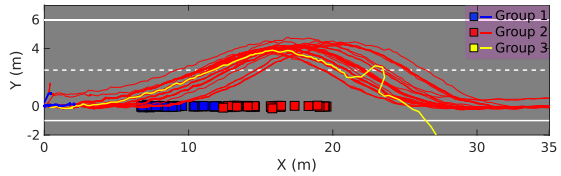

FIGURE 14. All traveled routes in the 87 selected scenarios are shown and divided into three groups as described.

According to the diagram, the scenarios in which the NPC was closer than 12 m belong to the first group. The DUT control software couldn't generate a safe trajectory, as shown by the traveled tracks in group 1. In the other cases, it passed the NPC (group 2), except for the case in which the DUT lost its localization (group 3) and the mission was aborted as the DUT hit the sidewalk.

To check the mission progress and safety, we marked each scenario in Figure 15 with the corresponding distance traveled and the minimum DTC. We then assigned a color to each circle (scenario) based on its average speed and clustered all 87 scenarios. Overall, they were divided into three groups based on their DUT average speed and distance traveled: 50 scenarios with a speed less than 0.05 m/s (G1), 36 scenarios with less than 1.35 m/s (G2), and one with more than 1.7 m/s (G3).

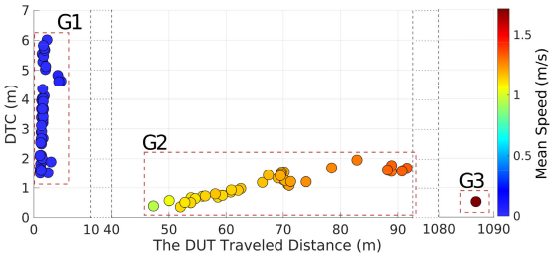

FIGURE 15. Result recorded from the high-fidelity simulation of selected scenarios. Each point on the chart represents the traveled distance and the minimum distance to collision (DTC). The color bar also shows the mean speed of the DUT in the mission. G1, G2, and G3 represent each group's scenarios.

TABLE 6. Simulations result of the scenarios classified in three groups.

Gr.	# Scenes	$\min(\bar{s})$	$\max(\bar{s})$	mean(NDT-s)
1	50	$0.00(m/s)$	$0.05(m/s)$	3.33
2	36	0.94	1.34	8.56
3	1	1.71	1.71	13461.8

In group 1 scenarios, the DUT has traveled less than $5m$ in total. This means that the DUT control software could not find any safe trajectory for the vehicle to follow. The second group cases, on the other hand, resulted in a minimum DTC between 0.36 and 1.95 m and a distance traveled between 47 and 91 m. Since the distance is greater than Dx, it implies that the DUT has successfully passed the NPC. Finally, there is an unexpected traveled distance in the third group as a result of the loss of localization. From the group 2 scenarios, it is evident that situations, where lower traveled distances combined with lower mean speeds, had smaller DTCs. This is indicative of a riskier trajectory being generated for passing. In addition, we also discovered from the data that as we increased the distance of the initial scenario Dx, the distances traveled increased as well. Table 6 provides more details for each group. The last column contains the mean NDT value, which indicates the localization accuracy during the mission. As one can see, the score of group 3 is higher than the others, indicating a non-localized situation. Furthermore, the average velocity of the scenario, which was about 1.7 m/s, confirms that the DUT was not under control.

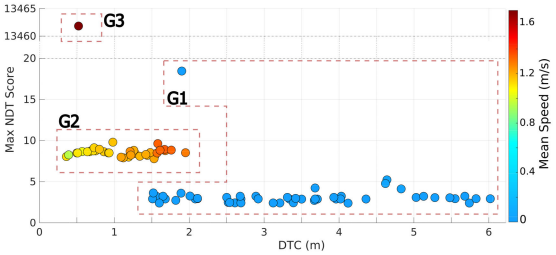

FIGURE 16. Scenarios represented by DTC and NDT score. The color bar also shows the mean speed of the DUT in the mission. G1, G2, and G3 represent each group's scenarios.

Figure 16 shows the maximum NDT scores, minimum DTC, and average speed of the DUT for each scenario. From the graph, it can be seen that group 1 has the lowest average NDT score due to less mobility. Nevertheless, there is one case with a high NDT score which may reflect poor initial localization. Group 2, on the other hand, has an average NDT score of 8.5, which is higher than that of group 1. This is because of the DUT turning motions during the passing maneuver, which reduce localization accuracy. A single scenario with a high significant NDT score indicates that localization of the DUT is lost in the maneuver. Figure 14 shows this scenario trace in group 3. It can be seen that the DUT almost passed the NPC and then lost localization and deviated from the path. Generally speaking, DUT motion makes the NDT matching algorithm for localization more challenging, as our NDT score rises as a result. Unexpectedly high NDT scores are indicative of a system failure and are reflected in the same level of crash severity.

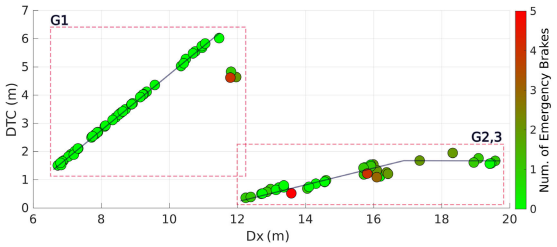

FIGURE 17. Scenarios represented by the initial longitudinal distance to the NPC and the minimum DTC reached during the simulation. Also, the color bar demonstrates the normalized brakes magnitude through each scenario.

To evaluate the safe performance of the DUT during operation, we plotted each scenario Dx against the minimum DTC to check how far the DUT can reach the NPC (see Fig. 17). Also, the color bar shows the total number of emergency braking during the mission, which explains the relative ride comfort and safety. According to the figure, the DUT did not move in the scenarios (G1) with an initial longitudinal distance of less than 12 m, although there were a few scenarios where negligible motion was recorded. The correlation and trend are represented by a straight line.

The next two groups are boxed and show that the DUT reaches the NPC closer than the originally specified distance, indicating that the DUT moved and attempted to pass the NPC. The G2,3 box contains all scenarios in which the DUT succeeded in passing the NPC, except for the one with the highest number of emergency brakes. Thus, the scenarios in which the shuttle was farther than 12 m from the NPC were successful.

Another interesting finding is the gradual increase of the DTC from 0.36 to 1.5 m, while we increased the initial distance from 12 to 16 m. Between 16 and 20 m, the minimum DTC did not change significantly and remained around 1.7 m. This means that the planning algorithms generate a path with a safer distance for the passing maneuver when the DUT starts to pass from a distance greater than 16 m instead of 12 to 16 m. In addition, to find edge case scenarios and evaluate the algorithms under critical conditions, we need to focus on the range where the DTC is about to collide (12 to 16 m). Moreover, DUT control software developers should consider making the DUT capable of passing objects that are less than 12 meters behind it.

TABLE 7. Summary over 87 scenarios simulated by the high-fidelity simulator.

	duration (sec)	D_y (m)	D_x (m)	$\max(s)$ $(\frac{m}{s})$	$\bar{s}$ $(\frac{m}{s})$	DTC (m)	NDT-s
mean	69.98	0.15	11.53	1.07	0.51	2.45	158.4
std	18.02	0.09	3.75	1.84	0.6	1.56	1434.4
min	46.89	0.01	6.72	0.01	0.00	0.36	2.39
max	108.51	0.37	19.57	15.80	1.71	6.02	13461.8

Table 7 reports some essential statistical features of the high-fidelity simulation results, including duration (sec), lateral and longitudinal initial distance to the NPC (m), max and average speed (m/s), minimum DTC (m), and the maximum NDT score. On average, it took almost twice as long to simulate the same scenario with the high-fidelity platform compared to the low-fidelity one (see Table 5). No scenario has been completed in less than 46 seconds in the high-fidelity setting, while the shortest simulation has been completed in less than three seconds in the low-fidelity simulation. In this example, we clearly see the importance of using low-fidelity simulations to avoid unnecessary simulation computation and thus generate high time savings.

Besides, the speed of the DUT in the high-fidelity tests was lower than the similar one in the low-fidelity simulation as the software controlling the DUT (Pure Pursuit Controller [57]) automatically adjusts the vehicle speed. For the same reason, none of the SVL simulations produced collisions as compared to the low-fidelity simulations. DTC values are smaller in high-fidelity cases, according to the data. It is because, in high-fidelity cases, DTC is measured from body to body, while in low-fidelity cases, it is measured from center to center.

V. EXPERIMENT

The purpose of this section is to present results from the practical application of the real DUT, iseAuto, in support of the simulation results. Figure 18 shows the setup and environment for conducting the experiment. The test was conducted on a straight two-lane road with passing capability in a private area designated for experiments. During the test, an intersection on the left side of the road was blocked to prevent any conflict. Based on Figure 17, an initial relative longitudinal distance of 18 meters was set in this setup to operate the shuttle in the safest possible range (max DTC). Furthermore, the vehicle was controlled using the same control algorithms used in the simulation to pass the NPC.

The experiment was recorded using a drone while recording all the sensors' data as a rosbag file. Figure 19 displays

FIGURE 18. Passing scenario setup. The entrance to the T intersection was blocked to avoid interruption.

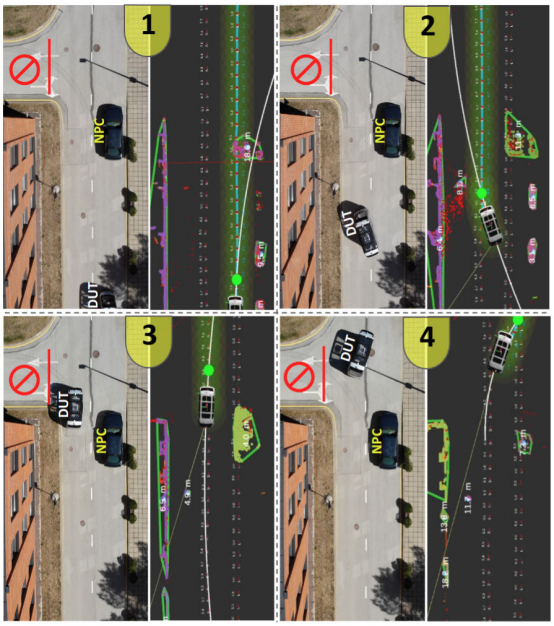

FIGURE 19. Four different time frames of the passing experiment are shown in the real test environment beside the RViz visualizer. The T intersection entrance was closed during the test.

four consecutive time frames captured by the drone (left images) and recorded from the RViz screen (right images) during the passing. In the RViz images, all detected objects defined by a green contour have a number that indicates the distance to the AV. In Fig 19, frame 1 shows the initial setup of the mission, where the AV (DUT) was following its straight route, and detected the NPC via the point cloud retrieved from sensors. At this point, the control algorithms drew a red line on the road to stop the AV and plan for passing.

In the next frame, the shuttle starts to follow the passing trajectory generated by its software while keeping a safe distance from the NPC. Then, frame 3 shows that the DUT almost passes the NPC while it was within its 2 *m* distance range as expected from simulations. Finally, in the fourth frame, the AV tried to change the lane and follow its original path.

The controller's steering commands were observed in both experiment and simulation (see Fig. 20). This was done to determine if we could estimate the control software behavior accurately. A number of factors are involved in getting a closer result to the real-world experiment, including vehicle dynamics and kinematics, sensor performance, and the quality of the virtual environment. It is evident from the figure that

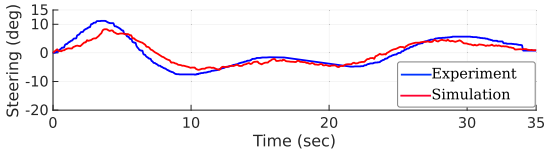

FIGURE 20. Values of Steering angle on the wheels recorded during the same scenario in the high-fidelity simulation and real experiment.

the high-fidelity simulation was able to predict the steering motion with a reasonable degree of accuracy. Although the high-fidelity simulation environment and the performance of the virtual sensors are not completely identical to the real-life ones, experimental results show that they can be considered valid for validation and evaluation. It is worth noting that the time and additional measures required to create such a simple scenario with a static actor are not comparable to simulation, which can be done easily and quickly, especially in complex and life-threatening situations.

VI. DISCUSSION

Nowadays, car manufacturers perceive safety and reliability as strongly related to the hardware components. For instance, the engine should not fail, the axles must be robust, the brakes must work, etc. However, in these items, the (human) driver seems to be seen as a passive component only relying on the hardware being working properly, and having no connection to the vehicle itself. However, most accidents are caused, to a certain extent, by human error. Manufacturers can provide safe hardware and safety devices (belts, airbags, etc.), but there is little control over the driver and its behaviour. In an autonomous driving paradigm, however, the driving agent is an active component and can, therefore, be controlled by developers and manufacturers to ensure passenger safety.

In this frame, autonomous driving is seen, already, as safer than non-autonomous driving, and the intent of this work is to equip researchers with a tool to improve safety and perform tests, verification, and validation for autonomous vehicles. Validation and verification are used to ensure that any AV meets the desired safety and performance criteria. This iterative process can lead to continuous improvement of AVs performance over time.

The presented approach provides a safe environment to test vehicle capabilities and identify potential flaws at zero risk. It allows researchers and developers to test AVs in a virtual environment, which reduces testing time and cost. Besides, repeatability and scalability enable AV experts to evaluate and optimize intended performance in a variety of scenarios. This two-layered validation approach integrates low-fidelity and high-fidelity simulations, commonly used in autonomous vehicle validation, to make the most of the advantages of each type of simulation. Users benefit from low-fidelity simulations since they are more accessible, faster to execute, and offer a broader range of scenarios to explore. As opposed to low-fidelity simulations, high-fidelity ones provide a highly realistic virtual environment that closely resembles the real world. It also provides more accurate results and can be

used to validate low-fidelity simulations or real-world tests. According to [58], this platform not only supports AV safety evaluation, but also enables experts to simulate advanced cyberattacks, such as sensor spoofing.

In low-fidelity simulations, the focus is typically on the planning part of the algorithm, excluding the other critical components of the autonomous feature, such as localization and perception that require sensor input. As a result, it is difficult to determine the reliability of the outcome derived from these types of simulations with a limited number of test cases. As mentioned previously, their rapid response and broader scenario coverage make them an appropriate tool for identifying more practical and critical scenarios for the next level of testing. By using this type of simulator, a comprehensive set of simulations can be conducted covering a wide range of ODDs and then the riskiest cases can be identified through search strategies (e.g. eagle strategy) for further analysis [5]. This initiative was taken, and the low-fidelity simulation was used, to nominate scenarios for the high-fidelity simulator to save time and explore vulnerabilities in AV control software efficiently.

This paper showcased an implementation of the proposed method on a passing maneuver. Findings confirm that the simulations based on low-fidelity were faster, but likely to have a lower reliability. This is due to the sacrifice of details in these simulations and the simplification of the system. It is acknowledged that, however, in these simulations, the AV was controlled by the rules defined for the scenario and not by AV software. While high-fidelity simulations are able to evaluate all autonomous features integrated into the AV software at once. The high-fidelity results corroborate that in a small batch of runs, developers can explore the algorithms' performance and behavior in the target scenarios without having to conduct experiments in real life. Obviously, this does not mean that the limited number of tests provides full safety assurance, but it can be used as a tool to identify more critical and corner cases.

In order to conduct a successful analysis, it is also imperative to define proper metrics to evaluate simulation results. Particularly in large numbers of runs, it is almost impossible to manually check the results, for this reason, metrics are expected to detect criticalities and errors during the simulation. Based on the analysis type and priority, several criticality metrics can be used, including time, distance, intensity, and velocity-based metrics described in detail in [59]. We have employed acceleration, velocity, distance, as well as intensity-based metrics in the current study. Even though no critical-safety cases were observed in the limited tests, we reported performance issues and corner cases that could pose a safety risk. It is notable that unexpected failures may occur during the testing process that has an adverse effect on the entire system, such as localization loss due to sharp maneuvers. These failures might not be observed while testing individual parts of the system in a low-fidelity setup. In this study, we carried out a real-life experiment to check the validity of the simulation results. Although the comparison test is limited and not enough to make a strong conclusion, the findings suggest a reasonable correlation. It should be admitted that implementing real-world experiments requires considerable effort and time due to the requirements and considerations involved.

We have discussed the advantages of simulation thus far, but they also have some limitations that may result in complications in the future. It is still necessary, however, to evaluate the reliability and naturalistic level of high-fidelity simulations. This can be accomplished by carrying out a high number of real-life experiments that are very labor-intensive, time-consuming, and in some cases potentially hazardous. Furthermore, high-fidelity simulations suffer from a number of limitations including costly hardware, time consumption, and synchronization. High-fidelity simulators, especially those based on game engines, require powerful CPUs and GPUs based on the simulator configuration. It is often the case that results are inaccurate and not well synchronized as a result of insufficient computational resources. Furthermore, due to the computational burden to simulate the sensors and the physics of the environment, the simulation time is different from the system time(real). Typically, this is the case particularly when there are multiple sensors on the AV (LiDARs and cameras). For instance, for simulating a scenario that lasts t seconds in the simulator, it takes $n \times t$ where $(n \in R^{+}, n > 1)$. It is expected that high-fidelity simulators will overcome these limitations in the near future with the advancement of game engines and GPUs.

In the future, research should be devoted to developing low-fidelity simulations that incorporate AV software to increase their reliability and accuracy for the first step of scenario evaluation. This can bring two benefits. First, it enables users to eliminate as many unnecessary scenarios as possible for time-consuming simulations. Secondly, it provides an agile platform for optimizing the motion algorithms parameters without taking into account other autonomous components. In addition, future research should investigate more challenging maneuvers with many actors involved and possibly using stochastic agents (featuring unpredictable behavior). It is then necessary to test a large number of scenarios in simulation and real-life environments to provide adequate evidence of the method's reliability.

VII. CONCLUSION

In recent years, autonomous driving technology has seen rapid development. However, to the best of the authors' knowledge, to date, there are still no agile, flexible, and comprehensive validation methods for such safety-critical systems. In our work, we presented an efficient and innovative technique for evaluating AV control software safety and performance on a target mission. This method combines a low-fidelity simulator with a highly detailed simulator to achieve fast and reliable validation results. This combination enables us to identify the corner case scenarios in an AV shuttle maneuver that may pose critical challenges to the control software. We found, in a small sample of runs, that we

could generate and nominate a limited number of scenarios for naturalistic simulations, which are generally more time-consuming. Further, high-fidelity simulation results suggest promising evidence for in-depth analysis of autonomous software that will shed new light on future developments.

To examine the simulation results, we implemented one of the proposed scenarios in a real experimental setup. Despite the fact that the real-life scanty results cannot be used to draw a strong conclusion, they do suggest that the proposed approach was successful in predicting vehicle performance and behavior. The results of this study will provide a basis for further research into the reliability of the AV simulation by conducting more empirical tests in the real world.

In the future, engineers and researchers can utilize this approach as a prerequisite for real experiments to increase evaluation efficiency and reduce safety-critical problems. The proposed approach could also be used to investigate and target various operational design domains and complex maneuvers in a large number of simulations in the future.

REFERENCES

[1] C. J. Haboucha, R. Ishaq, and Y. Shiftan, "User preferences regarding autonomous vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 78, pp. 37–49, May 2017.
[2] I. Panagiotopoulos and G. Dimitrakopoulos, "An empirical investigation on consumers' intentions towards autonomous driving," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 773–784, Oct. 2018.
[3] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, Apr. 2018.
[4] T. Zhang, D. Tao, X. Qu, X. Zhang, R. Lin, and W. Zhang, "The roles of initial trust and perceived risk in public's acceptance of automated vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 98, pp. 207–220, Jan. 2019.
[5] Q. Goss and M. İ. Akbaş, "Eagle strategy with local search for scenario based validation of autonomous vehicles," in *Proc. Int. Conf. Connected Vehicle Expo (ICCVE)*, Mar. 2022, pp. 1–6.
[6] D. Petrović, R. Mijailović, and D. Pešić, "Traffic accidents with autonomous vehicles: Type of collisions, manoeuvres and errors of conventional vehicles' drivers," *Transp. Res. Procedia*, vol. 45, pp. 161–168, Jan. 2020.
[7] R. L. McCarthy, "Autonomous vehicle accident data analysis: California OL 316 reports: 2015–2020," *ASCE-ASME J. Risk Uncertainty Eng. Syst., B, Mech. Eng.*, vol. 8, no. 3, Sep. 2022, Art. no. 034502.
[8] "Highway accident report, collision between vehicle controlled by developmental automated driving system and pedestrian," US Nat. Transp. Saf. Board (NTSB), Tech. Rep. PB2019-101402, 2018.
[9] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transp. Res. A, Policy Pract.*, vol. 94, pp. 182–193, Dec. 2016.
[10] A. Rassõlkin, L. Gevorkov, T. Vaimann, A. Kallaste, and R. Sell, "Calculation of the traction effort of ISEAUTO self-driving vehicle," in *Proc. 25th Int. Workshop Electr. Drives, Optim. Control Electric Drives (IWED)*, Jan. 2018, pp. 1–5.
[11] R. Wang, R. Sell, A. Rassõlkin, T. Otto, and E. Malayjerdi, "Intelligent functions development on autonomous electric vehicle platform," *J. Mach. Eng.*, vol. 20, no. 2, pp. 114–125, 2020.
[12] A. Rassõlkin, T. Vaimann, A. Kallaste, and R. Sell, "Propulsion motor drive topology selection for further development of ISEAUTO self-driving car," in *Proc. IEEE 59th Int. Sci. Conf. Power Electr. Eng. Riga Tech. Univ. (RTUCON)*, Nov. 2018, pp. 1–5.
[13] C. Medrano-Berumen, M. Malayjerdi, M. İ. Akbaş, R. Sell, and R. Razdan, "Development of a validation regime for an autonomous campus shuttle," in *Proc. SoutheastCon*, Mar. 2020, pp. 1–8.
[14] E. Malayjerdi, R. Sell, M. Malayjerdi, A. Udal, and M. Bellone, "Practical path planning techniques in overtaking for autonomous shuttles," *J. Field Robot.*, vol. 39, no. 4, pp. 410–425, Jun. 2022.
[15] E. Thorn, S. C. Kimmel, M. Chaka, and B. A. Hamilton, "A framework for automated driving system testable cases and scenarios," Dept. Transp., Nat. Highway Traffic Saf., Washington, DC, USA, Tech. Rep. DOT HS 812 623, 2018.
[16] J. A. Matute-Peaspan, A. Zubizarreta-Pico, and S. E. Diaz-Briceno, "A vehicle simulation model and automated driving features validation for low-speed high automation applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7772–7781, Dec. 2021.
[17] P. Brunner, F. Denk, W. Huber, and R. Kates, "Virtual safety performance assessment for automated driving in complex urban traffic scenarios," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 679–685.
[18] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," in *Proc. 4th Int. Conf. Connected Auton. Driving (MetroCAD)*, Apr. 2021, pp. 62–70.
[19] M. O'Kelly, A. Sinha, H. Namkoong, J. Duchi, and R. Tedrake, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9849–9860.
[20] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robot. Auton. Syst.*, vol. 94, pp. 43–52, Aug. 2017.
[21] M. Malayjerdi, V. Kuts, R. Sell, T. Otto, and B. C. Baykara, "Virtual simulations environment development for autonomous vehicles interaction," in *Proc. ASME Int. Mech. Eng. Congr. Expo.* New York, NY, USA: American Society of Mechanical Engineers, 2020, p. 5.
[22] M. Malayjerdi, B. C. Baykara, R. Sell, and E. Malayjerdi, "Autonomous vehicle safety evaluation through a high-fidelity simulation approach," *Proc. Estonian Acad. Sci.*, vol. 70, no. 4, pp. 413–421, 2021.
[23] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
[24] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.
[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
[26] (2021). *The 2021 IEEE Autonomous Driving AI Test Challenge*. [Online]. Available: http://av-test-challenge.org/
[27] V. Bolbot, G. Theotokatos, L. M. Bujorianu, E. Boulougouris, and D. Vassalos, "Vulnerabilities and safety assurance methods in cyber-physical systems: A comprehensive review," *Rel. Eng. Syst. Saf.*, vol. 182, pp. 179–193, Feb. 2019.
[28] R. Razdan, W. Mahoney, A. Kalia, J. Smith, T. Zarola, M. İ. Akbaş, J. Taiber, E. Straub, and R. Sell, "Unsettled topics concerning automated driving systems and the transportation ecosystem," Soc. Automot. Eng. (SAE), Warrendale, PA, USA, EDGE Res. Rep. EPR2020004, Mar. 2020.
[29] R. Razdan, M. İ. Akbaş, R. Sell, M. Bellone, M. Menase, and M. Malayjerdi, "PolyVerif: An open-source environment for autonomous vehicle validation and verification research acceleration," *IEEE Access*, vol. 11, pp. 28343–28354, 2023.
[30] C. Stark, C. Medrano-Berumen, and M. İ. Akbaş, "Generation of autonomous vehicle validation scenarios using crash data," in *Proc. SoutheastCon*, Mar. 2020, pp. 1–6.
[31] C. Medrano-Berumen and M. İ. Akbaş, "Abstract simulation scenario generation for autonomous vehicle verification," in *Proc. SoutheastCon*, Apr. 2019, pp. 1–6.
[32] H. Winner, K. Lemmer, T. Form, and J. Mazzega, "Pegasus—First steps for the safe introduction of automated driving," in *Road Vehicle Automation 5*, G. Meyer and S. Beiker, Eds. Cham, Switzerland: Springer, 2019, pp. 185–195.
[33] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 315–329, Mar. 2020.
[34] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, Apr. 2020.
[35] *Road Vehicles—Functional Safety ISO 26262*, Int. Org. Standardization (ISO), Geneva, Switzerland, 2018.
[36] *PAS 21448-Road Vehicles-Safety of the Intended Functionality*, Int. Org. Standardization (ISO), Geneva, Switzerland, 2019.
[37] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Commun.*, vol. 12, no. 1, p. 748, Feb. 2021.

[38] P. Koopman, U. Ferrell, F. Fratrik, and M. Wagner, "A safety standard approach for fully autonomous vehicles," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2019, pp. 326–332.
[39] P. Koopman and F. Fratrik, "How many operational design domains, objects, and events?" in *Proc. SafeAI@ AAAI*, 2019, pp. 1–4.
[40] P. Koopman and M. Wagner, "Toward a framework for highly automated vehicle safety validation," SAE Tech. Paper 2018-01-1071, 2018.
[41] G. Chance, A. Ghobrial, K. McAreavey, S. Lemaignan, T. Pipe, and K. Eder, "On determinism of game engines used for simulation-based autonomous vehicle verification," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20538–20552, Nov. 2022.
[42] S. Khastgir, S. Brewerton, J. Thomas, and P. Jennings, "Systems approach to creating test scenarios for automated driving systems," *Rel. Eng. Syst. Saf.*, vol. 215, Nov. 2021, Art. no. 107610.
[43] E. de Gelder, J. Hof, E. Cator, J.-P. Paardekooper, O. O. den Camp, J. Ploeg, and B. de Schutter, "Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18794–18807, Oct. 2022.
[44] S. Hallerbach, Y. Xia, U. Eberle, and F. Koester, "Simulation-based identification of critical scenarios for cooperative and automated vehicles," SAE Tech. Paper 01-1066, 2018.
[45] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, p. 648, Feb. 2019.
[46] M. Aydin and M. İ. Akbaş, "Identification of test scenarios for autonomous vehicles using fatal accident data," *SAE Int. J. Connected Automated Vehicles*, vol. 4, no. 1, pp. 119–132, Mar. 2021.
[47] J. M. Thompson, Q. Goss, and M. İ. Akbaş, "Boundary adherence and exploration in high dimensions for validation of black-box systems," in *Proc. IEEE 8th World Forum Internet Things (WF-IoT)*, Oct. 2022, pp. 1–6.
[48] Q. Goss, Y. AlRashidi, and M. İ. Akbaş, "Generation of modular and measurable validation scenarios for autonomous vehicles using accident data," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 251–257.
[49] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: A language for scenario specification and scene generation," in *Proc. 40th ACM SIGPLAN Conf. Program. Lang. Design Implement.*, Jun. 2019, pp. 63–78.
[50] Foretellix, "Measurable scenario description language reference," Tech. Rep. 0.9, Sep. 2019.
[51] Foretellix. (2020). *Technology—Foretellix*. [Online]. Available: https://www.foretellix.com/technology/
[52] G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne, and S. K. Gupta, "Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles," *J. Syst. Softw.*, vol. 137, pp. 197–215, Mar. 2018.
[53] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1061–1075, May 2014.
[54] M. Bersani, S. Mentasti, P. Dahal, S. Arrigoni, M. Vignati, F. Cheli, and M. Matteucci, "An integrated algorithm for ego-vehicle and obstacles state estimation for autonomous driving," *Robot. Auton. Syst.*, vol. 139, May 2021, Art. no. 103662.
[55] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 287–296.
[56] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. thesis, School Sci. Technol., Örebro Universitet, Örebro, Sweden, 2009.
[57] M. Samuel, M. Hussein, and M. Binti, "A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle," *Int. J. Comput. Appl.*, vol. 135, no. 1, pp. 35–38, Feb. 2016.
[58] M. Malayjerdi, A. Roberts, O. M. Maennel, and E. Malayjerdi, "Combined safety and cybersecurity testing methodology for autonomous driving algorithms," in *Proc. Comput. Sci. Cars Symp. (CSCS)*, Dec. 2022, pp. 1–10.
[59] C. Wang, C. Popp, and H. Winner, "Acceleration-based collision criticality metric for holistic online safety assessment in automated driving," *IEEE Access*, vol. 10, pp. 70662–70674, 2022.

MOHSEN MALAYJERDI received the B.S. degree in mechanical engineering from IAUM, and the master's degree in mechanical engineering from the Ferdowsi University of Mashhad, Iran, in 2014. He is currently pursuing the Ph.D. degree in mechanical engineering with the Tallinn University of Technology, Estonia. His research interests center around the intersection of autonomous vehicle developments and safety verifications. His current research is a simulation approach for safety verification and development of a real AV shuttle in a virtual environment.

QUENTIN A. GOSS (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in computer science from Florida Polytechnic University. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with Embry-Riddle Aeronautical University. His research interests are in simulation-based approaches to autonomous systems validation, solving complex problems with agent-based modeling and simulation, and software-based complex systems.

MUSTAFA İLHAN AKBAŞ (Member, IEEE) received the Ph.D. degree in computer engineering from the University of Central Florida. He is an Assistant Professor with the Electrical Engineering and Computer Science Department, Embry-Riddle Aeronautical University. His research interests in connected and autonomous cyber-physical systems, validation and verification, wireless networks, and mobile computing. His research resulted in theory, tools, and applications for projects spanning these areas. He has industry experience in projects with multinational defense industry partners and large enterprises. He is a member of ACM, AAAI, and SAE and serves on editorial boards and program committees of ACM and IEEE journals and conferences.

RAIVO SELL (Member, IEEE) received the Ph.D. degree in product development from the Tallinn University of Technology, in 2007. He is currently a Professor of robotics with TalTech. His research interests include mobile robotics and self-driving vehicles, smart city, and early design issues of mechatronic system design. He is with the Autonomous Vehicles Research Group, TalTech, as a Research Group Leader with a strong experience and research background in mobile robotics and self-driving vehicles. He has been a Visiting Researcher with ETH Zürich, Aalto University, and most recently at Florida Polytechnic University, USA, awarded as a Chart Engineer and an International Engineering Educator.

MAURO BELLONE (Member, IEEE) received the M.S. degree in automation engineering and the Ph.D. degree in mechanical and industrial engineering from the University of Salento, Lecce, Italy, in 2014. His research interests comprise mobile robotics, autonomous vehicles, energy, computer vision, and control systems. His research is focused on the area of advanced sensory perception for mobile robotics and artificial intelligence. From 2015 to 2020, he was with the Applied Artificial Intelligence Research Group, Chalmers University of Technology, where he actively contributed to several autonomous driving projects. In 2021, he was appointed as an Adjunct Professor with the Tallinn University of Technology supporting the research team in the area of smart transportation systems.