DOCTORAL THESIS

Scenario-based Validation of Safety and Performance of an Autonomous Vehicle by a Software in Loop Simulation Method

Mohsen Malayjerdi

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2023

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
43/2023

Scenario-based Validation of Safety and Performance of an Autonomous Vehicle by a Software in Loop Simulation Method

MOHSEN MALAYJERDI

TAL TECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Mechanical and Industrial Engineering
This dissertation was accepted for the defense of the degree on 04/09/2023

Supervisor: Professor Raivo Sell,
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia

Opponents: Professor Dietmar Pfahl,
Autonomous Driving Lab
Institute of Computer Science
University of Tartu
Tartu, Estonia

Dr. Libor Preucil,
Head of Intelligent and Mobile Robotics & Center for Advanced Field Robotics
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague
Prague, Czechia

Defense of the thesis: 09 October 2023, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Mohsen Malayjerdi

signature

ISSN 2585-6898 (publication)
ISBN 978-9916-80-039-3 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-040-9 (PDF)
Printed by Koopia Niini & Rauam

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
43/2023

Autonoomse sõiduki ohutuse ja jõudluse stsenaariumipõhine valideerimine tsüklisimulatsiooni meetodi abil

MOHSEN MALAYJERDI

TALTECH KIRJASTUS

"To my beloved family"

Contents

List of Publications

The following papers provide the basis for this thesis. Inside the thesis, they are referred to as Roman numbers.

Paper I Christopher Medrano-Berumen, **Mohsen Malayjerdi**, Mustafa İlhan Akbaş, Raivo Sell, and Rahul Razdan. "Development of a validation regime for an autonomous campus shuttle." In 2020 SoutheastCon, pp. 1-8. IEEE, 2020.

Paper II **Mohsen Malayjerdi**, Vladimir Kuts, Raivo Sell, Tauno Otto, and Barış Cem Baykara. "Virtual simulations environment development for autonomous vehicles interaction." In ASME International Mechanical Engineering Congress and Exposition, vol. 84492. p. V02BT02A009. American Society of Mechanical Engineers, 2020.

Paper III **Mohsen Malayjerdi**, Bariş Cem Baykara, Raivo Sell, and Ehsan Malayjerdi. "Autonomous vehicle safety evaluation through a high-fidelity simulation approach." Proceedings of the Estonian Academy of Sciences 70, no. 4, pp. 413-421, 2021.

Paper IV **Mohsen Malayjerdi**, Andrew Roberts, Olaf manual Maennel, and Ehsan Malayjerdi. "Combined Safety and Cybersecurity Testing Methodology for Autonomous Driving Algorithms." In Proceedings of the 6th ACM Computer Science in Cars Symposium, pp. 1-10. 2022.

Paper V **Mohsen Malayjerdi**, Quentin A. Goss, M. Ilhan Akbas, Raivo Sell, and Mauro Bellone. "A Two-Layered Approach for the Validation of an Operational Autonomous Shuttle", in IEEE Access, vol. 11, 2023.

Author's Contributions to the Publications

Paper I The author designed and tested a low-fidelity virtual testing environment inside the MATLAB Automated Driving toolbox for the TalTech iseAuto shuttle. That environment was used to investigate corner case scenarios in the TalTech campus track designated for AV operations.

Paper II The author proposed the Idea of building and customizing the digital twin of the working environment for the AV shuttle based on the research that had been conducted. Furthermore, the author integrated the new virtual environment inside the evaluation toolkit proposed in this thesis.

Paper III The author proposed the idea and developed a high-fidelity simulation methodology to evaluate the safety and performance of the AD software of the TalTech iseAuto. Based on the evaluation, the author suggested some improvements in the detection and planning parameters which led to further research.

Paper IV The author contributed to developing the idea and built the platform to analyze advanced cyberattacks in a high-fidelity simulated environment. Also, the author contributed to conducting real-world experiments to validate the simulation results.

Paper V The author contributed mainly to developing the idea and performing high-fidelity tests based on an end-to-end simulator. Moreover, the author conducted experimental tests and compared them to the high and low-fidelity simulation outputs.

Introduction

Driving has been an exclusively human function for more than a century. During the last few years, the automobile and technology industries have made substantial strides in bringing digital technology into what was previously an exclusively human activity. Advanced Driver Assistance Systems (ADAS) technologies have reached the inflection point and are ready to take control of almost all aspects of the driving task. A number of lives have already been saved and injuries prevented by this system, improving driver and passenger safety and convenience. Several companies have taken one step beyond by developing Autonomous Vehicles (AVs), known as self-driving vehicles, that can drive themselves on conventional roads and maneuver in a variety of roadways and environments without a human operator. These vehicles utilize a variety of sensors, such as cameras, radars, and LiDARs, to perceive their surroundings and make navigation decisions [1].

In light of the fact that these technologies can be found in the mass market and have become successful, AVs are capable of revolutionizing the transportation system [2]. One of the key drivers of this research is the potential benefits AVs could offer. These benefits include improving safety, preventing fatal crashes, reducing traffic congestion, sustainable transportation, and facilitating mobility for people who cannot drive. Numerous studies, conducted in different parts of the world, have shown that driver-related factors (i.e., error, impairment, fatigue, and distraction) are responsible for almost 90% of reported crashes over the past two decades [3, 4, 5, 6]. In contrast, vehicle component failure or degradation accounted for less than 10% of the crashes [7]. AVs have the potential to reduce and even eliminate human error as a cause of accidents, thus substantially reducing the hazards associated with motor vehicles.

With the rapid advancement of AVs, ensuring their safety and performance in real-world environments has become a critical challenge [8]. Simulation testing has emerged as a highly effective tool for evaluating AVs, allowing developers to assess their behavior and capabilities in a controlled and repeatable virtual environment. In this study, a testing framework is developed to evaluate the safety and performance of an autonomous shuttle through simulation testing. The framework provides users with a comprehensive toolkit that leverages both low- and high-fidelity simulations, enabling effective assessment of autonomous software in the loop of simulations. The developed toolkit offers several key features and capabilities. Combining low and high-fidelity simulations allows users to simulate a wide range of scenarios and evaluate AV behavior under diverse conditions. The toolkit provides a customizable set of tools that facilitate the assessment of AV software, including perception, decision-making, and control algorithms. Users can recreate complex traffic scenarios, pedestrian interactions, and various environmental conditions to thoroughly evaluate AV's responses. The benefits of the proposed framework are numerous. Firstly, it offers cost and time efficiency by reducing expensive, time-consuming, and labor-intensive real-world testing. Simulation testing allows for rapid iterations and extensive testing coverage, accelerating validation. Additionally, the framework ensures scenario reproducibility, allowing for consistent assessment of critical scenarios, edge cases, and rare events that are challenging to encounter consistently in the real world. Moreover, the framework prioritizes safety assurance, as simulation testing provides a safe environment to identify and address potential system failures, software bugs, or hardware malfunctions before conducting real-world tests or deploying the vehicle on public roads.

The framework's importance lies in its contribution to AV validation. Providing a reliable and efficient tool for assessing AV safety and performance helps build public trust in this emerging technology. Furthermore, it supports compliance with regulations by ensuring that the AV meets the required standards. The framework also facilitates iterative development, allowing developers to rapidly implement changes and improvements to vehicle algorithms, sensors, and behavior. Overall, the developed framework represents a significant step towards widespread AV adoption and deployment by providing an effective and comprehensive simulation-based validation approach.

Abbreviations

AD	Autonomous Driving
ADAS	Advanced Driver Assistance System
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AS	Autonomous System
ASIL	Automotive Safety Integrity Level
AV	Autonomous Vehicle
DT	Digital Twin
DUT	Device Under Test
FP	False Point
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
Hil	Hardware in Loop
HF	High-Fidelity
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
LF	Low-Fidelity
NPC	Non-Player Character
ODD	Operational Design Domain
PID	Proportional Integral Derivative
ROS	Robot Operating System
RViz	Ros Visualization
SAE	Society of Automotive Engineers
SiL	Simulation in Loop
SOTIF	Safety Of The Intended Functionality
V&V	Verification and Validation

1 Literature Review

Over the past decade, significant progress has been made in developing AV technologies [9, 10, 11], and there are now a number of companies and organizations working on bringing AVs to market [12, 13, 14]. For example, Waymo, a subsidiary of Alphabet (Google), has been developing AV technology since 2009 and has conducted extensive testing of its self-driving cars on public roads [15]. Also, several start-up companies have been formed with the aim of developing automated vehicles, including Zoox, EasyMile [16, 17], Navya [18], etc. Furthermore, this study focuses on an autonomous shuttle operating on the campus of Tallinn University of Technology as a public last-mile vehicle. By developing AVs, we are contributing to the significant technological challenge of mastering an unprecedented level of complexity not yet seen in the automotive industry [19]. Thus, there has been extensive research into the V&V of AVs, among others, in several national and international research projects and several approaches to validation have been developed [20].

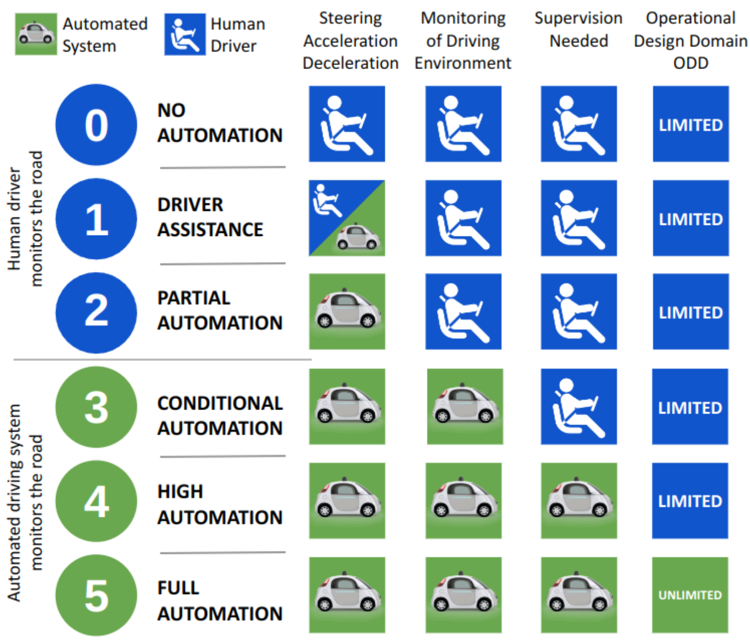

Figure 1.1: SAE J3016 levels of driving automation.

1.1 SAE's Levels of Autonomy

According to the Society of Automotive Engineers (SAE) [21], there are six levels of autonomy for vehicles, ranging from no automation (driver in full control) to fully AVs that require no human input at all (see Fig. 1.1). Intermediate levels are differentiated by the number of automated systems and the necessity of the driver to be available at all times. These levels are often referred to as SAE's Levels of Autonomy or SAE J3016. By using this taxonomy, we can easily distinguish AVs depending on who is charged with monitoring the driving environment. Currently, many car manufacturers mass-produce vehicles featuring level 2 autonomy including Tesla, Volvo, and Volkswagen. As we move forward with the level of autonomy, there are still many challenges and concerns, especially with regard to safety and performance. In order to address this issue, this study is dedicated to the development of testing tools for cars that feature levels 3 to 5 of autonomy.

1.2 AVs Verification and Validation

AV verification and validation (V&V) is the process of testing and verifying the safety, reliability, and performance of AVs before they are deployed on public roads [22]. It is an essential step in ensuring the safety of AVs and building public trust in this technology. As follows, each term describes a specific aspect of the testing process:

Verification: The process of providing objective evidence that a system, software, or hardware meets the requirements. "Did we build the right system?"

Validation: An evaluation process designed to demonstrate that a system, software, or hardware meets its intended use and meets the needs of its users. "Did we build the system right?"

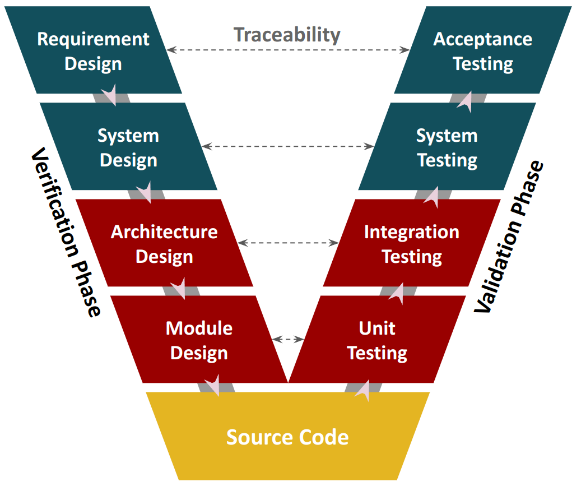

Figure 1.2: V-model testing approach in ISO 26262

Various approaches and techniques exist for the V&V of conventional software systems; however, autonomous systems differ from ordinary systems in that they learn, adapt, and change as they encounter new challenges [23, 24]. The V&V process for AVs involves a range of activities, including simulation, testing, and validation. These activities are designed to ensure that the AV operates as intended, is safe for passengers and other road users, and meets all regulatory requirements. Based on known system development methodologies, including the V-model [25, 26], they identify the principal activities required to ensure an acceptable safety level during the system development process. Figure 1.2 illustrates a simplified version of the V-model development methodology adopted by the Functional Safety Standard for Road Vehicles ISO 26262 [27]. The verification steps are located on the left side of the process, while the validation steps are located on the right side. The following is a brief overview of the different stages involved in AV V&V:

Testing: Unit or system testing is one of the effective measures to assess the performance or safety of the Device Under Test (DUT). The following are the primary methods of testing used in the automotive industry.

- **Simulation:** Simulation testing involves creating virtual environments that simulate real-world driving scenarios. AV software and sensors are tested in these environments to verify that they perform as intended. Simulations are conducted in either a lower detail environment known as low-fidelity or a higher detail environment known as high-fidelity. Simulation testing allows manufacturers to test their vehicles in a safe and controlled environment and to collect data for further analysis.
- **Track testing:** In track testing, AVs are tested in controlled environments, such as test tracks or closed-off areas of public roads. This type of testing allows manufacturers to evaluate the performance of their vehicles under a variety of conditions and to collect data on how the vehicle behaves in different situations.
- **On-road testing:** On-road testing involves testing AVs on public roads, where they will eventually be deployed. This type of testing allows manufacturers to evaluate the performance of their vehicles in real-world situations and to collect data on how the vehicle interacts with other road users.

Validation: Validation involves analyzing the data collected during simulation and testing to determine if the AV meets all safety and performance requirements. This step is critical in ensuring that the vehicle is safe for passengers and other road users.

Certification: Once a manufacturer has completed the V&V process, the vehicle may be certified for use on public roads. Certification requirements vary by region and may involve demonstrating compliance with local regulations and safety standards.

Overall, the AV V&V process is a complex and iterative process that involves multiple stages of testing and analysis. Through this process, manufacturers can ensure that their vehicles are safe and reliable, and build public trust in this emerging technology.

1.3 Certification

There are several safety standards and certifications that AVs must meet to ensure their safety and compliance with regulations. Here are some of the most important ones:

ISO 26262: Road Vehicles—Functional Safety, The standard outlines functional safety requirements for road vehicles, including AVs [28]. Over the years, it has been upgraded to meet the needs of self-driving vehicles [29]. It covers the entire lifecycle of the vehicle, from concept through to decommissioning. As part of ISO 26262, the Automotive Safety Integrity Level (ASIL), as a key component, is determined at the beginning of the development process.

ISO 21448: Road Vehicles-Safety of the Intended Functionality (SOTIF), This standard aims to address faults that can occur in an AV because of issues occurring in the perception, classification, and path planning subsystems, which are not covered by the ISO 26262 standard addressing the actuation subsystem. Originally published in January 2019, this document provides guidance for SAE J3016 Levels 0-2; the guidance is applicable, but likely insufficient for higher levels of automation [28]. SOTIF provides guidance on systemic failure analysis in AVs. Contrary to ISO 26262, which focuses on malfunctions, SOTIF employs a complexity approach that accounts for a wide range of possible hazards. Moreover, SOTIF recommends developing new strategies for V&V involving statistical analysis.

ISO/SAE 21434: Road Vehicles - Cyber security Engineering, This standard, as a successor of SAE J3061 [30], is being developed by a joint working group of ISO and SAE in 2020 to establish a set of high-level principles for cyber security. The purpose of the first standard was to (a) provide a structure for ensuring cyber secure design, (b) thus reduce the potential for a successful attack and reduce the likelihood of losses, and (c) to ensure that cyber-security threats are dealt with consistently across global industries by providing clear guidelines [31].

SAE J3016: This is a set of guidelines developed by the Society of Automotive Engineers (SAE) that defines the six levels of automation for vehicles, from Level 0 (no automation) to Level 5 (full automation) [21].

FMVSS: The Federal Motor Vehicle Safety Standards (FMVSS) are a set of safety standards established by the US National Highway Traffic Safety Administration (NHTSA) that all motor vehicles must meet. These standards cover a range of safety features, including crashworthiness, occupant protection, and safety systems [32].

UL 4600: This is a set of safety standards developed by Underwriters Laboratories (UL) specifically for AVs. It covers a range of topics, including cyber security, validation and verification, and safety case development [33].

NHTSA's Automated Vehicle Safety Checklist: This is a voluntary guidance document developed by the NHTSA that outlines a series of safety considerations for AVs, including data recording and sharing, cyber security, and crashworthiness [34]. In February 2020, NHTSA released AV 4.0—Ensuring American Leadership in Automated Vehicle Technologies: Automated Vehicles 4.0. There have been a large number of voluntary safety self-assessments developed by AV developers that provide them with the opportunity to demonstrate their technology while demonstrating public safety [35].

It's important to note that while these standards and certifications are essential, they do not guarantee complete safety for AVs. A standard procedure or consensus has not yet been reached regarding how AVs should be tested and evaluated, to the best of the author's knowledge. Although various AV developers, government agencies, professional organizations, and academic institutions have analyzed the problem of AV testing extensively over the past few years, the theory and methods to support such testing and evaluation still remain undeveloped [36]. AV technology is still evolving, and new safety challenges may emerge over time. It will be critical to continue monitoring and updating safety standards and certifications to ensure the ongoing safety of AVs.

1.4 AV Testing Methods

Tests and evaluations are critical to AV development and deployment. There have been established procedures for testing human-driven vehicles for many years, such as the FMVSS [32]. Currently, however, automobile safety standards do not yet fully consider the driver's performance when performing driving tasks. It is essential to evaluate the intelligence of an AV [37], similar to the driver's license test, to determine whether an AV can function safely and efficiently without human involvement. Figure 1.3 displays three primary platforms used in AV testing and evaluation. there are pros and cons to each of the three platforms described as follows:

Figure 1.3: Three main AV testing platforms: a) On-road testing b) Track testing c) Simulation

a) On-road Test: The on-road test is the most realistic, however, it is extremely inefficient and risky. To demonstrate the safety of an AV at the level of a human-driven vehicle, an AV must drive hundreds of millions of miles [38]. However, test miles are not, by themselves, a good measure of AV safety. This is because most on-road scenarios do not present significant challenges for AV evaluation. As an example, if we want to determine how safe an AV is by observing its reaction to red light-running vehicles at intersections with traffic lights, it may be necessary for the AV to pass thousands of intersections before it reaches sufficient accident events, which is extremely difficult. In addition, on-road testing is a labor-intensive and hazardous process.

b) Track Test: Track testing, on the other hand, has a number of advantages. In comparison to the on-road test platform, it is a more controlled and therefore safer environment for the testing of AVs. As well as this, track testing has the potential to greatly improve the efficiency of the testing process, i.e., obtaining the evaluation results with the same accuracy with fewer tests. It should be noted, however, that both methods mentioned above are not time and cost-efficient.

c) Simulation: Simulation testing is an alternative to the previous two, as it is cost and time-effective, scalable, and repeatable [39]. However, despite the enormous development of the past few years, challenges remain, including the accuracy of vehicle models and dynamics, advanced virtual sensors, and virtual environments.

Even though simulation tests are advantageous, on-road and track testing remains indispensable before deployment. It is possible, however, to reduce the effort for on-road testing through the proper design of simulation scenarios and the track testing facility. It is therefore essential to generate a testing scenario library for each operational design domain (ODD) in order to maximize the benefits of both simulation and track testing. ODD refers to the operating conditions under which an ADAS is specifically designed to function [4]. Depending on the parameters of an ODD, there can be millions of different scenarios (e.g., different behaviors of Non-Player Characters (NPCs)). Typically, a testing library consists of a subset of scenarios that can be utilized to assess certain pre-defined performance criteria (e.g., safety). Since, the library contains more safety-critical scenarios, testing in a virtual environment is usually more efficient and safe than testing in a physical environment [40].

1.5 Scenario-based Validation

In this section, an overview of the current state-of-the-art techniques related to scenarios-based assessments of AV safety [41, 42, 43, 44, 45, 46] is presented. Scenario-based testing involves testing the DUT in selected target scenarios. It is characterized by the partitioning of the driving space into individual traffic situations and testing them through the use of virtual simulation. A scenario may include, for example, overtaking an NPC car, or a red-light runner at an intersection who crosses the DUT's path straight ahead. Currently, the German ordinance of regulations for automated driving requests that applications for AV operating licenses include a catalog of test scenarios [41]. This shows that scenario-based testing is relevant even for the certification of AVs. It is also important to note that, due to the high level of competition in this field across car companies and tech giants, it is not possible to perform a comprehensive review of the state of the art.

1.5.1 Coverage-based methods

Methods that use coverage-based approaches work towards covering as much of the driving space as possible. Assuming all scenarios have the same probability of occurrence, they generate new scenario samples either within input parameter ranges or from parameter distributions that include the probability of occurrence of scenarios during testing.

Parameters Ranges:

Parameter ranges are standard techniques that consider all possible combinations of scenario parameters. In the use case inputs, all continuous parameters such as position and velocity of traffic vehicles are transformed into discrete parameters following a step size through coarse discretization, as demonstrated in the studies [47] and [48]. As a result of turning the input parameters of the continuous space into a finite set of scenarios, any scenario that does not fall within this discrete set will not be addressed. A small step size will result in a large number of scenarios being simulated, requiring greater computational resources. Consequently, the approach is to minimize the number of simulations by increasing the step size as much as possible. The technique is capable of determining different failed scenarios across a broad range of inputs in a short period of time. Nevertheless, if the step size is too large, many critical scenarios could be overlooked and not detected. One of the key disadvantages of these methods is the need to tailor the step size for each use case. This resulted in other methods being developed to address this issue. Many studies have utilized regression testing [49], Signal Temporal Logic (STL) [50], randomization techniques [51], Scenario Importance strategy [52], and Design of Experiments (DoE) [53] to generate scenarios from a set of parameter ranges.

Parameters Distributions:

This method's focus is on sampling with parameter ranges that are used in the simulation software to determine the driving situation. In [54, 55, 56], alternative methods that employ parameter distributions are described. Methods based on accelerated Monte Carlo are mainly used in these approaches. A Monte Carlo technique can be used to generate new samples by estimating the probability of an event occurring, which is the failure probability in our example. However, it can take a long time to execute if implemented in a basic random manner, making it inefficient.

1.5.2 Falsification-based methods

These falsification-based approaches are designed to identify only edge case scenarios, which are the examples that cause the AVs to fail. It is possible to perform simulation-based falsification in combination with an optimizer in a feedback loop or to consider methods that do not involve simulation, such as accident databases or selection methods based on criticality and complexity.

Non-Simulation Based:

- **Accident-database:** In general, they are used to test ADAS. To enhance AV safety, scenarios in which human driver behavior results in a safety violation are collected. The German In-Depth Accident Study (GIDAS), for example, assesses the requirements of ADAS in urban environments [57, 58] or in limited highways [59] to reduce the potential for accidents. However, accident data cannot be extensively used to validate AVs at levels 3 and higher. By mitigating accidents that have already occurred, the system can rely on them to be updated, but it is important to use different methods for preventing accidents that are not yet occurring [60].
- **Criticality-based:** This method is based on an optimization that tries to increase the criticality of safe scenarios. The works in [61], [62], and [63] followed this methodology based on an evolutionary optimization to maximize scenario criticality. However, they did not examine the likelihood of an AV failure in the critical scenarios proposed.
- **Complexity-based:** This approach entails considering the parameter ranges of a driving situation and detecting critical scenarios by increasing the complexity of it. As an example, in the Analytic Hierarchy Process, weights are assigned to each scenario parameter, which then defines a complexity index that is combined with combinatorial testing. Studies [64, 65, 66] show a correlation between scenario complexity and the number of system failures.

Simulation Based:

Compared to previous approaches, simulation-based approaches require access to a simulator in order to suggest new edge-case scenarios. In that case, they need an optimizer to generate the next scenario, and then forward it to the simulator for running. The optimizer monitors each simulation result in order to create the next simulation plan. The optimizer detects more critical scenarios throughout iterations depending on the cost function considered. A number of approaches have been employed, which differ mostly in their optimizers and methodologies, including Reinforcement Learning (RL) [67], stochastic optimization [68], Bayesian optimization [69], and Simulated Annealing (SA) [70].

1.6 Simulation-based Validation

Studies that have been conducted on simulation-based validation for AD systems [71, 72, 73] have produced numerous safety-critical scenarios, either through the use of scenario modeling languages [74, 75] or from publicly available databases [76].

There is a well-known probabilistic programming language known as SCENIC [74], which allows for the generation of scenarios for autonomous driving systems. The Paracosm software system [75] also allows users to describe complex driving situations and generate scenarios using a variety of parameter configurations. In many studies, search-based algorithms are utilized in order to generate scenarios that challenge AVs. As an example, AV-fuzzer [77] uses a Genetic Algorithm (GA) based search to detect situations in which an autonomous driving system may violate safety requirements. Based on vehicle dynamics, the search perturbs the driving maneuvers of traffic users and enhances the safety requirements. In [76], authors proposed MOSAT (Multi-Objective, Search-based Approach to Testing), a Multi-Objective Search-based GA that exposes ADS safety violations by creating a diverse and adversarial driving environment.

Using broadcast modeling and Signal Temporal Logic (STL), a method is proposed [78] for controlling the generation of traffic. This method is capable of generating realistic traffic trajectories that mimic real traffic and meet the STL formulas' desired objectives. Nevertheless, it does not address V&V issues. However, the effectiveness of the generated traffic trajectories in identifying potential safety concerns is not yet clear. Attempts are also made to generate scenarios based on the topology of maps. This work in [79] extracts a wide variety of road networks from OpenStreetMap in order to facilitate the virtual testing of motion planners for automated vehicles. Using the traffic simulator SUMO [80], it generates traffic scenarios for these road networks. Nonlinear optimization is used to increase the criticality of the scenarios. In [81], junction lanes are classified based on collision avoidance maneuvers of the DUT, and GA is applied to create scenarios based on map topology. Using zone graphs for behavior analysis, SOCA [82] abstracts traffic situations at junctions into individual zones, where each zone graph represents the intentions of an individual vehicle. Additionally, many works describe traffic rules and the safety properties of autonomous driving systems using linear temporal logic [83, 84, 85], or STL [78].

AV validation has also been conducted using various testing techniques, such as combinatorial testing [86, 75], metamorphic testing [87, 88], and fuzz testing [77]. Combinatorial testing is one of these techniques that can guarantee the coverage of the scenarios based on the parameters given [86, 75]. The inconsistency between the outputs can be detected with the assistance of metamorphic relationships on the inputs, such as generating scenes with differing weather conditions [87] or providing noise [88], which removes the need to use a test oracle. By mutating existing scenarios, fuzz testing can identify potential bugs or safety violations [77].

A general approach to scenario-based AV testing for safety validation is presented in [43]. Using this approach, scenario-based testing is divided into two phases: scenario extraction and testing. Scenarios can be categorized into three abstraction levels: functional, logical, and concrete [89]. The functional scenario is merely a vague description, whereas the logical scenario includes some parameters with ranges, and the concrete scenario has a specific value for the parameters.

1.7 AV Virtual Simulators

Many simulators are available for testing AV, each with its own advantages and disadvantages. Most of them are proprietary tools, including the Waymo simulator platform [90] and Nvidia Drive sim [91], however, there are many open-source simulators available as well. Several open-source simulators are presented in the following. In terms of fidelity, AV simulations range from low- to high-fidelity, depending on the level of detail they represent. Low-fidelity simulators mimic the actual scenario, but do not include detailed factors, and are therefore useful for unit-level testing. In general, they are used to assess the criticality of scenarios and algorithms for AV motion planning. Unlike low-fidelity simulations, high-fidelity simulations are based on realistic characterizations of a validation scenario and include a large number of features suitable for system-level testing. Figure 1.4 shows two different fidelity simulator platforms. In this study, both methods are utilized in a comprehensive validation toolchain.

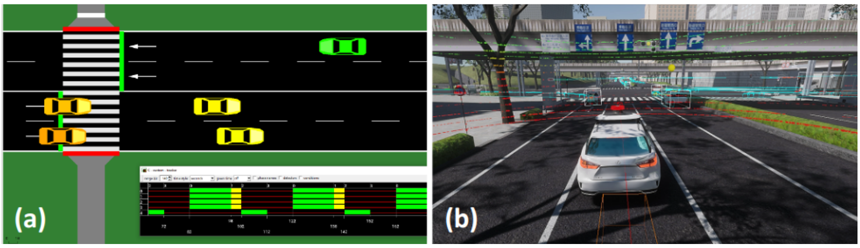

Figure 1.4: Fidelity of AV simulation: a) Low-Fidelity SUMO simulator [80] b) High-Fidelity AWSIM simulator [92]

1.7.1 Low-Fidelity Simulators

The following is a list of some of the most popular low-fidelity simulators. Nevertheless, many low-fidelity platforms have been deployed in various works [93, 94, 95], and mentioning them is beyond the scope of this study.

MATLAB/Simulink: MATLAB provides an Automated Driving Tool-box™ [96], a set of tools that facilitate the design, simulation, and testing of advanced driver assistance systems (ADAS) and automated driving systems. Users are able to test core functions such as perception, path planning, and vehicle control through this application. Additionally, it offers the RoadRunner interactive editor [97], which allows the creation of 3D scenes for testing and simulating automated driving systems. You can create road signs and markings specific to a particular region to customize roadway scenes. The exported scenes can be used in various automated driving simulators and game engines.

SUMO: Simulation of Urban Mobility (SUMO), is an open-source, highly portable, microscopic, and continuous multi-modal traffic simulation package designed to handle large networks [80]. In addition to modeling road traffic, SUMO allows pedestrians and public transportation to be included in the model. SUMO includes a wide range of support tools that allow users to perform tasks such as route finding, visualization, network import, and emission calculations. With SUMO, custom models can be created and the simulation can be controlled remotely via various APIs [98].

CommonRoad: It provides a benchmark collection for motion planning algorithms on roads. This platform specifies in depth the motion planning problem in terms of initial state, goal region, road network, static and dynamic obstacles, and DUT model [99]. The company provides a scenario database containing a collection of naturalistic datasets, handcrafted scenarios, as well as automatically generated scenarios. These databases are provided in XML format [100].

Autoware Planning Simulator: As part of the Autoware software stack [101], this simulator allows users to simulate their motion planning algorithm in a simplified virtual environment based on a kinematic-based approach [102]. By using fake perception data, the simulator bypasses the sensing component of the software and focuses exclusively on the planning component [103].

1.7.2 High-Fidelity Simulators

This type of simulation attempts to simulate the scenario as closely as possible to reality. However, this requires highly detailed 3D virtual environments and very accurate calculations at the lower levels of the vehicle, such as the vehicle dynamics. Listed below are some of the top-notch game-engine-based simulators.

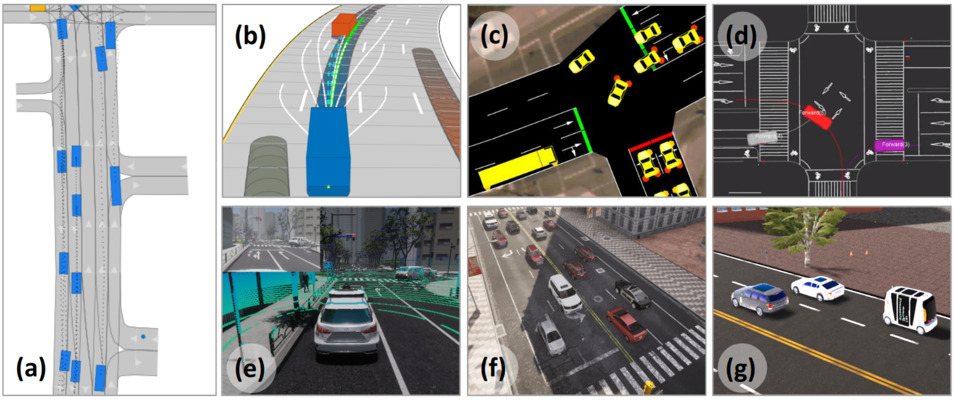

Figure 1.5: Popular AV simulators with different fidelity levels. Low-Fidelity: a) CommonRoad b) MATLAB c) SUMO d) Autoware, High-Fidelity: e) AWSIM f) CARLA g) SVL

SVL: This is a multi-agent AV simulator developed by LG Electronics America R&D Center [104]. They provide a highly detailed virtual platform as a solution for V&V of AV algorithms. It is integrated into some of the autonomous software stacks such as Autoware [101] and Baidu Apollo [105], which make it easy to test and validate the entire system. It is an open-source simulator that is developed using the Unity game engine [106]. Several bridges are available to facilitate message passing between the AV stack and the simulator backbone in SVL. This simulator fully supports Robot Operating System (ROS) bridges to transfer sensor data. The simulator platform provides several functions, including traffic, physical environment, sensor, and vehicle dynamics simulation. It also provides a Python API to control different environment entities and generate various scenarios. This platform supports cameras, LiDAR, IMU, GPS, and radar sensors. The project, however, has been sunset since 2022, and it has not been updated since then.

AWSIM: A newfound simulator platform [92] based on the Unity game engine that offers simulation services for "Autoware.universe" software stack. It is compatible with the ROS2 platform. However, it is significantly improved in terms of the virtual sensors engine as compared to the SVL. AWSIM uses Robotec GPU Lidar, an RTX-accelerated, CUDA/C++ library developed by Robotec.AI [107] to improve simulation performance. The simulator team is currently updating its technical documentation to include more tutorials and walkthroughs for easier customization.

CARLA: CARLA [108, 109] is an open-source well-known simulator that contributes to many research projects related to autonomous driving. It has been developed using the Unreal Engine [110]. With its modular and flexible design, this tool serves as a powerful tool that facilitates the training and validation of AV systems. Due to its high-fidelity characteristics, it is suitable for a variety of ADAS applications, including training algorithms for perception and planning. CARLA offers an API that can be customized by users and allows them to control the simulation. The API is based on Python and C++, and it is constantly evolving in parallel with the project. Among all simulators, CARLA has the largest community that constantly contributes to its development.

Figure 1.5 displays different scenes simulated in the aforementioned simulators. The majority of these simulators were used in the research included in this thesis. In Table 1.1, the simulator type used in the research papers is listed. Further, a safety and performance V&V toolchain is established based on these simulators.

Table 1.1: Simulators used in the included research papers.

Simulator	Paper I	Paper II	Paper III	Paper IV	Paper V
MATLAB	✓				
SUMO					✓
Autoware				✓	
SVL		✓	✓		✓
CARLA				✓	

✓Indicates the usage of the corresponding simulator

1.8 TalTech Autonomous Shuttle

Among all types of AVs, last-mile autonomous shuttles such as Zoox, EasyMile [16, 17], and Navya [18] have operated in many places. These shuttles operate in limited areas, such as airports or large residential areas. Tallinn University of Technology (TalTech) successfully demonstrated and implemented an autonomous shuttle, TalTech iseAuto™ [1], on its campus (see Fig. 1.6). This shuttle was built by the TalTech autonomous system research group in conjunction with AuVeTech company and ABB in Estonia. The objective of the TalTech iseAuto project was to develop an open-source AV shuttle and establish a smart city testbed on the university campus so that different types of urban mobility and autonomy-related research could be conducted there [111, 112, 113]. Since 2018, there have been a number of studies conducted on AVs with the help of this testbed, including [114], [115], and the current study. Additionally, one of the contributions of this study is the use of this operational autonomous shuttle as a testbed to assess the validity of the results obtained.

1.8.1 TalTech iseAuto Autonomous Software

TalTech iseAuto is controlled by an open-source software stack known as "Autoware.ai". This autonomous solution is ROS-based software that enables users to control mobile robots including self-driving vehicles. An overview of this platform and its components has already been provided in [101, 114]. The following is a brief description of the Autoware.ai system architecture. It is worth mentioning that Autoware is designed for the urban environment, but it can also be used on highways, although additional modules will be required. Figure 1.7 illustrates the core modules of Autoware.ai's architecture.

Perception: AVs must maintain a high level of safety. Therefore, the perception modules must be capable of calculating the position of the AV within a 3D map as well as identifying objects in the environment and traffic signals. LiDAR scanners and cameras are primarily used by Autoware in order to identify road environments. LiDAR

[1]TalTech iseAuto is a trademark registered by Tallinn University of Technology.

Figure 1.6: Autonomous shuttle, TalTech iseAuto

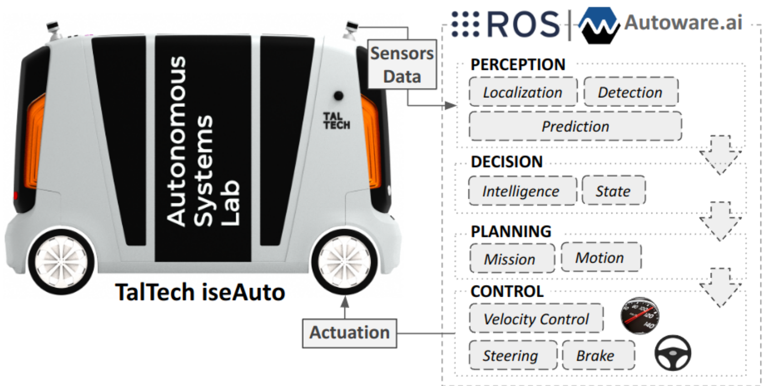

Figure 1.7: Overview of Autoware.ai architecture

scanners measure the distance to objects by illuminating pulsed lasers at a target and measuring the time at which the laser pulses are reflected. A digital 3D representation of scanned objects can be generated from the point cloud data produced by LiDAR scanners. The raw LiDAR point cloud data obtained from the scanners is filtered and pre-processed by Autoware in order to achieve real-time processing. Cameras are commonly used to recognize traffic signals and extract additional features of scanned objects. Localization, detection, and mapping can be refined using data from other sensors, including radars, GNSS (Global Navigation Satellite System), and IMUs (Inertial Measurement Units).

Decision: Once obstacles and traffic signals have been detected, trajectories of other moving objects can be calculated. These estimated results are used by the mission planning and decision-making modules to determine the direction the AV should move. Using an intelligent state machine, Autoware understands, forecasts, and responds to the road's status. Moreover, Autoware also allows AV users to supervise automation, overwriting the state determined by this module.

Planning: Trajectories are generated based on the results of the decision-making module.

Path planning can be classified into mission and motion planning. Using the current location and the given destination, Autoware determines a global trajectory. Alternatively, local trajectories are generated by the motion planning module along with global trajectories.

Control: AVs must follow local trajectories once determined. A control module, such as pure-pursuit or MPC, generates the actuation commands and adjusts the velocity for the AV.

1.8.2 TalTech iseAuto Planner Algorithm

One of the most commonly used path-planner modules in AV software is OpenPlanner which is integrated inside Autoware.ai. This module has become significantly more advanced in terms of supporting various high-definition map formats, predicting other actors' trajectories, and using a kinematics-based trajectory generator [102, 116].

OpenPlanner integrates global and local planners to generate local waypoints based on a global route and manage discrete behaviors, such as avoiding dynamic obstacles and following traffic signals. Local planner modules generate tracks parallel to the main global planner path. The tracks are referred to as rollouts (see figure 1.8). Local planner modules generate tracks parallel to the main global planner path. The tracks are referred to as rollouts. A trajectory evaluator considers all possible rollouts if an obstacle blocks the path. After that, the behavior selector will lead the AV to the new safe rollout. Figure 1.8 illustrates how open-planner selected rollout number 6 to pass the NPC. Furthermore, it detects curb lines and avoids rollouts that intersect them.

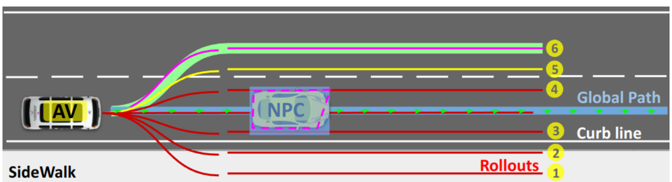

Figure 1.8: OpenPlanner, local and global path planning

1.9 Motivation and research gaps

The development and deployment of AVs have the potential to revolutionize transportation systems and make them safer, more efficient, and more sustainable. The reliability, speed, and ability of computers today to eliminate human driver vulnerabilities and prevent risks are undeniable. The assumption that their decisions will be flawless, however, is naive [8]. Researchers are increasingly warning that when transitioning to AV, human-driven vehicles and AVs will have to coexist for considerable periods of time [117]. It remains a critical challenge, however, to ensure the safety and performance of AVs. Testing in the real world alone is limited in its ability to comprehensively assess the vehicles' performance under complex and unpredictable circumstances. It is expensive, time-consuming, and potentially risky. The use of simulation testing has emerged as a crucial strategy for overcoming these limitations, providing a controlled and repeatable virtual environment in which AVs can be evaluated.

The motivation behind this study lies in the need for an effective and efficient V&V process specifically tailored for AVs. These vehicles operate in diverse urban environments, interacting with pedestrians, cyclists, and other vehicles. Validating their safety and performance becomes paramount to ensuring the well-being of passengers, other road users, and infrastructure. By leveraging simulation testing, the proposed framework aims to provide a comprehensive toolkit that enables users to assess the software in the loop of simulations, addressing critical aspects such as perception, decision-making, control, and planning algorithms. Listed below are some of the motivations for the simulation testing that supported this study:

Safety: AV safety is a primary concern, and simulation-based verification and validation assist in ensuring that AVs are capable of working safely in a variety of situations.

Cost-effectiveness: Conducting physical tests on AVs is expensive and time-consuming. Simulation-based verification and validation provide a cost-effective and efficient way to test and validate the performance of AVs.

Scalability: Simulation-based verification and validation enable the testing of AVs in a wide range of scenarios and conditions, including extreme conditions that may be difficult to replicate in real-world testing.

Flexibility: Simulation-based verification and validation also provide greater flexibility in testing different software and hardware configurations and enable testing of various scenarios before the actual deployment of the AV.

Furthermore, the motivation stems from the desire to advance AV development and deployment. It is crucial to build public trust in this transformative technology in order to ensure its widespread acceptance and integration into existing transportation systems. Through the framework, we aim to contribute to this trust-building process by providing a reliable and robust validation approach. The framework's scalability, scenario reproducibility, and safety assurance capabilities ensure a thorough evaluation of the AV's behavior, thereby enhancing its reliability, predictability, and performance.

Overall, the study's motivation lies in bridging the gap between real-world testing and comprehensive AV validation. By harnessing the power of simulation testing, this research aims to provide a valuable tool that not only accelerates the validation process but also enhances AV safety and performance. The framework's benefits, such as cost and time efficiency, scenario reproducibility, and iterative development support, align with the industry's motivation to bring AVs to the forefront of transportation innovation while maintaining the highest standards of safety and reliability.

As with any cutting-edge technology, simulation V&V methods are still in development and require high levels of industry and academic contribution. According to the state-of-the-art methodologies reviewed in this study, there are still a number of research gaps in this area that need to be addressed to ensure the safety of AVs. These research gaps include the need for improved data collection, testing, and evaluation of V&V methods to ensure accuracy and reliability. Additionally, V&V methods need better integration into existing development processes. Finally, better tools and techniques are needed to streamline the V&V process. Overall, this study is intended to contribute to the following research gaps:

Uncertainty modeling: AVs operate in complex and uncertain environments, and there is still a gap in how to model and simulate uncertainty in the verification and validation process.

Realism: The accuracy and realism of the simulation models used in the verification and validation of AVs are critical to ensuring the validity of the results. There is a lack of attention given to improving simulation fidelity, including the incorporation of real-world data, which increases the realism of simulation models.

Metrics and standards: There is a research gap for standardized metrics and evaluation criteria for simulation-based verification and validation of AVs. This helps ensure consistency in testing and evaluation across different testing environments.

Integration: AVs are complex systems that rely on the integration of multiple subsystems and components. There is a gap in research on how to effectively simulate and test the integration of these subsystems and components in a virtual environment.

1.10 Objectives and research questions

So far the importance of the V&V method of AVs has been identified. Therefore, this research aims to develop a comprehensive structure for evaluating the vehicle's safety, performance, and behavior in various conditions by focusing on the following Research Objectives (ROs). It examines existing quality assurance approaches, identifies challenges, and proposes solutions. This study addresses the following primary research objectives.

Research Objectives :

- **RO1:** To provide a scenario-based assessment toolkit based on the simulation that enables the evaluation of the AV's performance in various scenarios representing real-world driving conditions, such as heavy traffic, adverse weather, and unexpected obstacles.
- **RO2:** To be able to test the AV's decision-making and perceptions algorithms and identify potential failure modes and safety hazards, at a system and unit level, in various scenarios, and to validate that these algorithms produce safe and efficient behavior in different driving situations.
- **RO3:** To be able to modify the fidelity of the proposed toolkit based on the evaluation requirements. Moreover, to compare the results of different fidelity levels.
- **RO4:** To evaluate the system against advanced cyber threats, specifically those targeting sensors and controllers.
- **RO5:** To compare the simulation results with real-world testing data and field experience, and to validate the accuracy and relevance of the simulation approach for verifying and validating the AV's behavior.

Overall, the objective of this study can be summarized as providing a systematic toolkit that allows a comprehensive evaluation of the vehicle's safety, performance, and behavior in different scenarios, and to validate that the vehicle meets the required standards and regulations for autonomous driving. This research addresses the following questions in particular:

Research Questions :

- **RQ1:** What are the main challenges in performing software V&V of safe AVs?

- **RQ2:** What are the open issues and opportunities in software V&V of safe AVs?
- **RQ3:** What are the key factors that affect the accuracy and reliability of simulation-based V&V of AVs, and how can these factors be addressed to improve the simulation results?
- **RQ4:** How can simulation be integrated with other verification and validation methods, such as physical testing and field trials, to provide a more comprehensive evaluation of AVs?
- **RQ5:** How can the proposed approach be used to verify and validate the resilience and cyber security of AVs against potential cyber threats and attacks?

Scientific Contributions

Table 1.2 maps how the included papers contribute to the research questions and document the scientific contributions.

Table 1.2: Relationship between the research questions and the included papers.

RQ	Paper I	Paper II	Paper III	Paper IV	Paper V
RQ1	✓		✓		✓
RQ2	✓		✓		✓
RQ3		✓	✓	✓	✓
RQ4				✓	✓
RQ5				✓	

✓Indicates the relevance of the corresponding research question.

2 The Validation Toolkit

In this section, a brief but comprehensive description of the evaluation methodology proposed in this study is presented. This methodology as a toolkit can be used to evaluate the safety, security, and performance of a DUT (refer to **RO1**). In this toolkit, three main steps are included: (A) scenario generation, (B) Software in Loop (SiL) or Hardware in Loop (HiL) simulation, and (C) results analysis. Figure 2.1 illustrates the main steps of the toolkit, indicating different platforms that are capable of being used at each stage. The toolkit is highly flexible and allows users to conduct low- and high-fidelity simulations with SiL or HiL configurations. A detailed discussion of these steps will follow.

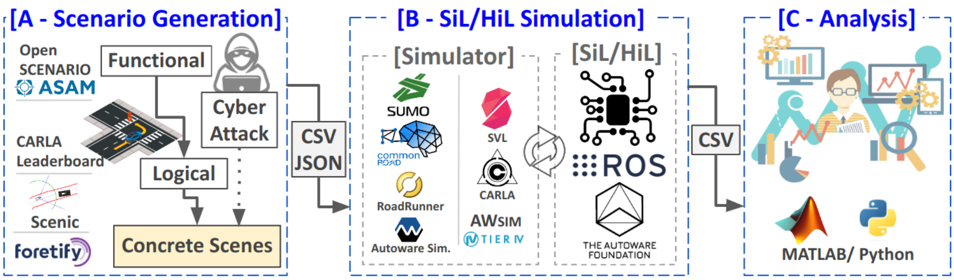

Figure 2.1: Three main steps of the evaluation toolkit

2.1 Scenario Generation

A scenario is the main plan for testing, whether it is a road test or a simulation. As discussed earlier, designing an appropriate scenario will have a significant impact on the efficiency and validity of testing. Having an adequate scenario plan will facilitate the detection of failures more quickly. Our proposed toolkit begins with the generation of scenarios (see Fig. 2.1 [A]). There are three levels of abstraction used to generate scenarios: **Functional**, **Logical**, and **Concrete**. In this study, scenarios in which a cyber attack has happened (refer to **RO4**) are also considered. In Paper IV [118] included in this thesis, several cyber attacks were simulated with the help of the proposed regime.

A functional scenario is described using natural language at the highest level of abstraction. After formalizing a functional scenario, a logical scenario L can be expressed by a state space and its interrelations. The set of m logical scenarios is denoted by $\mathcal{L} = L_1, ..., L_m, \forall m \in \mathbb{N}$. L_m includes a list of essential parameters $\mathcal{A}_L = (\alpha_1, ..., \alpha_n)$ and their value ranges $\mathcal{R}_L = (V_1, ..., V_n)$ where $\forall n \in \mathbb{N} : V_i \subseteq \mathbb{R}$. The state space of L can be described by $\mathcal{V}_L = V_1 \times ... \times V_n \subseteq \mathbb{R}^n$. A correlation between parameters and numerical constraints may also be included as an option. Next, a concrete scenario C requires a single value for each parameter. Therefore, for a logical scenario L_m, $m \in \mathbb{N}$ concrete scenarios C_i, $i \in \mathbb{N}$ can be derived by instantiating all parameters $\alpha_i \in \mathcal{A}_L$ with some $v_i \in V_i$, for instance selecting a $v \in \mathcal{V}_L$. A set of concrete scenarios is denoted by $\mathcal{C}$. Accordingly, $\mathcal{C}_m$ is the set of all concrete scenarios derived from the logical scenario L_m.

Several scenario description languages (SDLs) can be used based on the testing objectives to generate and translate human-readable plans into machine-readable scenarios. Users can generate meaningful simulation scenarios using platforms such as Scenic, M-SDL (introduced by Foretify™ [119]), OpenSCENARIO (by ASAM [120]), and CARLA leaderboard (ScenarioRunner) [121]. In this research, CARLA ScenarioRunner has been used to configure our concrete scenarios for simulations. In the final stage, generated scenarios are exported as a CSV or JSON file format for further processing.

2.2 SiL/HiL Simulations

Once the concrete scenarios have been prepared, it is time to conduct the test. This step is the foundation of the proposed evaluation method (see Fig. 2.1 [B]). Depending on the test objectives, users will choose whether to conduct a SiL simulation or a HiL simulation. Furthermore, the level of testing, whether it is at the unit or system level, serves as a measure of the simulator's fidelity (refer to **RO3**). In the current study, SUMO, Autoware Simulator, SVL, and CARLA have been utilized as simulators to demonstrate the efficiency of the approach. An in-depth discussion of these simulators can be found in section 1. The following sections examine simulations in low- and high-fidelity configurations.

2.2.1 Low-Fidelity Simulations

In this type of simulation, the focus is on a specific unit of the system (e.g. motion planner algorithms) and sacrifice unnecessary details to achieve faster evaluation speeds and, as a result, analyze a larger number of scenarios. It should be noted that in some cases, this may affect the accuracy of the results. This type of simulation, however, has been shown to be useful for identifying critical scenarios and optimizing planning algorithms, according to the literature.

A low-fidelity MATLAB test setup was used in Paper I [122] to look for critical scenarios. MATLAB Automated Driving Toolbox was used to simulate the TalTech campus track designated for TalTech iseAuto (see figure 2.2). This test setup is integrated into a closed-loop simulation where the AV is controlled by the TalTech iseAuto autonomous controller. Figure 2.3 illustrates the workflow for identifying critical scenarios and improving the performance of the motion planner by means of a closed-loop simulation (refer to **RO2**).

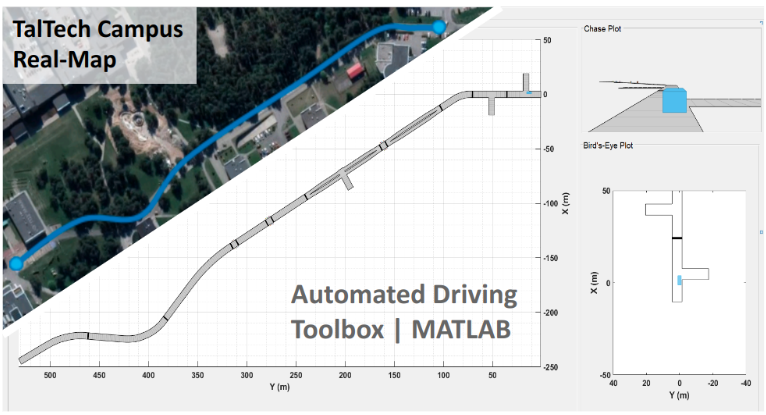

Figure 2.2: MATLAB Automated Driving Toolbox, simulated TalTech campus track

In [123], the Autoware low-fidelity simulator, a ROS-based testing platform, is used to test a cyberattack on the planning algorithm of the Autoware.ai autonomous software. Figure 2.4 illustrates the simulator setup used in the tests. A kinematic simulator engine and a fake perception node are the main components of the Autoware simulator (refer to **RO2**). The fake perception provides data for localization and detection of other actors added to the scenario setup during the simulation. Autoware's main perception nodes are bypassed during simulation.

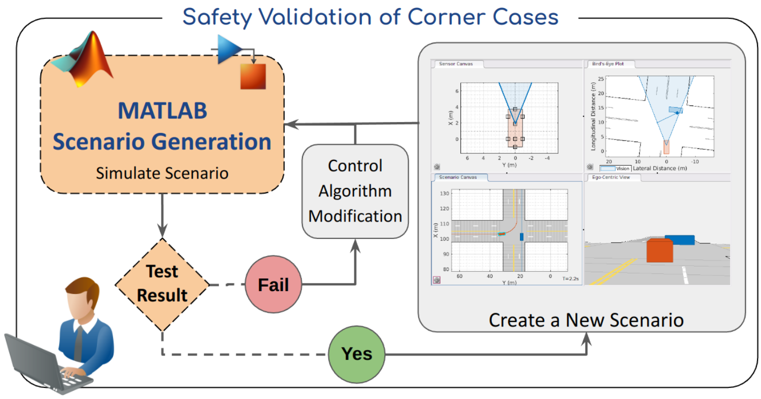

Figure 2.3: Workflow of the MATLAB closed-loop simulation to identify corner case scenarios.

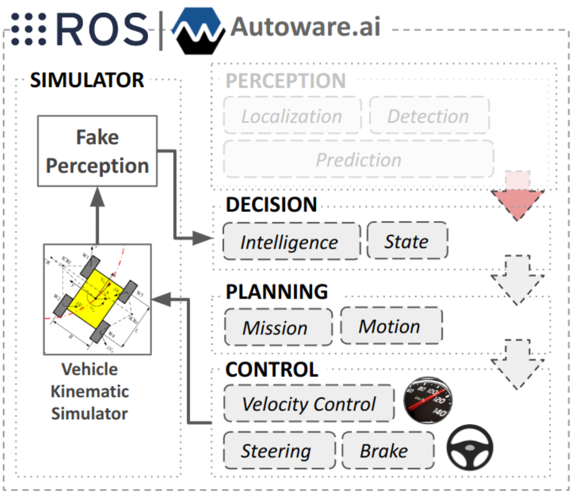

Figure 2.4: Autoware planning simulator configuration

2.2.2 High-Fidelity Simulations

The term "end-to-end" or "high fidelity" refers to the simulation of scenarios in a realistic virtual environment. Nowadays, with the emergence of giant computer game engines like Unity and Unreal, users have greater access to realistic simulation. Common users can now develop a virtual copy of any object or environment, known as a digital twin, and integrate them into their simulated environment. Simulators of this type are already discussed in detail in section 1. SVL and CARLA were used in this study to validate an autonomous shuttle's safety, security, and performance. Figure 2.5 displays the core module of the high-fidelity simulation conducted by the CARLA simulator. This figure represents a SiL/HiL simulation setup. There has already been an implementation of this simulation setup in a study investigating sensor-based cyberattacks on a public autonomous shuttle (see Paper IV included in this document).

CARLA is based on the Unreal engine communicating through a ROS bridge with autonomous software, Autoware.ai, which is based on the ROS platform. With this setup, Autoware.ai receives the data from the virtual sensors, and all of its core modules, including perception, decision, planning, and control, are engaged in the testing process.

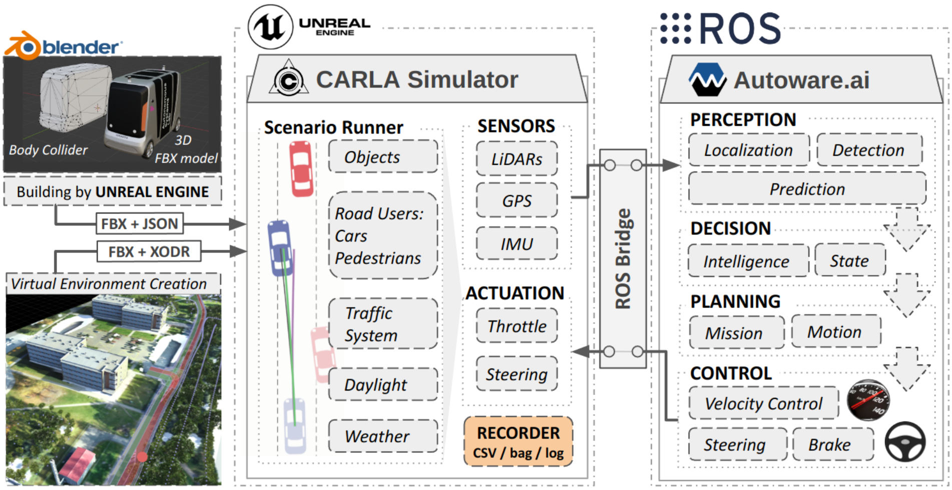

Figure 2.5: CARLA high-fidelity simulation architecture

It then sends the actuation command back to the simulator, closing the simulation loop. CARLA has a built-in module called ScenarioRunner that generates custom scenario plans. Through ScenarioRunner, users can define objects, road users, traffic systems, and weather conditions within a virtual environment. Additionally, it is capable of importing scenarios from other platforms, such as SUMO, OpenSCENARIO, etc. For a more realistic result, this platform allows users to customize their vehicle and environment models. Following is a brief description of the steps involved in building a digital twin of a vehicle and its environment:

Vehicle Model: Initially, a mesh file must be created for the vehicle in order to build its digital twin. Next, the tires, steering axis, and a simplified collision body should be defined in accordance with the simulator instructions. This can be accomplished using 3D modeling software such as Blender [124]. Unreal needs an additional simplified mesh called "ray cast sensor mesh" that sets up the vehicle's shape that will be detected by the ray cast sensors (RADAR, LiDAR, and Semantic LiDAR). Afterward, these FBX files should be imported into the Unreal CARLA project in order to configure all textures, materials, and vehicle dynamics. Vehicle sensor configurations can be stored in a JSON file and loaded by the agent-wrapper module. Figure 2.6 displays the 3D graphical model of the TalTech iseAuto inside the CARLA environment.

Figure 2.6: 3D model of TalTech iseAuto integrated into CARLA virtual environment.

Environment Model: Creating new environments allows users to test more diverse scenarios. Test locations such as complex urban environments can challenge AVs to handle difficult situations. In particular, the corresponding virtual environment can be used to test an AV before implementation in a real-world environment. There are two main ways to create a realistic graphical environment. The first method involves processing aerial imagery and converting it to terrain. The steps are shown in figure 2.7. A description of these steps can be found in Paper II [125]. The second method is using a high-definition map (e.g. "xodr" OpenDrive) of a desired area and converting it to a graphical virtual environment. MATLAB RoadRunner, for example, is one of the toolboxes that uses the OpenDRIVE format to convert a map into a graphical virtual environment.

Figure 2.7: Steps to build a virtual environment from aerial imagery

2.3 Results Analysis

Analyzing the simulation report is the last chain of the proposed toolkit. To maximize the effectiveness of simulation analysis, it is also essential to define appropriate metrics for evaluating the analysis results. Particularly with large numbers of runs, it is almost impossible to check the results manually. For this reason, metrics are expected to detect criticalities and violations during simulation. Several criticality metrics can be used based on the type and priority of the analysis, including time, distance, intensity, and velocity metrics described in detail in [126, 127]. A variety of metrics based on acceleration, velocity, distance, and intensity have been used in this study. Table 2.1 lists some of the metrics used in the research. The listed metrics are deployed in the next section for different studies.

Table 2.1: Safety and Performance Evaluation Criteria

Safety Criteria	Label	Description	Metric
Collision	Col	AV collides with NPC	Pass/Fail
Distance-to-Collision	DTC	Violation of the safe distance between AV and NPC	AV within 0.5m of other vehicle
Deceleration	Acc	Sharp deceleration	Greater than 6 (m/s^2)
Break on Driving Lane	BrD	AV initiates emergency break on driving lane	Pass/Fail
Break on Passing Lane	BrP	AV initiates emergency break on passing lane	Pass/Fail
Performance Criteria	**Label**	**Description**	**Metric**
Succeed	Suce	AV Successful complete the mission	Pass/Fail
Not Finished	NotF	Failure to finish the mission	Pass/Fail
Localization	NDT	AV lost its localization	NDT score greater than 100
Long Pass	LoPa	DUT passes the NPC but does not return the lane	Distance away > 25 m

It is necessary to record all the necessary data in order to observe any criteria violations. This data is monitored during the simulation (online) to identify metrics violations and after the simulation (offline) to investigate the cause of the violation. The online observer is a Python script whereas the offline analyzer is MATLAB software. It should be noted that, in case of any accident in simulation, the analyzer produces an accident sketch and analyzes the crash intensity to categorize crashes based on their criticality. Log files are recorded in CSV format for ease of use. A large set of scenarios, however, require the use of an SQL database, particularly if the data will be shared with other researchers.

3 Case Studies and Results

In this section, the results obtained from the comprehensive scenario-based SiL/HiL simulation toolkit for the safety, security, and performance evaluation of AVs are presented. The SiL/HiL simulation technique enables us to assess the effectiveness and reliability of the software under realistic scenarios while minimizing risks and costs associated with physical testing. By executing a series of carefully designed scenarios, the system's behavior and overall performance are evaluated, shedding light on its capabilities and potential areas for improvement. The aim of this section is to provide an overview of the implementation of our toolkit and highlight the significance of the results obtained, setting the stage for a detailed analysis of the findings to follow. The findings of this study are derived based on simulations and real-world experiments. In the following subsections, simulation results are categorized according to the level of simulator fidelity.

3.1 Study Used-Case Scenario

One of the most challenging maneuvers for AVs is passing and overtaking. To improve the safety and performance of the TalTech iseAuto, this issue was used as the main scenario plan to showcase the implementation of the evaluation toolkit. Here, a brief review of the overtaking scenario primarily used in the research is presented. Figure 3.1 illustrates the overtaking process, including the cut-out, passing of the NPC, and cut-in maneuvers.

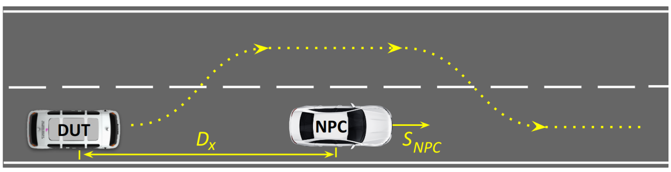

Figure 3.1: Overtaking scenario sketch and its parameters

For simplicity, two parameters were chosen to describe the scenario. Dx and S_{NPC} represent the initial longitudinal relative distance between DUT and NPC and the NPC constant speed respectively. A brief definition of the parameters and actors' task is listed in Table 3.1. Each study will identify parameter ranges at a later stage.

Table 3.1: Target scenarios definition

Actor	Speed (m/s)	D_x (m)	Goal
DUT	$[0:6]$	0	overtake the NPC safely
NPC	$[0:S_{max}]$	$[D_{min}:D_{max}]$	drive straight

3.2 Low-Fidelity Testing Setup

It has already been discussed that this type of simulation is useful for finding critical scenarios and for unit testing (e.g. planning algorithms). In the following, the findings of some of the studies conducted with a low-fidelity simulator will be presented. This section covers **RO2,3,4**.

3.2.1 Optimization of Low-level Planning Parameters

As part of another study [115], authors used the MATLAB Automated Driving Toolbox to optimize algorithms for smother mission and motion planning. After 500 simulation runs, using a Genetic Algorithm optimization solution, the optimal parameters for sigmoid-based trajectory generation and accurate path tracking were determined. Figure 3.2 illustrates two different passing simulations (a and b) performed by the AV employing the default and optimal parameters values respectively. In each run, the figure displays the desired path defined by the mission planner besides the path that the DUT traveled. Based on the lowest steering effort and least trajectory following error, optimization reached the performance presented in the inset of Figure 3.2.(b).

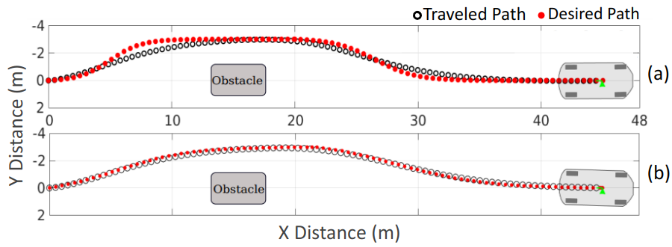

Figure 3.2: Optimization carried out by means of MATLAB low-fidelity simulation on s trajectory generator and following algorithm (a) simulation with default values and (b) simulation with optimized values for algorithm parameters

Additionally, two different experiments were conducted in that study to demonstrate the efficiency of the algorithms after optimization (see figure 3.3). First, with the default non-optimized Astar planner, and then with the proposed optimized Sigmoid planner. According to the figure, steering output command efficiency and smoothness have significantly improved.

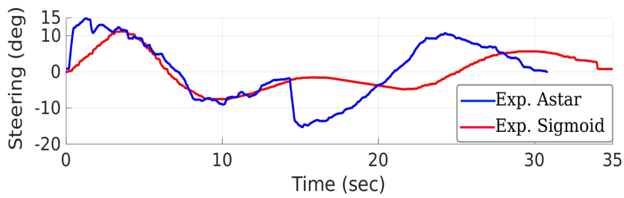

Figure 3.3: Steering angle recorded from two experiments with different mission planners in an overtaking maneuver; Non-optimized Astar and Optimized Sigmoid algorithms

3.2.2 Security Evaluation

Low-fidelity platforms are useful for simulating cyberattacks on the low level of the autonomous system (e.g. GPS and IMU sensors). Different attack models including position offset and message time-delay were observed during an overtaking mission in [123]. An overtaking maneuver is illustrated in Figure 3.4 through three sequences captured during simulation from Rviz. Findings suggested the planner's vulnerabilities against cyberattacks. According to Figure 3.5, different types of attacks have caused AV to violate safety criteria (b,c,d), whereas in the non-cyber case, it succeeded in overtaking (a).

Figure 3.6 shows the study's metric evaluation results derived from simulation tests on a simple overtaking mission. Simulations are performed to observe metrics violations

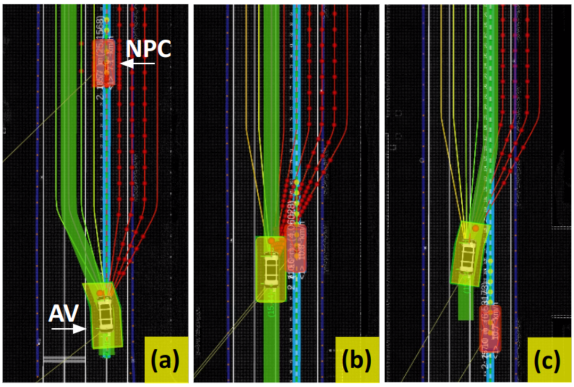

Figure 3.4: Three sequences of an overtaking mission simulated in the low-fidelity simulator, a) starting the mission b) passing the NPC c) AV cuts in

in pure safety, attack model 1 (position offset), and attack model 2 (message time-delay) cases. In pure safety cases, 300 repetitions are conducted to reach a meaningful statistical population. According to the pure-safety experiments bar, almost half of the experiments were completely successful in completing the target mission. However, other safety violations are observed as follows: 15% collisions, 10% DTC violations, about 20% emergency brake in the passing lane, and 5% in the driving lane. Each attack case is repeated 100 times with different sensitivity deviations for the target attack parameter. A conclusion can be drawn from the attack case experiment bars regarding how those attacks affected the safety of the target shuttle. There were cases in which the attack resulted in more brake on passing lane violations significantly (1a, 1b, 1c) and other cases in which it resulted in more collisions (1d, 1e, 1f). As well, time-delay attacks (2a,2b,2c) demonstrate that they are less effective than position offset attacks in reducing the success rate in the tests. The 2c attack configuration, however, leads to a doubled collision rate. Overall, this figure shows how different cyber attack models can alter AV safety evaluation results.

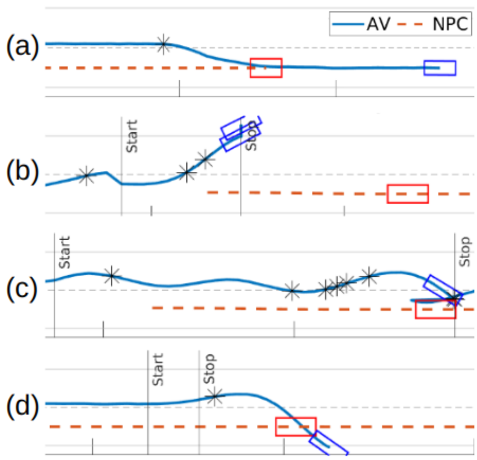

Figure 3.5: 2D representation of an overtaking simulation under different situations. a) a successful overtaking safety simulation, b-d) under a cyber attack that led to a safety violation (collision). The vertical lines identify the start and stop point of the attack

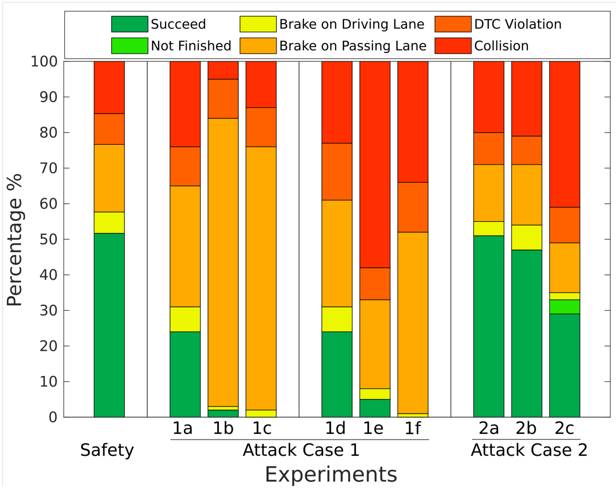

Figure 3.6: All simulation results with different cyberattack models based on the proposed safety criteria

3.3 High-Fidelity Testing Setup

A high level of detail in simulation allows users to achieve more realistic and reliable results. There is, however, a cost associated with setting up the virtual environment and the utilization of computing resources. Described in this section are some of the implementations of the high-fidelity proposed toolkit for the evaluation of autonomous shuttle safety and security. This section covers **RO2,3,4,5**.

3.3.1 Scenario Definition

As discussed in Section 2, creating a scenario plan is the first step in the proposed evaluation toolkit. A third-party platform and software can be used to accomplish this. As part of this study [128], authors contributed to the development of an open-source validation and verification framework, PolyVerif, by implementing it on the TalTech iseAuto testbed. Scenic was used to generate probabilistic scenarios for the TalTech test track and the SVL simulator to create high-fidelity simulations based on those plans (see figure 3.7). A domain-specific language is used in Scenic to describe scenarios that are distributed over scenes and the behaviors of actors over time.

3.3.2 Safety and Performance Improvements

Perception is one of the most important components of an autonomous system. It interprets the working environment for AVs. Testing the perception module which deploys advanced sensors including RGB cameras, Radars, and LiDARs requires a high-fidelity simulator. In Paper III [129], a high-fidelity simulation platform for testing and developing AVs is proposed. For this study, the AV model was built within the simulator platform and configured with all LiDAR sensors as shown in Figure 3.8. There is a Velodyne VLP-32 mounted at the top of the shuttle's front, and a Velodyne VLP-16 mounted at the top of the shuttle's back. They are primarily responsible for scanning road objects and determining the location of the vehicle. Then, both the left and right sides of the vehicle are equipped with Bpearl sensors from Robosense which cover the sides of the vehicle for maneuvers.

For perception, this LiDAR configuration creates an adequate point-cloud coverage

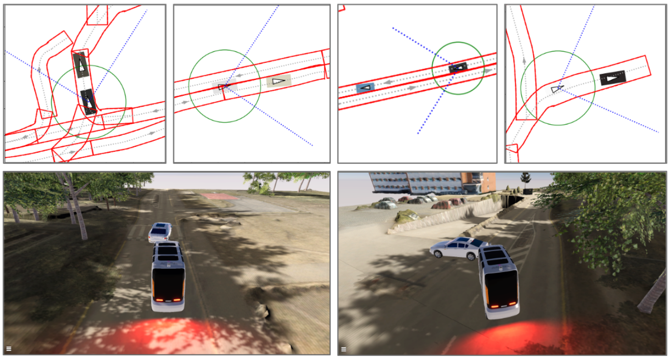

Figure 3.7: Different scenario plans generated by scenic over the TalTech track map and simulated in SVL

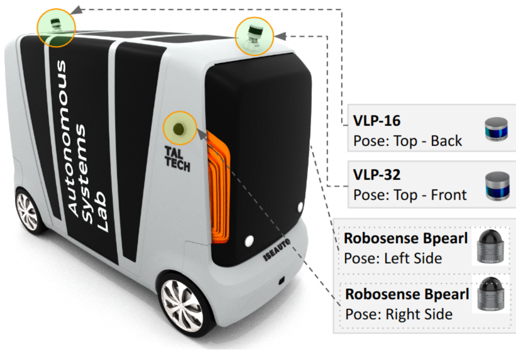

Figure 3.8: Digital twin of the TalTech iseAuto and its LiDAR sensor configuration

around the shuttle as shown in Figure 3.9. The simulator enabled us to observe the quality of LiDAR's perception and how other objects are seen in the autonomous software. Moreover, it is examined how point-cloud filtration affects the AV's ability to detect objects. Finally, improvements were examined by an overtaking maneuver with an additional vehicle (NPC2) trying to overtake AV and NPC1. First, it was expected to detect both NPCs with its LiDAR sensors and secondly, to overtake NPC1 considering the approaching vehicle (NPC2). Figure 3.10 shows simulation frames that validate the detection of both NPCs and safe maneuvers.

3.3.3 Security Evaluation

The advent of virtual sensor modules in high-fidelity simulator engines allows the simulation of advanced sensor cyberattacks. One of the most common cyberattacks on LiDAR sensors is point spoofing. In this research, Paper IV [118], the aim was to develop a method of evaluating system-to-system interactions in developmental AD algorithms by combining safety and cyber security testing. As part of the study, cyber-attack simulations were conducted based on the proposed combined methodology and examined how they affect AD algorithms. The LiDAR spoofing attack was chosen since it is a realistic attack that can take place in the TalTech shuttle's driving environment. As the shuttle's main perception system is based on LiDAR sensors, spoofing attacks can affect driving behavior and result in collisions, emergency braking, and lane change violations.

A spoofing attack scenario was set up as shown in Figure 3.11. As soon as the AV begins

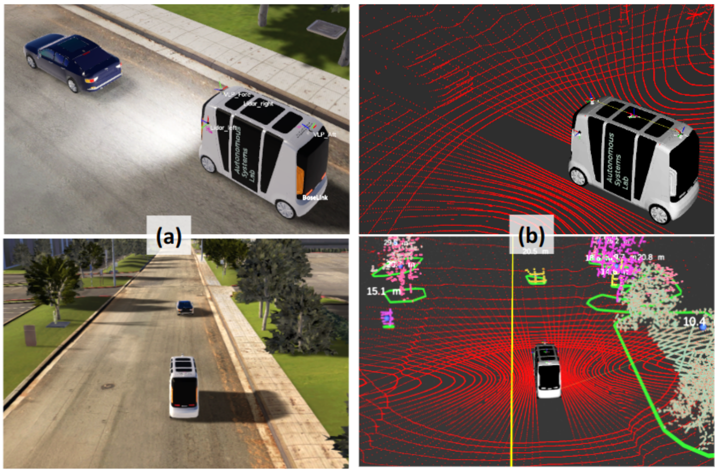

Figure 3.9: Virtual sensors provided by the high-fidelity simulator perceive the virtual environment

to overtake an NPC, an attacker on the other side of the road starts to expose the AV's front LiDAR sensor with an external laser beam. As a result of this laser light, spoof points are created in the direction of exposure. An attack commences from a position relative to the NPC and continues for a specific period of time. During the study, parameters such as density, frequency, duration, and position of the AV were considered to create different attack experiments.

Simulations were conducted using the CARLA high-fidelity simulator. Figure 3.12 illustrates how the combined testing methodology was set up for simulating the spoof points attack. In addition to the main virtual sensor data stream, attack false points (FPs) are entered into the main perception algorithms. Mission and motion planning nodes are directly influenced by perception outputs as shown in the figure.

Figure 3.13 graphs simulation outcomes for cyber and non-cyber scenarios. With the proposed combined approach, at first 15 non-cyber overtaking scenarios were simulated to examine the safety assurance condition of the current operating AD algorithm. For a sufficient statistical population, all simulation scenarios were repeated 50 times. After reviewing the safety violation graph (see Figure 3.13.a), two different spoofing attacks were conducted on scenario 2 (the least safe one scenario based on the result from Figure 3.13 (a)) and scenario 10 (the safest scenario). Figure 3.13.b and Figure 3.13.c represent cyber simulations for scenario 2 and scenario 10 respectively.

In each cyber test scenario, the influence of four proposed attack parameters including the FPs density, the FPs frequency, the attack duration, and the attack location was quantified using the Taguchi design of experiment (DOE) [130]. A total of nine experiments were designed with three different levels for each of the four attack factors. As a result, the Taguchi L9 matrix was suggested.

A significant drop can be seen in the safety assurance level for scenario 10 cases that reveal a noticeable impact of the cyber attack. In particular, the LiDAR spoofing attack increases the number of collisions and emergency brakings in the passing lane, as well as the number of safety violations. This can also be seen in cyber scenario 2. It is observed that the Euclidean clustering and kf_countour nodes detected the spoofed LiDAR injection as an object in the safety violation cases. Due to this false positive detection, the local planning is affected, thereby forcing the AV to make the cut-in during the overtaking maneuver. In particular, as the adversarial LiDAR device is placed to the left of the AV, the

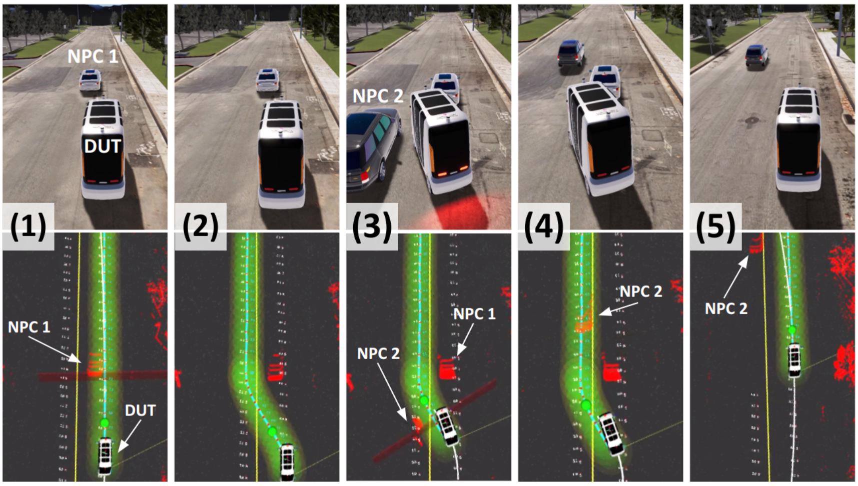

Figure 3.10: 5 sequence frames of simulation of an overtaking maneuver while another car (NPC2) approaches from behind inside the high-fidelity simulator

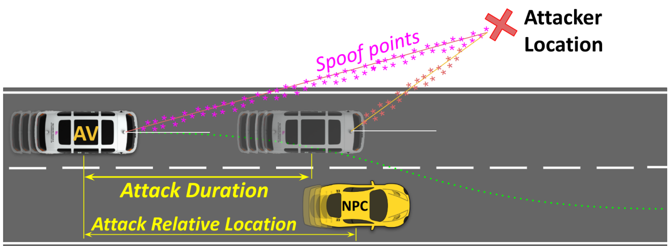

Figure 3.11: Attack scheme of the LiDAR sensor spoofing

trajectory evaluator blocks the roll-outs on the left side. As a result, the AV is forced to veer right and attempt the cut-in procedure, which results in safety violations, predominantly collisions or DTC.

3.4 Combined Low and High-Fidelity Testing

Different fidelity-level simulations offer various benefits, including fast evaluation and detailed analysis. In the validation process, they can, however, be combined to generate more accurate and quicker results. The following two studies were conducted employing two different levels of fidelity simulations to investigate the validation process performance.

3.4.1 Two-Layer Validation Regime

In the study presented in Paper V, scenarios of a passing maneuver for the TalTech iseAuto shuttle were simulated in a two-layer approach. A simplified representation of this process can be found in Figure 3.14. Fortify ™ [119] and MSDL were utilized to describe distributed scenarios in the defined scenario domain. The resulting scenarios were then simulated by SUMO, a 2D low-fidelity microscopic simulator. In this step, the DUT was controlled based on the criteria defined in the scenario setup without using autonomous software.

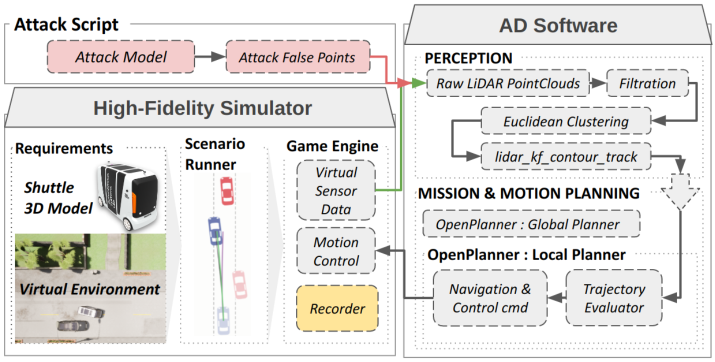

Figure 3.12: Combined safety and security testing methodology architecture

The simulated scenarios were then analyzed, and parts of them were selected for high-fidelity simulations. SVL was used to perform high-fidelity simulations. As opposed to the previous step, these tests involved autonomous software (Autoware.ai) controlling the DUT within the simulation.

Figure 3.15 illustrates each of the aforementioned steps in detail. The test scenario description is transformed from functional to logical and then to a more concrete abstraction level. Each actor in the scenario is described using the minimum number of parameters necessary. These sub-steps are shown in the (see Figure 3.15, step-A, scenario description language box).

According to the target scenario, the DUT intended to pass the immobile NPC. The logical scenario is described by defining two relative parameter ranges, the longitudinal distance between the DUT and NPC and the NPC lateral shift, Dx and Dy respectively as depicted in Figure 3.16. Table 3.2 reports the scenario's required parameter ranges at the logical level.

The speed range for the DUT is 0 to 6 m/s which is controlled by autonomous software. There is an NPC parked along the road, immobile, and initially located between 5 and 50 m from the DUT. The scenario generator also defines a small lateral shift, Dy. In order to proceed down the road, the DUT must maneuver safely around the NPC that is parked.

Table 3.2: Target scenarios definition

Actor	Speed (m/s)	$[D_x, D_y]$ (m)	Goal
DUT	[0 - 6]	[0, 0]	To overtake the NPC safely
NPC	0	[5:50, -0.4:0.4]	To stay immobile

120 concrete scenarios were generated and simulated by the low-fidelity platform (see Figure 3.15, step-B, SUMO simulations) to begin a case study using the proposed platform. As illustrated in Figure 3.17, each point represents the NPC location. The results are divided into "Failure" and "Success" groups based on crash criteria. According to the figure, the failure probability is examined in three different regions.

Table 3.3 summarizes the number of scenarios in the subdivided areas in two main categories. The table indicates that 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. As well, 47$\%$ and 28$\%$ of all failures occurred in [5-10] m and [10-20] m, respectively. 87 scenarios in the range of [5-20] m have been selected for the next step. There are two reasons for this: first, there are more failures seen before 20 m, and second, it is impractical for

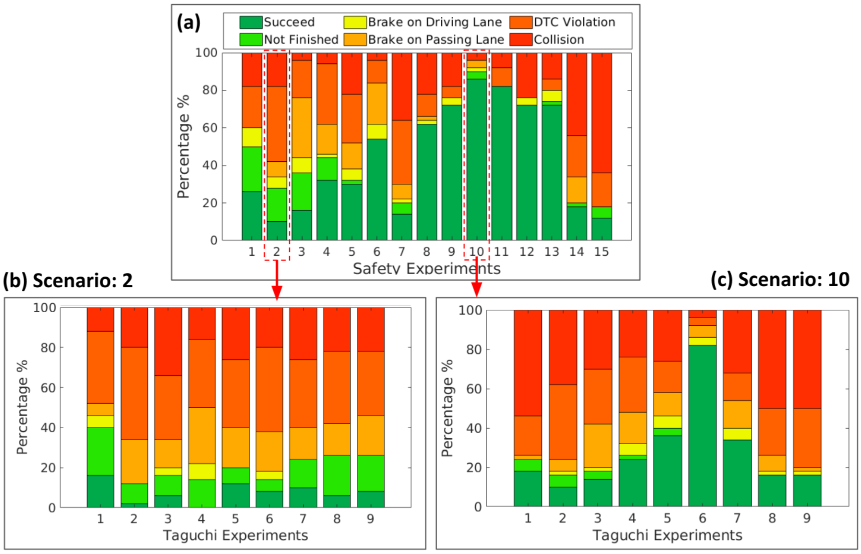

Figure 3.13: Violation results derived from simulations of 15 distinct overtaking scenarios in CARLA high-fidelity simulator; (a) Pure safety tests (b) and (c) Scenario no. 2 and 10, respectively, under spoofing attack with different values for attack parameters. All 9 attack experiments were designed by the Taguchi method.

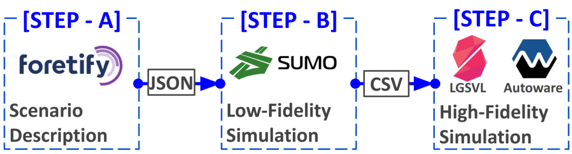

Figure 3.14: Three main steps of the two-layer validation method. The data format transferred between the layers is annotated.

Table 3.3: Failure and Success scenarios in the three regions of D_x

	[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

the shuttle to initiate its passing operation over a distance of 20 m.

Table 3.4 reports a summary of the results for the 87 low-fidelity simulations. It includes the duration of simulations in seconds, the difference in lateral and longitudinal distances between the NPC and the DUT, the average speed of the DUT in meters per second, and the closest distance between actors at any given point in the simulation (DTC).

Using the SiL high-fidelity platform, the selected scenarios were simulated. As part of the evaluation process, we gather and store all the corresponding data of the evaluation metrics in a Rosbag file, along with a general tabular report. The data includes DUT speed, normalized braking intensity, localization score (NDT-score), and the closest distance to the NPC from the DUT during the simulation.

According to Figure 3.18, a spaghetti diagram shows all trajectories traveled by the DUT (curves) next to the location of the NPC (squares) in each scenario. Depending on

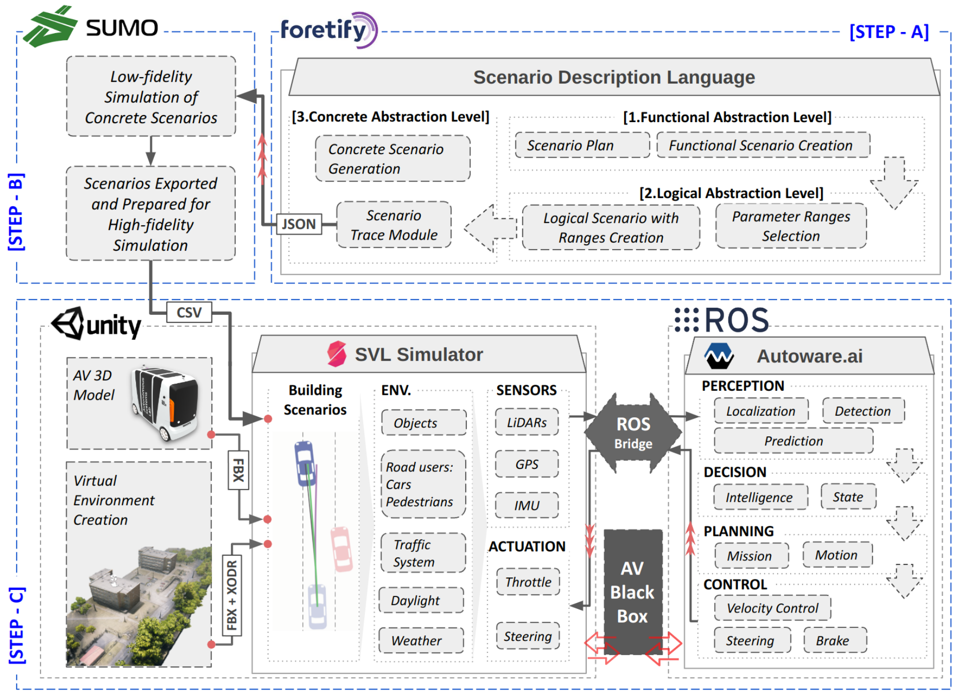

Figure 3.15: High-level architecture of the Two-layer validation regime including scenario generation (Step-A), low-fidelity simulation (Step-B), and high-fidelity simulation with the control software in the loop (Step-C).

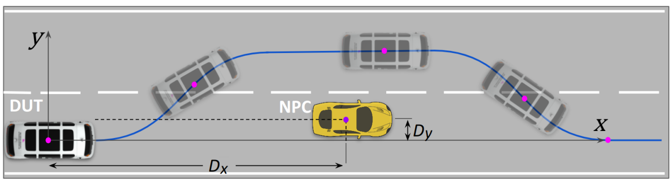

Figure 3.16: Describing a passing scenario by the two relative distance parameters, Dx, and Dy.

Table 3.4: Summary over the 87 runs in the 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
max	98.48	0.37	19.92	4.16	3.81	11.00

the progress of the mission, the results were divided into three groups as follows:

- Not started missions (group 1): This refers to situations in which the DUT was unable to begin the passing maneuver and remained behind the NPC.
- Completed missions (group 2): DUT completes the missions.
- Aborted missions (group 3): The scenarios in which the DUT started the maneuver

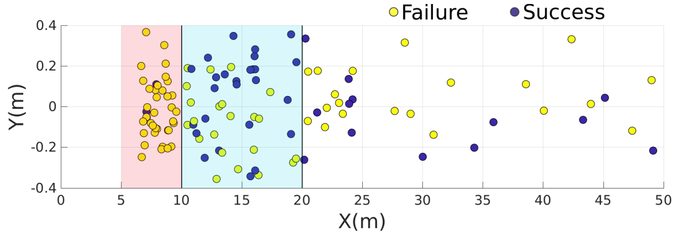

Figure 3.17: Points representing all initial relative NPC locations in 120 scenarios that are marked based on their simulation result.

but could not complete it. If localization is lost, for instance, uncontrolled movements can lead to the failure of the mission.

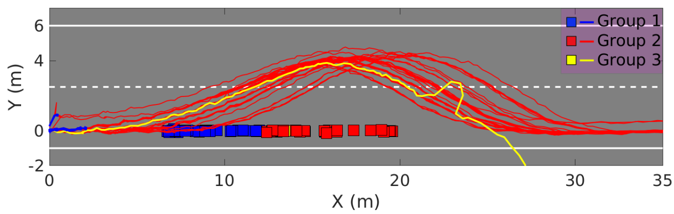

Figure 3.18: All traveled routes in the 87 selected scenarios are shown in the three described groups.

To assess the mission's progress and the probability of an accident, each scenario was shown in Figure 3.19 with the corresponding distance traveled and the minimum DTC during the run. A color was assigned to each scenario (circle) based on its average speed, and three groups were identified. According to the Figure each group can be identified based on a specific DUT average speed range; 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).

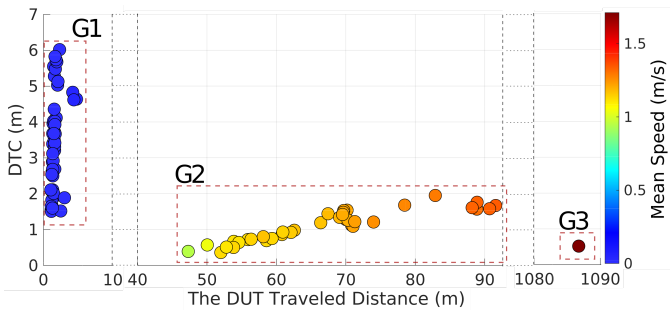

Figure 3.19: High-fidelity simulation results are presented in terms of the traveled distance and the minimum distance to collision (DTC) along with the mean speed of the DUT in the mission. G1, G2, and G3 represent the groups.

To analyze the safe performance of the DUT during operation and check how far the DUT can reach the NPC, each scenario Dx was plotted against the minimum DTC (see Fig. 3.20). Moreover, the color bar indicates how many sharp brakes were applied during the mission which explains the relative safety and comfort of the ride. According to the figure, the DUT did not move during the G1 scenarios with an initial longitudinal distance of less than 12 m, although there were a few scenarios where there was negligible movement. A straight line represents the correlation and trend. Two of the remaining groups are boxed, indicating that the DUT reached the NPC closer than the originally specified distance, indicating that the DUT moved and attempted to pass the NPC. In the G2,3 box, all scenarios in which the DUT succeeded in passing the NPC are included, except for the one with the highest number of emergency brakes. As a result, scenarios where the DUT was more than 12 m behind 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. There was no significant change in the minimum DTC between 16 and 20 m, remaining at approximately 1.7 m. Thus, when the DUT starts the mission from a distance greater than 16 m, the planning algorithms generate a path with a safe distance for the maneuver. Furthermore, to identify edge case scenarios and evaluate the algorithms under critical conditions, it is needed to focus on the range of DTCs that are about to collide (for 12< Dx<16m). Furthermore, software developers should consider making the DUT capable of passing objects less than 12 meters in front of the DUT.

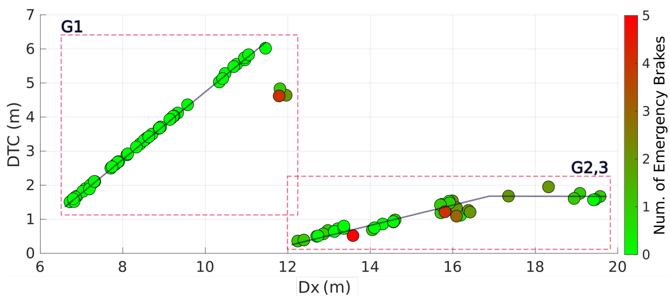

Figure 3.20: Results are represented by the Dx and the minimum DTC during the simulation. The color bar displays the normalized brake magnitude for each scenario.

Table 3.5 provides information about some key aspects of the high-fidelity simulation results, including duration (sec), initial distance to the NPC (lateral and longitudinal) (m), maximum and average speed (m/s), minimum DTC (m), and the maximum NDT score. The high-fidelity platform, on average, took almost twice as long to simulate the same scenario as the low-fidelity one. In the high-fidelity setting, no scenario was completed in less than 46 seconds, while the shortest simulation was completed in less than 3 seconds. It can be seen clearly in this example that the use of low-fidelity simulations can avoid unnecessary simulation computations and thus generate significant time savings. Furthermore, the speed of the DUT in the high-fidelity tests was lower than that of the similar DUT in the low-fidelity simulation, since the vehicle speed is automatically adjusted by the software controlling the DUT (Pure Pursuit Controller[131]). Similarly, none of the high-fidelity simulations produced collisions compared to low-fidelity ones.

Table 3.5: Summary over 87 scenarios runs in 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

The results confirm that simulations based on low-fidelity were faster but likely to be less reliable. This is because of the sacrifice of details and system simplification in these simulations. However, in these simulations, the AV was controlled by the rules specified for the scenario rather than by AV software. Simulators with high fidelity are able to evaluate the autonomous features of AV software simultaneously. Based on the high-fidelity results, developers can explore the algorithms' performance and behavior in the target scenarios without requiring real-life experiments. There is no doubt that the limited number of tests doesn't ensure complete safety, but they can be used to identify more critical and corner cases.

3.4.2 Evaluation with Different Fidelity Levels

Differentiating between low-fidelity (LF) and high-fidelity (HF) simulations and their functionalities is essential. As a part of this study [132], 15000 simulation runs were conducted to compare the evaluation outcome derived from a low- and high-fidelity simulation setup. The study was performed on a DUT overtaking a constant-speed NPC. The Autoware.ai simulator is used for the low-fidelity setup and CARLA for the high-fidelity setup. In both setups, autonomous software is involved in the simulation loops. The focus of this work was on the planning part of the system, which was controlled by OpenPlanner. Two parameters define the scenario domain, the initial relative distance between the DUT and NPC (Dx) and the constant speed of the NPC (Snpc). Table 3.6 provides a detailed description of the parameters used in the study to describe logical scenarios. The concrete scenario was selected from the 2D domain represented by the scenario parameters. The domain was divided into 5X5 tiles (totaling 25 distinct scenarios), with a centric point indicating parameter values.

Table 3.6: Scenario parameters description

Actor	Speed (m/s)	ACC (m/s^2)	D_x (m)	Goal
DUT	[0 : 6]	[-8 : 5]	0	overtake the NPC safely
NPC	[0 : 5]	0	[10 : 25]	drive straight

Table 3.7: Safety Evaluation Criteria

Safety Metrics	Lable	Description
Collision	Col	DUT collides with NPC
Distance-to-Collision	DTC	Violation of the safe distance between DUT and NPC (<40cm)
Deceleration	ACC	Violation of a sharp brake (ACC< -6 m/s^2)
Performance Metrics	**Lable**	**Description**
Succeed	Suce	DUT Successful complete the mission
Long Pass	LoPa	DUT has passed the NPC for 25 m but did not return the main path
Time	Time	DUT did not make progress in the allotted time

Table 3.7 presents the metrics that were considered in the evaluation of the simulations. Collision, DTC, and Deceleration relate to safety concerns, and Long Pass and Time relate to algorithm performance.

Low-Fidelity (LF) Simulations:

Testing begins with low-fidelity simulations. Each scenario was repeated to achieve reasonable statistical reliability. Four different repetitions were carried out to ensure consistency, given that a large number of repetitions would increase simulation time.

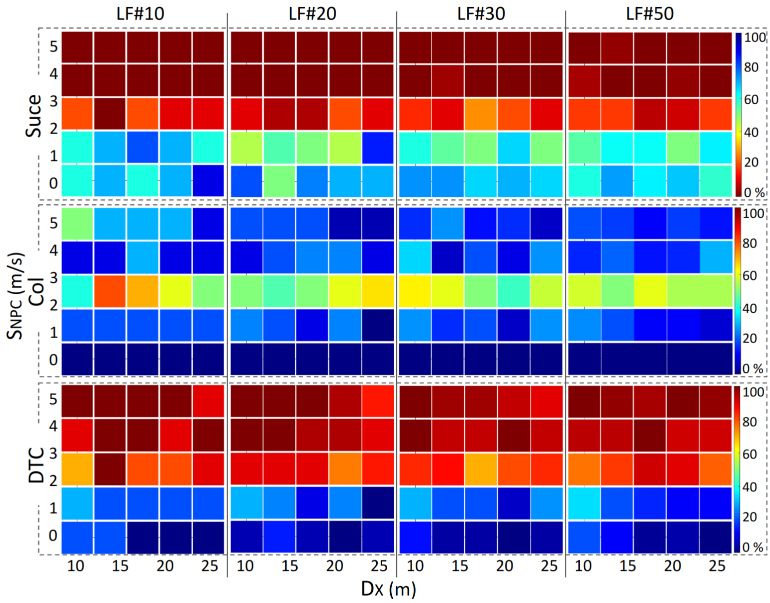

Figure 3.21: Low-Fidelity (LF) simulation results over the scenario domain; each column represents a different repeat number, and each row represents the result of an evaluation metrics.

These four samples are summarized in Figure 3.21. According to the figure, each column represents a simulation set with a specific number of repetitions, each row describes a safety criterion, and each tile indicates the probability of an event occurring within the scenario. In this instance, each safety criterion has almost the same likelihood between LF30 and LF50. The first row in the figure indicates the likelihood that the DUT will successfully complete the mission without committing any safety violations. According to the figure, as the NPC drives faster (vertical axis), there is an increased risk of the DUT failing missions (vertical axis). Taking over the NPC faster than 2.5 m/s was almost impossible for the DUT.

The following row (Col) in the figure indicates which scenarios are most likely to result in a collision. When the NPC was immobile, no crashes were observed, however, when the NPC was moving at 2.5 m/s, the risk of collision grew significantly. In more than 85 percent of the observed collisions, the DUT took a sharp cut-in and the NPC collided with the right side of the DUT. The non-optimized planner and the prediction module, which do not consider the passing NPC, are responsible for these collisions. However, at speeds exceeding 2.5 m/s, the collision rate dropped as the DUT was unable to accelerate sufficiently to capture the NPC. Instead, DTC violations occurred (see third row), indicating that the distance between the DUT and the NPC was unsafe.

Furthermore, other metrics including ACC, LoPa, and Time were considered in the evaluation of the LF30 sample. As shown in Figure 3.22, sharp decelerations are more likely to occur when the DUT violates the DTC. Long pass violations occur when the DUT is in the opposing lane 25 meters ahead of the NPC and does not return to the main route.

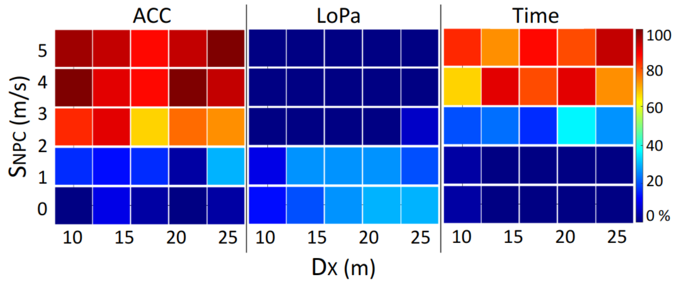

Figure 3.22: Result of the LF#30 simulation evaluation over the scenario domain for ACC, LoPa, and Time metrics.

The LoPa metric results show that the DUT only violates long passes at speeds below 2.5 m/s. By analyzing time metrics, it can be concluded that the probability of an NPC not completing a mission increases as the speed of the NPC exceeds 2.5 m/s.

Using the same platform, GA optimization was conducted to establish optimal values for 10 planner parameters to improve the evaluation outcome. The best planner parameters were selected after 1700 attempts with five repetitions of each. A new set of simulations with 30 repetitions was conducted in the scenario domain using the optimal parameters (marked as LF(a)). Figure 3.23 shows the promising results of the evaluation of the metrics.

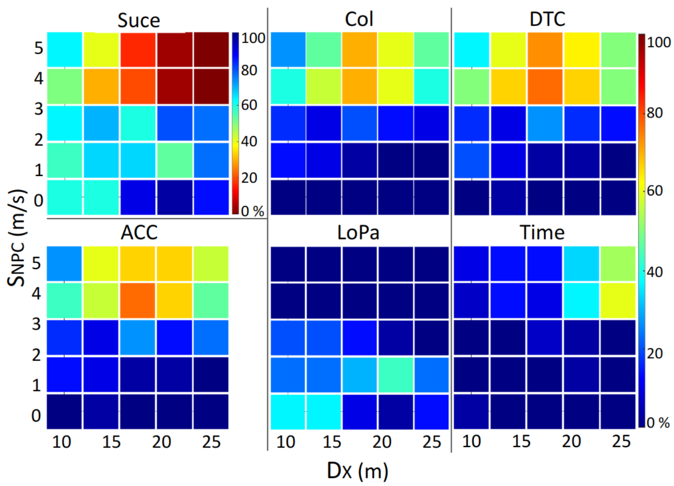

Figure 3.23: Simulation result with optimal planner parameters repeated #30 (marked as LF(a)). Different metrics evaluations are shown separately.

By comparing the "Suce" metric of the optimized set with that of the non-optimized (LF#30) sample, it is evident that the optimized setup was more successful in completing the mission safely. It was observed that more collisions, DTCs, and ACC violations occurred at speeds higher than 2.5 m/s, indicating that the DUT is capable of overtaking fast-driving NPCs at speeds closer to its own. In spite of this, the rate of DTC and ACC violations has been reduced across the entire scenario domain. While long passing has remained the same with a small increase, time violations have dramatically decreased, indicating more missions have been completed.

High-Fidelity (HF) Simulations:

In the same scenario domain, high-fidelity simulations were conducted to observe performance compared to LF simulations. In this setup, DUT software modules including localization, detection, and planning were completely in the simulation loop. Three sets of HF simulations were conducted. In the first set of simulations, the default planning parameters (HF30) are used. In the second set, previously optimized parameters derived from LF simulations are used (HF(a)), and in the third set, optimized parameters are used from HF simulations (HF(b)).

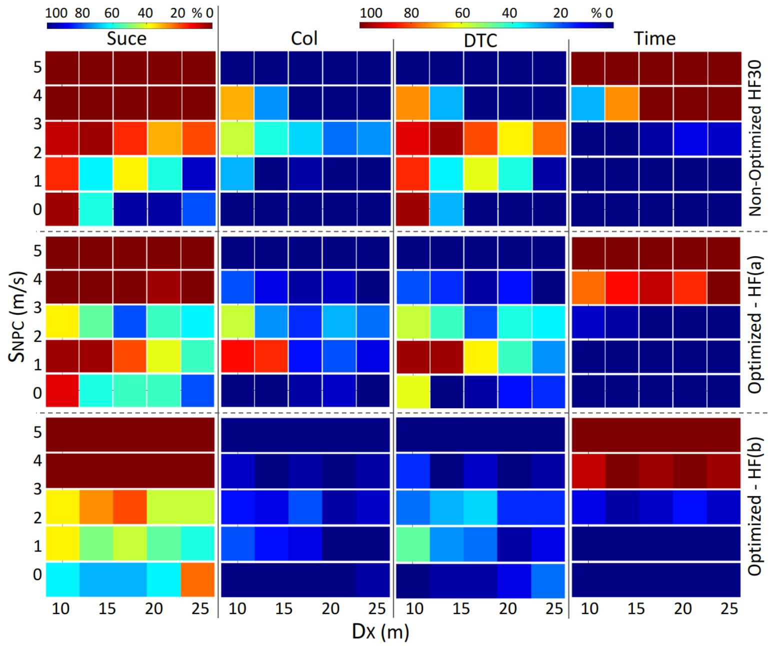

Figure 3.24: HF simulation result for three different sets; Non-Optimized HF30 is the simulations with default planning parameters, Optimized-HF(a) is the simulations with planning parameters optimized by LF simulations, Optimized-HF(b) is the simulations with optimized parameters suggested by HF platform.

Figure 3.24 presents the metric evaluation results for the mentioned sets. According to the figure, optimization efforts increased success rates in the scenario domain (a and b). The comparison of the two optimized cases indicates that the parameters optimized with the lower-fidelity platform are not optimal for the higher-fidelity case, which is more realistic. The results shown in the last row of the figure are more promising when optimized using the HF platform. Another interesting observation is the occurrence of collisions at low speeds for NPCs in the optimized case HF(a). There were almost full collision rates for scenarios [10, 1.25] and [13.75, 1.25] inside the domain.

The crash plots of the scenes revealed that the optimized prediction and trajectory evaluator did not take into account the moving NPC when performing the safe cut-in maneuver. It is important to note that the LF simulation's detection module considers the NPC as a whole cube, which is why the planner always has a full picture of it. Inside the HF, however, detection relies on LiDAR sensors that detect the NPC's face when it is exposed to laser light. As a result, the size of the objects is constantly changing and occasionally disappears, resulting in a loss of the object in the prediction module (see Figure 3.25).

Thus, the optimized case (a) using the LF optimal parameters could not demonstrate satisfactory results when simulated in HF.

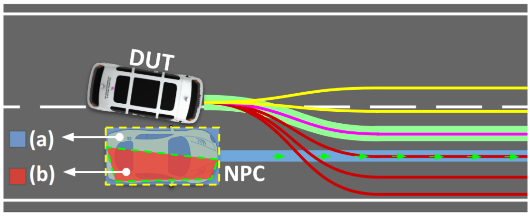

Figure 3.25: Sample plot of a crash in simulation scenarios. It shows (a) how NPC was seen in LF simulations and (b) how it is seen in HF simulations.

Table 3.8 provides details for 15000 simulations conducted in this study, including the number of simulation runs, the duration of each set, and the performance of the DUT within the overall scenario domain described by the three main criteria. The table contains LF with different repetitions, optimization trials with low- and high-fidelity platforms (gaLF and gaHF), LF simulation with optimum parameters (LF(a)), HF simulations in not optimized cases (HF30), and optimized cases (HF(a) and (b)).

Table 3.8: Summary of all simulation tries

	LF10	LF20	LF30	LF50	gaLF	LF(a)	HF30	gaHF	HF(a)	HF(b)
#Runs	250	500	750	1250	8750	750	750	455	750	750
Time(h:m)	1:37	3:25	5:00	8:30	50	4:15	22:30	14	23:00	24:00
Suce	30%	27%	29%	28%	-	52%	27%	-	27%	28%
Col	25%	20%	22%	21%	-	24%	12%	-	17%	4%
DTC	61%	60%	59%	60%	-	29%	34%	-	27%	12%

Based on the information, HF simulations took almost 4.5 times longer than LF simulations (comparing LF30 with HF30). It is particularly important to consider this when optimizing. The gaLF case is an example of the optimization process with the LF platform involving 1700 attempts and 8750 simulations over a period of 50 hours. In contrast, 91 simulation attempts comprising 455 simulations took 14 hours using the HF platform (see gaHF case). With the optimal values suggested by these attempts, we ran three simulation sets including LF(a), HF(a), and HF(b). The two (a) cases utilized gaLF optimum values while case (b) utilized gaHF optimal values. As can be seen from the violation rates, although gaLF optimization improved the safety condition in LF(a) (comparing LF30 with LF(a)), it could not do the same in HF(a) (comparison of HF30 with HF(a)). HF(b), which inherits the optimum value from the gaHF attempts, provides superior safety performance (compare HF30 with HF(b)). As a result, optimization based on LF platforms may not be appropriate for evaluating a unit (e.g. planner) within a system (e.g. autonomous software). However, they provide better time performance and are ideal for separate unit testing.

3.5 Real-World Experiments

TalTech iseAuto has been utilized in a number of real-life experiments to evaluate and examine the AD algorithm's performance and behavior in certain situations. Furthermore, it is used to investigate the results derived from the simulation test bed (refer to **RO5**).

However, as it has been noted, track testing is a time-consuming and labor-intensive process. Following is a description of some real-life experiments that are conducted as part of this research in order to justify the findings.

3.5.1 Safety and Performance Evaluation Tests

Figure 3.26: A Real-world experiment on iseAuto autonomous shuttle; AV shuttle behaviour evaluated during a passing maneuver

These real-world experiments were conducted in order to validate the shuttle's behavior in certain planned maneuvers. In the TalTech track testing area, scenarios including pedestrian and traffic light interactions, object passing, and overtaking missions are tested. The tests have enabled the shuttle to identify and interpret objects and other actors detected by its sensors. Figure 3.26 shows drone images taken during an NPC passing mission from the campus testbed. There is also a visual representation of perception and planning algorithms in Rviz software in addition to real-world pictures. The mission ended successfully, with no collisions or other safety issues. The results of this test suggest that the perception and planning algorithms are reliable.

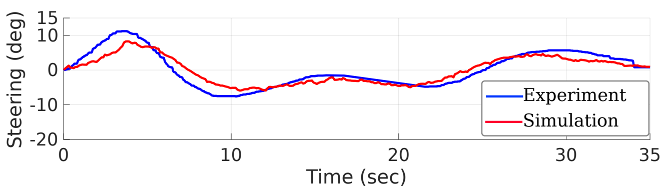

Figure 3.27: Steering angle output from the TalTech iseAuto in an overtaking maneuver conducted in a real-life experiment and a simulation

A comparison was also conducted between the steering output of the DUT in this experiment and that derived from the corresponding simulation to validate the simulation results (see Figure 3.27). Even though there were not enough tests to come to a definitive conclusion, the comparison reflects a reasonable match between the two results. Overall, the results are promising and indicate that the virtual environment is sufficiently accurate to simulate real models. Further real-life experiments are needed to evaluate the simulation results.

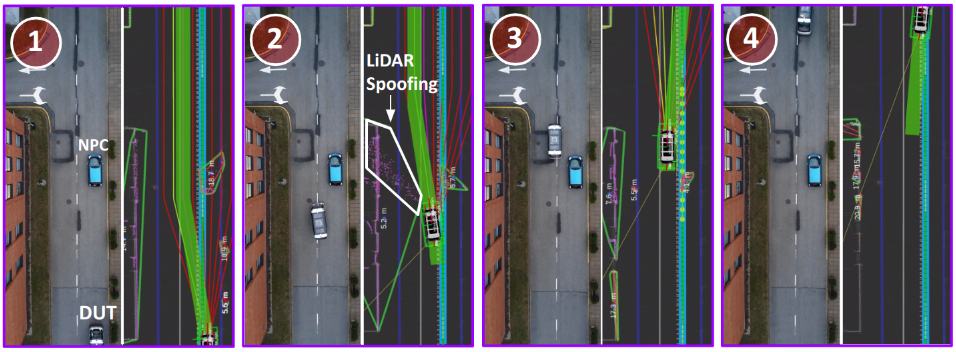

Figure 3.28: A Real-world experiment on iseAuto autonomous shuttle; The DUT experienced a LiDAR spoofing attack while overtaking an NPC

3.5.2 Cyber Security Evaluation Tests

The TalTech testbed also makes it possible for cyber-physical attacks to be conducted on the same testing track as safety experiments. In this way, it is possible to analyze the impact of cyberattacks and the effectiveness of existing countermeasures, as well as the impact of proposed security solutions on the safety of AV software. The implementation of attack scenarios and the identification of potential vulnerabilities are also possible. Finally, field tests can be used to verify the accuracy of simulation results. As part of the Paper IV study, a LiDAR FPs injection attack was performed on the shuttle during a passing maneuver. Figure 3.28 shows the frames captured during the experiments. Spoof points can be seen inside the second frame in the set of Figure 3.28. In the field tests, however, simulation of failed cyber scenarios was not practical due to safety concerns. Therefore, real experiments are limited to successful scenarios. In spite of this, the implementation of the cyber attack on a real AV with real sensors led to the discovery of new parameters and value ranges that made the attack more effective. Detailed explanations of these parameters are given in Paper IV.

As a whole, real-life track testing is a very useful tool for crediting simulation findings' reliability and assessing AV reliability and safety. Before the AV is deployed in the real world, it can be used to identify and address safety issues that cannot be detected in simulations. However, these benefits come with costs, such as the time and resources required for track testing. Additionally, track testing can be dangerous if not done correctly, putting both the AV and the human testers at risk.

4 Discussion

In recent years, scenario-based evaluation methods using simulation techniques have gained significant attention in the fields of safety, security, and performance analysis of AVs. One prominent approach involves utilizing SiL and HiL simulations with varying levels of fidelity. This discussion will delve into the methodology's benefits, highlight its limitations, and explore potential avenues for future research and development.

Scenario-based evaluation using SiL/HiL simulation offers numerous advantages in assessing AV system safety, security, and performance. By creating virtual environments that mimic real-world conditions, this method allows for comprehensive testing without incurring the risks and costs associated with physical prototypes. Furthermore, it enables the evaluation of critical scenarios that may be challenging or impossible to replicate in real life, such as extreme weather conditions or rare security breaches.

4.1 Low-Fidelity Simulation

Low-fidelity SiL simulations serve as a starting point in scenario-based evaluation. They provide a cost- and time-effective means of analyzing system behavior under various scenarios. In Paper I, a study based on the MATLAB low-fidelity simulator was conducted. The results suggested the potential of low-fidelity SiL simulations as a tool for validating AD software. This motivated us to further explore the use of SiL simulations in a variety of scenarios including overtaking and cyber security attacks (see Paper V).

However, due to the low-fidelity simplified representation of real-world dynamics, these simulations may fail to capture the nuances and complexities of complex systems accurately. Limitations of low-fidelity SiL simulations include:

Lack of Realism: Low-fidelity AV simulations often simplify the underlying physics, environment, and system interactions, potentially leading to inaccurate results. For instance, a lack of advanced sensors and poor dynamic models can contribute to inaccurate predictions of AV performance under complex conditions. This can also lead to a false sense of safety when evaluating an AV system's safety.

Limited Scope: Low-fidelity simulations may not consider all relevant system components, leading to incomplete evaluation. This can hinder the identification of potential safety or security vulnerabilities from interdependent subsystems. It is discussed in detail in Paper V.

4.2 High-Fidelity SiL/HiL Simulation

Addressing the limitations of low-fidelity SiL simulations, high-fidelity SiL/HiL simulations offer a more accurate representation of AV system behavior. By incorporating actual software/hardware components in addition to providing a realistic virtual environment that simulates advanced sensor data, these simulations provide a comprehensive evaluation of system performance, safety, and security. However, they come with their own set of challenges and limitations:

Cost and Complexity: Implementing SiL/HiL simulations requires significant investments in computational hardware resources, interfaces, and software tools. The complexity of integrating real hardware components with simulation models can also pose challenges, requiring specialized expertise and resources.

Scalability: As system complexity increases, it becomes more challenging to maintain real-time performance in SiL/HiL simulations. The computational demands of executing large-scale scenarios with numerous interacting components may strain existing hardware resources. A timing comparison between low- and high-fidelity simulations in Paper V shows that higher-detail simulations take longer to execute and may raise time synchronizing issues.

4.3 Limitations and Future Work

While scenario-based evaluation using SiL/HiL simulations offers substantial benefits, there are several areas that require further attention:

Model Fidelity Improvement: Enhancing simulation models' fidelity is crucial to bridge the gap between simulated and real-world behavior. Advancements in modeling techniques, such as multi-physics simulations, real-time data integration, and increased accuracy in cyber-physical interactions, will improve simulation accuracy and reliability.

Realism in Virtual Environments: Developing realistic virtual environments that accurately represent complex real-world scenarios is a key research direction. This includes incorporating dynamic weather patterns, traffic simulations, sensor models, and realistic human behavior to create a more comprehensive evaluation environment. A preliminary attempt is presented to create a realistic digital twin of the AV working environment in Paper II.

Future Work: Future research can concentrate on improving the fidelity of simulation models. This may involve integrating more accurate and detailed models of subsystems, capturing complex dynamics, and incorporating real-time data from the actual system to enhance realism. Leveraging machine learning and data-driven techniques can also help in automatically calibrating and updating simulation models based on real-world observations.

Uncertainty and Variability: Real-world autonomous systems often face various uncertainties and variations, including environmental conditions, human factors, and component failures. Capturing and incorporating these uncertainties and variations in the simulation environment can be challenging.

Future Work: Developing techniques to effectively model and simulate uncertainties and variations is an important area for future work. This may involve stochastic modeling, probabilistic analysis, and incorporating probabilistic models for environmental conditions, human behavior, and component reliability. These advancements would enable more robust evaluations that account for real-world variability.

Cyber security Assessment: As systems become increasingly connected and reliant on data exchange, evaluating security aspects of scenarios is of paramount importance. The cyberattack effectiveness on the AV shuttle has been investigated in two different studies using the proposed evaluation platform (see Paper IV and [123]).

Future Work: Future work should focus on integrating comprehensive cyber security evaluation frameworks into scenario-based simulations to identify vulnerabilities and potential threats.

Integration with Physical Testing: SiL/HiL simulations primarily focus on virtual testing, and there can be limitations in replicating all aspects of the physical system accurately. The integration of virtual simulations with physical testing is crucial to validate the simulation results and ensure their reliability.

Future Work: Future research should explore methods to seamlessly integrate SiL/HiL simulations with physical testing. This can involve developing hybrid testing approaches that combine virtual simulations and physical testbeds, enabling more comprehensive evaluations and verification of the system's performance.

System Adaptation and Learning: Traditional SiL/HiL simulations often assume a static system model and do not account for adaptive and learning capabilities. However, many modern AD systems incorporate adaptive algorithms, machine learning, and artificial intelligence techniques, which necessitate evaluating the system's performance under dynamic conditions.

Future Work: Future work can focus on extending SiL/HiL simulation approaches to accommodate adaptive and learning systems. This may involve incorporating adaptive models, reinforcement learning algorithms, and online adaptation mechanisms into the simulation framework to evaluate the system's performance in dynamic environments.

Scalable Simulation Platforms: One limitation is the challenge of scaling up the simulations to larger and more complex systems. As the size and complexity of the system increase, the computational requirements and simulation runtime also escalate. This can hinder the feasibility of conducting comprehensive evaluations for large-scale systems.

Future Work: Research efforts can focus on developing efficient algorithms and techniques to handle scalability challenges. This may involve exploring parallel computing, distributed simulation frameworks, or model abstraction techniques to reduce computational burden while maintaining sufficient accuracy.

By addressing these limitations and pursuing future research, scenario-based safety, security, and performance evaluation using SiL/HiL simulations can continue to evolve, providing more accurate, comprehensive, and reliable assessments of AV systems in diverse ODDs.

5 Conclusion and future research

In this study, according to its primary research objective (RO1), a scenario-based evaluation toolkit using SiL/HiL simulations in low- and high-fidelity setups was developed. Results showed that the proposed toolkit is a valuable and systematic methodology for evaluating AD systems' safety, security, and performance. This approach can be used to perform low-fidelity and high-fidelity simulations, depending on the level of testing and the requirements (as sketched in RO3).

It has been observed that, by creating simplified virtual environments, engineers can conduct early-stage assessments of vehicle performance and behavior without costly physical prototypes or real-world testing. These simulations aim to replicate various driving scenarios, including different road conditions, traffic patterns, and unexpected events. They aim to assess the vehicle's ability to make informed decisions and react appropriately. While low-fidelity simulations may lack some real-world intricacies, they provide a valuable initial platform for identifying potential flaws, refining algorithms, and optimizing vehicle responses. This cost-effective and time-efficient approach enables researchers and developers to iterate and enhance autonomous systems, ultimately advancing the safe and reliable integration of self-driving cars into our transportation ecosystem. RO2 is addressed by integrating low-fidelity simulation into the proposed toolkit.

Alternatively, high-fidelity simulations, due to providing a more realistic environment, can be employed to detect, analyze, and address system-level issues and investigate components' interactions. Further, they can be used for the development, testing, and evaluation of control algorithms and logic. They can analyze the system's performance, identify potential failure points, and generate system-level recommendations. Furthermore, this approach provides a controlled and repeatable way to evaluate the safety, security, and performance of the AV by creating a virtual environment that mimics real-world conditions. Various advanced cyber threats can be evaluated without posing any risk to the system which is aligned with RO4. As part of RO5, real-life tests were also conducted to verify simulation fidelity results. As a result, high-fidelity simulations can reduce the development time and cost of autonomous driving systems, making them more accessible to a wider audience.

This evaluation toolkit has several advantages including the ability to test systems in a safe environment, identify weak points and areas for improvement, and provide quantitative performance assessments. Businesses can use this toolkit to provide evaluation services to other institutions and industries. In addition, standard organizations can benefit from benchmarking their systems against those of other institutes. It is imperative, however, to acknowledge the limitations of this approach. These limitations include the challenge of accurately representing real-world conditions and the potential for incomplete capture of complex subsystem interactions.

Overall, the proposed toolkit offers significant benefits in assessing AV system safety, security, and performance. Taking into account the limitations and conducting further research (as mentioned in the discussion), this evaluation toolkit has the potential to contribute to the development of safer, more secure, and higher-performing AVs.

5.1 Future Work

To overcome the aforementioned limitations and further enhance the effectiveness of this evaluation method, the following future works are suggested:

- Improving the accuracy and reliability of simulation models including the digital twin of the environments and road users.

- Developing more advanced virtual sensors, including ray-cast LiDARs.
- Improving the simulation clock synchronization regarding utilizing high-fidelity simulators.
- Research efforts should also be directed towards developing more sophisticated simulations able to capture more subsystem interaction details.
- The establishment of standardized evaluation protocols would also promote consistency and comparability of results across different systems and industries.

References

[1] Jorge Vargas, Suleiman Alsweiss, Onur Toker, Rahul Razdan, and Joshua Santos. An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(16):5397, 2021.

[2] Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[3] David Shinar. *Traffic safety and human behavior*. Emerald Group Publishing, 2017.

[4] National Highway Traffic Safety Administration et al. National motor vehicle crash causation survey: Report to congress. *National Highway Traffic Safety Administration Technical Report DOT HS*, 811:059, 2008.

[5] Donald L Hendricks, Mark Freedman, James C Fell, et al. The relative frequency of unsafe driving acts in serious traffic crashes. Technical report, United States. National Highway Traffic Safety Administration, 2001.

[6] Dietmar Otte, Bernd Pund, and Michael Jänsch. A new approach of accident causation analysis by seven steps acass. In *ESV-Conference, Paper*, 2009.

[7] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, NHTSA's National Center for Statistics and Analysis, 2018.

[8] Ian Y Noy, David Shinar, and William J Horrey. Automated driving: Safety blind spots. *Safety science*, 102:68–78, 2018.

[9] Nirajan Shiwakoti, Peter Stasinopoulos, and Francesco Fedele. Investigating the state of connected and autonomous vehicles: A literature review. *Transportation Research Procedia*, 48:870–882, 2020.

[10] Yifang Ma, Zhenyu Wang, Hong Yang, and Lin Yang. Artificial intelligence applications in the development of autonomous vehicles: A survey. *IEEE/CAA Journal of Automatica Sinica*, 7(2):315–329, 2020.

[11] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

[12] Calin Iclodean, Nicolae Cordos, and Bogdan Ovidiu Varga. Autonomous shuttle bus for public transportation: A review. *Energies*, 13(11):2917, 2020.

[13] Carolin Altena and Hannes Thees. Planning and implementation of autonomous shuttle buses in tourism mobility in the region berchtesgaden-königssee, germany. *ROBONOMICS: The Journal of the Automated Economy*, 3:28–28, 2022.

[14] Manuel Trotta, Claudia Archetti, Dominique Feillet, and Alain Quilliot. Pickup and delivery problems with autonomous vehicles on rings. *European Journal of Operational Research*, 300(1):221–236, 2022.

[15] Sharon L Poczter, Luka M Jankovic, et al. The google car: driving toward a better future? *Journal of Business Case Studies (JBCS)*, 10(1):7–14, 2014.

[16] Twain Taylor. Top 8 self-driving startups to watch in 2018 and 2019. https://techgenix.com/self-driving-startups/. Accessed: 2023-4-18.

[17] Easymile, autonomous vehicle technology and solutions. https://easymile.com/. Accessed: 2023-4-18.

[18] Navya, self-driving made real. http://www.navya.tech/en/. Accessed: 2023-4-18.

[19] Johannes Deichmann, Georg Doll, Jan Paul Stein, Benjamin Klein, and Bernhard Mühlreiter. Cracking the complexity code in embedded systems development. https://www.mckinsey.com/industries/industrials-and-electronics/our-insights/cracking-the-complexity-code-in-embedded-systems-development. Accessed: 2023-4-18.

[20] Jan Erik Stellet, Matthias Woehrle, Tino Brade, Alexander Poddey, and Wolfgang Branz. Validation of automated driving a structured analysis and survey of approaches. In *13. Uni-DAS eV Workshop Fahrerassistenz und automatisiertes Fahren*, page 10, 2020.

[21] J SAE. 3016-2021, taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *Society of Automobile Engineers, sae. org*, 2021.

[22] Andrea Leitner, Daniel Watzenig, and Javier Ibanez-Guzman. *Validation and verification of automated systems*. Springer, 2019.

[23] Adina Aniculaesei, Jörg Grieser, Andreas Rausch, Karina Rehfeldt, and Tim Warnecke. Toward a holistic software systems engineering approach for dependable autonomous systems. In *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, pages 23–30, 2018.

[24] Nijat Rajabli, Francesco Flammini, Roberto Nardone, and Valeria Vittorini. Software verification and validation of safe autonomous cars: A systematic literature review. *IEEE Access*, 9:4797–4819, 2020.

[25] F Cechini, R Ice, and D Binkley. Systems engineering guidebook for intelligent transportation systems. *California Division of the United States Department of Transportation Federal Highway Administration and the California Department of Transportation*, 2009.

[26] Howard Cotterman, Kevin Forsberg, and Hal Mooz. *Visualizing project management: models and frameworks for mastering complex systems*. John Wiley & Sons, 2005.

[27] ISO 26262. Road Vehicles—Functional Safety. Standard, International Organization for Standardization, 2011.

[28] National instruments, what is the iso 26262 functional safety standard? http://www.ni.com/white-paper/13647/en/. Accessed: 2023-4-18.

[29] Jeffrey Wishart, Steve Como, Uilani Forgione, Jack Weast, Lewis Weston, Andrew Smart, and George Nicols. Literature review of verification and validation activities of automated driving systems. *SAE Int. J. Connect. Autom. Veh*, 3:267–323, 2020.

[30] Vehicle Cybersecurity Systems Engineering Committee. *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*, dec 2021.

[31] Georg Macher, Christoph Schmittner, Omar Veledar, and Eugen Brenner. Iso/sae dis 21434 automotive cybersecurity standard-in a nutshell. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*, pages 123–135. Springer, 2020.

[32] National Highway Traffic Safety Administration et al. Quick reference guide to federal motor vehicle safety standards and regulations. *DOT HS*, 805:878, 2004.

[33] Philip Koopman, Uma Ferrell, Frank Fratrik, and Michael Wagner. A safety standard approach for fully autonomous vehicles. In *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings 38*, pages 326–332. Springer, 2019.

[34] NHTSA Federal Automated Vehicles Policy—September. Us department of transportation. *Washington, DC, USA*, 2016.

[35] USDOT NSTC. Ensuring american leadership in automated vehicle technologies: Automated vehicles 4.0. *NSTC, USDOT: Washington, DC, USA*, 2020.

[36] Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):748, 2021.

[37] Li Li, Yi-Lun Lin, Nan-Ning Zheng, Fei-Yue Wang, Yuehu Liu, Dongpu Cao, Kunfeng Wang, and Wu-Ling Huang. Artificial intelligence test: A case study of intelligent vehicles. *Artificial Intelligence Review*, 50:441–465, 2018.

[38] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[39] Prabhjot Kaur, Samira Taghavi, Zhaofeng Tian, and Weisong Shi. A survey on simulators for testing self-driving cars. In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, pages 62–70. IEEE, 2021.

[40] Shuo Feng, Yiheng Feng, Chunhui Yu, Yi Zhang, and Henry X Liu. Testing scenario library generation for connected and automated vehicles, part i: Methodology. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1573–1582, 2020.

[41] Felix Beringhoff. Towards realizing test conditions for automated vehicles. In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pages 152–157, 2022.

[42] PEGASUS Consortium. Pegasus method - an overview. Technical report, PEGASUS Project, 2019.

[43] Christian Neurohr, Lukas Westhofen, Tabea Henning, Thies de Graaff, Eike Möhlmann, and Eckard Böde. Fundamental considerations around scenario-based testing for automated driving. In *2020 IEEE intelligent vehicles symposium (IV)*, pages 121–127. IEEE, 2020.

[44] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles. *IEEE access*, 8:87456–87477, 2020.

[45] Christian Roesener, Felix Fahrenkrog, Axel Uhlig, and Lutz Eckstein. A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pages 1360–1365. IEEE, 2016.

[46] Xinhai Zhang, Jianbo Tao, Kaige Tan, Martin Törngren, Jose Manuel Gaspar Sanchez, Muhammad Rusyadi Ramli, Xin Tao, Magnus Gyllenhammar, Franz Wotawa, Naveen Mohan, et al. Finding critical scenarios for automated driving systems: A systematic mapping study. *IEEE Transactions on Software Engineering*, 49(3):991–1026, 2022.

[47] Thomas Ponn, Dirk Fratzke, Christian Gnandt, and Markus Lienkamp. Towards certification of autonomous driving: Systematic test case generation for a comprehensive but economically-feasible assessment of lane keeping assist algorithms. In *VEHITS*, pages 333–342, 2019.

[48] Saritha Route. Test optimization using combinatorial test design: Real-world experience in deployment of combinatorial testing at scale. In *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 278–279. IEEE, 2017.

[49] Elias Rocklage, Heiko Kraft, Abdullah Karatas, and Jörg Seewig. Automated scenario generation for regression testing of autonomous vehicles. In *2017 ieee 20th international conference on intelligent transportation systems (itsc)*, pages 476–483. IEEE, 2017.

[50] István Majzik, Oszkár Semeráth, Csaba Hajdu, Kristóf Marussy, Zoltán Szatmári, Zoltán Micskei, András Vörös, Aren A Babikian, and Dániel Varró. Towards system-level testing with coverage guarantees for autonomous vehicles. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pages 89–94. IEEE, 2019.

[51] Siddartha Khastgir, Gunwant Dhadyalla, Stewart Birrell, Sean Redmond, Ross Addinall, and Paul Jennings. Test scenario generation for driving simulators using constrained randomization technique. Technical report, SAE Technical Paper, 2017.

[52] Li Huang, Qin Xia, Fei Xie, Hai-Lin Xiu, and Hong Shu. Study on the test scenarios of level 2 automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 49–54. IEEE, 2018.

[53] Halil Beglerovic, Abhishek Ravi, Niklas Wikström, H-M Koegeler, Andrea Leitner, and Jürgen Holzinger. Model-based safety validation of the automated driving functio highway pilot. In *8th International Munich Chassis Symposium 2017: chassis. tech plus*, pages 309–329. Springer, 2017.

[54] Daniel Åsljung, Jonas Nilsson, and Jonas Fredriksson. Comparing collision threat measures for verification of autonomous vehicles using extreme value theory. *IFAC-PapersOnLine*, 49(15):57–62, 2016.

[55] Ding Zhao, Henry Lam, Huei Peng, Shan Bao, David J LeBlanc, Kazutoshi Nobukawa, and Christopher S Pan. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE transactions on intelligent transportation systems*, 18(3):595–607, 2016.

[56] Zhiyuan Huang, Henry Lam, David J LeBlanc, and Ding Zhao. Accelerated evaluation of automated vehicles using piecewise mixture models. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2845–2855, 2017.

[57] Lukas Stark, Michael Düring, Stefan Schoenawa, Jan Enno Maschke, and Cuong Manh Do. Quantifying vision zero: Crash avoidance in rural and motorway accident scenarios by combination of acc, aeb, and lks projected to german accident occurrence. *Traffic injury prevention*, 20(sup1):S126–S132, 2019.

[58] Lukas Stark, Simone Obst, Stefan Schoenawa, and Michael Düring. Towards vision zero: Addressing white spots by accident data based adas design and evaluation. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–6. IEEE, 2019.

[59] Philip Feig, V Labenski, T Leonhardt, and J Schatz. Assessment of technical requirements for level 3 and beyond automated driving systems based on naturalistic driving and accident data analysis. In *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019.

[60] Marc Nabhan. *Models and Algorithms for the Exploration of the Space of Scenarios: Toward the Validation of the Autonomous Vehicle*. PhD thesis, Université Paris-Saclay, 2020.

[61] Matthias Althoff and Sebastian Lutz. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1326–1333. IEEE, 2018.

[62] Alyssa Pierson, Wilko Schwarting, Sertac Karaman, and Daniela Rus. Learning risk level set parameters from data sets for safer driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 273–280. IEEE, 2019.

[63] Moritz Klischat and Matthias Althoff. Generating critical test scenarios for automated vehicles with evolutionary algorithms. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2352–2358. IEEE, 2019.

[64] Qin Xia, Jianli Duan, Feng Gao, Tao Chen, and Cai Yang. Automatic generation method of test scenario for adas based on complexity. Technical report, SAE Technical Paper, 2017.

[65] Qin Xia, Jianli Duan, Feng Gao, Qiuxia Hu, and Yingdong He. Test scenario design for intelligent driving system ensuring coverage and effectiveness. *International Journal of Automotive Technology*, 19:751–758, 2018.

[66] Feng Gao, Jianli Duan, Yingdong He, Zilong Wang, et al. A test scenario automatic generation strategy for intelligent driving systems. *Mathematical Problems in Engineering*, 2019, 2019.

[67] Mark Koren, Saud Alsaif, Ritchie Lee, and Mykel J Kochenderfer. Adaptive stress testing for autonomous vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE, 2018.

[68] Halil Beglerovic, Michael Stolz, and Martin Horn. Testing of autonomous vehicles using surrogate models and stochastic optimization. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[69] Briti Gangopadhyay, Siddartha Khastgir, Sumanta Dey, Pallab Dasgupta, Giovanni Montana, and Paul Jennings. Identification of test cases for automated driving systems using bayesian optimization. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1961–1967. IEEE, 2019.

[70] Houssam Abbas, Matthew O'Kelly, Alena Rodionova, and Rahul Mangharam. Safe at any speed: A simulation-based test harness for autonomous vehicles. In *Cyber Physical Systems. Design, Modeling, and Evaluation: 7th International Workshop, CyPhy 2017, Seoul, South Korea, October 15-20, 2017, Revised Selected Papers 7*, pages 94–106. Springer, 2019.

[71] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.

[72] Guannan Lou, Yao Deng, Xi Zheng, Mengshi Zhang, and Tianyi Zhang. Testing of autonomous driving systems: where are we and where should we go? In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 31–43, 2022.

[73] Changwen Li, Joseph Sifakis, Qiang Wang, Rongjie Yan, and Jian Zhang. Simulation-based validation for autonomous driving systems. *arXiv preprint arXiv:2301.03941*, 2023.

[74] Daniel J Fremont, Edward Kim, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L Sangiovanni-Vincentelli, and Sanjit A Seshia. Scenic: A language for scenario specification and data generation. *Machine Learning*, pages 1–45, 2022.

[75] Rupak Majumdar, Aman Mathur, Marcus Pirron, Laura Stegner, and Damien Zufferey. Paracosm: A test framework for autonomous driving simulations. In *Fundamental Approaches to Software Engineering: 24th International Conference, FASE 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27–April 1, 2021, Proceedings*, pages 172–195. Springer International Publishing Cham, 2021.

[76] Haoxiang Tian, Yan Jiang, Guoquan Wu, Jiren Yan, Jun Wei, Wei Chen, Shuo Li, and Dan Ye. Mosat: finding safety violations of autonomous driving systems using multi-objective genetic algorithm. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 94–106, 2022.

[77] Guanpeng Li, Yiran Li, Saurabh Jha, Timothy Tsai, Michael Sullivan, Siva Kumar Sastry Hari, Zbigniew Kalbarczyk, and Ravishankar Iyer. Av-fuzzer: Finding safety violations in autonomous driving systems. In *2020 IEEE 31st international symposium on software reliability engineering (ISSRE)*, pages 25–36. IEEE, 2020.

[78] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. *arXiv preprint arXiv:2210.17366*, 2022.

[79] Moritz Klischat, Edmond Irani Liu, Fabian Holtke, and Matthias Althoff. Scenario factory: Creating safety-critical traffic scenarios for automated vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020.

[80] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[81] Yun Tang, Yuan Zhou, Tianwei Zhang, Fenghua Wu, Yang Liu, and Gang Wang. Systematic testing of autonomous driving systems using map topology-based scenario classification. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1342–1346. IEEE, 2021.

[82] Martin Butz, Christian Heinzemann, Martin Herrmann, Jens Oehlerking, Michael Rittel, Nadja Schalm, and Dirk Ziegenbein. Soca: Domain analysis for highly automated driving systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[83] Marius Bozga and Joseph Sifakis. Specification and validation of autonomous driving systems: A multilevel semantic framework. In *Principles of Systems Design: Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday*, pages 85–106. Springer, 2022.

[84] Klemens Esterle, Luis Gressenbuch, and Alois Knoll. Formalizing traffic rules for machine interpretability. In *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, pages 1–7. IEEE, 2020.

[85] Luis Gressenbuch and Matthias Althoff. Predictive monitoring of traffic rules. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 915–922. IEEE, 2021.

[86] Changwen Li, Chih-Hong Cheng, Tiantian Sun, Yuhang Chen, and Rongjie Yan. Comopt: Combination and optimization for testing autonomous driving systems. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7738–7744. IEEE, 2022.

[87] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pages 132–142, 2018.

[88] Zhi Quan Zhou and Liqun Sun. Metamorphic testing of driverless cars. *Communications of the ACM*, 62(3):61–67, 2019.

[89] Till Menzel, Gerrit Bagschik, and Markus Maurer. Scenarios for development, test and validation of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1821–1827. IEEE, 2018.

[90] John M Scanlon, Kristofer D Kusano, Tom Daniel, Christopher Alderson, Alexander Ogle, and Trent Victor. Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. *Accident Analysis & Prevention*, 163:106454, 2021.

[91] NVIDIA. NVIDIA DRIVE Sim powered by omniverse. https://www.nvidia.com/en-us/self-driving-cars/simulation/. Accessed: 2023-4-18.

[92] Autoware Foundation. TIER IV AWSIM. https://github.com/tier4/AWSIM, 2022.

[93] Jaeyoung Lee, Aravind Balakrishnan, Ashish Gaurav, Krzysztof Czarnecki, and Sean Sedwards. W ise m ove: a framework to investigate safe deep reinforcement learning for autonomous driving. In *Quantitative Evaluation of Systems: 16th International Conference, QEST 2019, Glasgow, UK, September 10–12, 2019, Proceedings 16*, pages 350–354. Springer, 2019.

[94] Aravind Balakrishnan. *Closing the Modelling Gap: Transfer Learning from a Low-Fidelity Simulator for Autonomous Driving*. Master's thesis, Computer Science University of Waterloo, Waterloo, Canada, 2019.

[95] Marko Ilievski. *WiseBench: A Motion Planning Benchmarking Framework for Autonomous Vehicles*. Master's thesis, Computer Science University of Waterloo, Waterloo, Canada, 2020.

[96] MATLAB. Automated Driving Toolbox. https://www.mathworks.com/products/automated-driving.html. Accessed: 2023-4-18.

[97] MATLAB. RoadRunner. https://www.mathworks.com/products/roadrunner.html. Accessed: 2023-4-18.

[98] Eclipse Foundation. Eclipse SUMO Simulator. https://www.eclipse.org/sumo/. Accessed: 2023-4-18.

[99] CommonRoad. TUM CommonRoad. https://commonroad.in.tum.de/. Accessed: 2023-4-18.

[100] Matthias Althoff, Markus Koschi, and Stefanie Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726. IEEE, 2017.

[101] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE, 2018.

[102] Hatem Darweesh, Eijiro Takeuchi, and Kazuya Takeda. Openplanner 2.0: The portable open source planner for autonomous driving applications. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pages 313–318. IEEE, 2021.

[103] Autoware Foundation. Planning Simulation. https://autowarefoundation.github.io/autoware-documentation/main/tutorials/ad-hoc-simulation/planning-simulation/. Accessed: 2023-4-18.

[104] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2020.

[105] Baidu Apollo team (2017), Apollo: Open Source Autonomous Driving. https://github.com/ApolloAuto/apollo. Accessed: 2023-4-18.

[106] John K Haas. A history of the unity game engine. *Diss. Worcester Polytechnic Institute*, 483(2014):484, 2014.

[107] Adam Dąbrowski. Open-source release of ROS2 for Unity. https://robotec.ai/open-source-release-of-ros2-for-unity/. Accessed: 2023-4-18.

[108] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[109] Carla Simulator: Open-source simulator for autonomous driving research. https://carla.org/. Accessed: 2023-4-18.

[110] Andrew Sanders. *An introduction to Unreal engine 4*. CRC Press, 2016.

[111] A. Rassõlkin, L. Gevorkov, T. Vaimann, A. Kallaste, and R. Sell. Calculation of the Traction Effort of ISEAUTO Self-Driving Vehicle. In *IEEE International Workshop on Electric Drives: Optimization in Control of Electric Drives*, 2018.

[112] Ruxin Wang, Raivo Sell, Anton Rassõlkin, Tauno Otto, and Ehsan Malayjerdi. Intelligent functions development on autonomous electric vehicle platform. *Journal of Machine Engineering*, 20(2), 2020.

[113] A. Rassõlkin, T. Vaimann, A. Kallaste, and R. Sell. Propulsion motor drive topology selection for further development of ISEAUTO self-driving car. In *IEEE 59th Annual International Scientific Conference on Power and Electrical Engineering*, 2018.

[114] Ehsan Malayjerdi. *Advanced Autonomous Vehicle's Functions for Safety Improvements in Urban Mobility Context*. Phd thesis, Mecahnical and Industrial Engineering Department, Tallinn University of Technology, Tallinn, Estonia, 2022.

[115] Ehsan Malayjerdi, Raivo Sell, Mohsen Malayjerdi, Andres Udal, and Mauro Bellone. Practical path planning techniques in overtaking for autonomous shuttles. *J. field robot.*, 39(4):410–425, June 2022.

[116] Hatem Darweesh, Eijiro Takeuchi, Kazuya Takeda, Yoshiki Ninomiya, Adi Sujiwo, Luis Yoichi Morales, Naoki Akai, Tetsuo Tomizawa, and Shinpei Kato. Open source integrated planner for autonomous navigation in highly dynamic environments. *Journal of Robotics and Mechatronics*, 29(4):668–684, 2017.

[117] Anshuman Sharma and Zuduo Zheng. Connected and automated vehicles: opportunities and challenges for transportation systems, smart cities, and societies. *Automating Cities: Design, Construction, Operation and Future Impact*, pages 273–296, 2021.

[118] Mohsen Malayjerdi, Andrew Roberts, Olaf manuel Maennel, and Ehsan Malayjerdi. Combined safety and cybersecurity testing methodology for autonomous driving algorithms. In *Proceedings of the 6th ACM Computer Science in Cars Symposium*, pages 1–10, 2022.

[119] Foretellix. Technology - Foretellix. https://www.foretellix.com/technology/, 2023.

[120] ASAM - Standardization for Automotive Development. OpenSCENARIO. `https://www.asam.net/standards/detail/openscenario/`. Accessed: 2023-4-18.

[121] CARLA. Autonomous Driving Leaderboard. `https://leaderboard.carla.org/`. Accessed: 2023-4-18.

[122] Christopher Medrano-Berumen, Mohsen Malayjerdi, Mustafa İlhan Akbaş, Raivo Sell, and Rahul Razdan. Development of a validation regime for an autonomous campus shuttle. In *2020 SoutheastCon*, pages 1–8. IEEE, 2020.

[123] Andrew Roberts, Mohsen Malayjerdi, Mauro Bellone, Olaf Maennel, and Ehsan Malayjerdi. Analysing adversarial threats to rule-based local-planning algorithms for autonomous driving. In *Symposium on Vehicle Security and Privacy (VehicleSec)*. IEEE, 2023.

[124] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[125] Mohsen Malayjerdi, Vladimir Kuts, Raivo Sell, Tauno Otto, and Barış Cem Baykara. Virtual simulations environment development for autonomous vehicles interaction. In *ASME International Mechanical Engineering Congress and Exposition*, volume 84492, page V02BT02A009. American Society of Mechanical Engineers, 2020.

[126] Cheng Wang, Christoph Popp, and Hermann Winner. Acceleration-based collision criticality metric for holistic online safety assessment in automated driving. *IEEE Access*, 10:70662–70674, 2022.

[127] Gunel Jahangirova, Andrea Stocco, and Paolo Tonella. Quality metrics and oracles for autonomous vehicles testing. In *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, pages 194–204. IEEE, 2021.

[128] Rahul Razdan, Mustafa İlhan Akbaş, Raivo Sell, Mauro Bellone, Mahesh Menase, and Mohsen Malayjerdi. Polyverif: An open-source environment for autonomous vehicle validation and verification research acceleration. *IEEE Access*, 11:28343–28354, 2023.

[129] Mohsen Malayjerdi, Bariş Cem Baykara, Raivo Sell, and Ehsan Malayjerdi. Autonomous vehicle safety evaluation through a high-fidelity simulation approach. *Proceedings of the Estonian Academy of Sciences*, 70(4):413–421, 2021.

[130] Alessandro Freddi, Mario Salmon, Alessandro Freddi, and Mario Salmon. Introduction to the taguchi method. *Design principles and methodologies: from conceptualization to first prototyping with examples and case studies*, pages 159–180, 2019.

[131] Moveh Samuel, Mohamed Hussein, and Maziah Binti Mohamad. A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. *International Journal of Computer Applications*, 135(1):35–38, 2016.

[132] Mohsen Malayjerdi, Gemb Kaljavesi, Frank Diermeyer, and Raivo Sell. Scenario-based validation for autonomous vehicles with different fidelity levels. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2023.

Acknowledgements

This research was supported by the European Union's Horizon 2020 Research and Innovation Programme, under grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No 2014-2020.4.01.20-0289.

In the journey of completing this thesis, I am profoundly grateful to my family, and at the heart of this gratitude is my loving wife, Mansoureh. Her unwavering support, understanding, and patience have been the cornerstone of my academic pursuit. Her belief in me, even during the most challenging times, has been a source of motivation that I can never repay. I am also deeply indebted to my supervisor, Prof. Raivo Sell. His guidance, expertise, and mentorship have been invaluable throughout this research endeavor. I would like to extend my sincere thanks to my family, my respected supervisor, and all those who have contributed their support, guidance, and encouragement to the completion of this thesis.

Abstract
Scenario-based Validation of Safety and Performance of an Autonomous Vehicle by a Software in Loop Simulation Method

AV technology requires robust and reliable validation methods to ensure their safety and optimize their performance. This thesis is based on five research papers that focus on the scenario-based validation of an AV through the implementation of a Software-in-Loop (SiL) simulation method. The SiL approach allows for extensive testing in virtual environments, replicating real-world scenarios, and evaluating AV responses. The proposed methodology has been implemented on an autonomous shuttle currently operating on the TalTech campus.

The research begins with a comprehensive review of existing validation techniques, emphasizing the need for scenario-oriented testing to cover a wide range of critical situations. Traditional validation methods, such as track-based testing and real-world road trials, are limited in their ability to systematically explore a vast number of scenarios due to cost, time, and safety constraints. Scenario-based evaluation offers a more efficient and cost-effective approach, enabling engineers to assess AV performance and safety under various challenging circumstances.

The thesis then presents the design and implementation of the SiL/HiL low- and high-fidelity simulation platform, which involves integrating AV control software with the virtual testbed. The software control system, including perception, decision-making, and control modules, is connected to the virtual environment to create a closed-loop simulation. AV responses to simulated scenarios are evaluated based on safety metrics, performance indicators, and adherence to predefined rules and regulations.

To demonstrate the efficiency and validity of the proposed evaluation toolkit, a wide range of studies have been conducted. These studies include low- and high-fidelity simulations of an AV shuttle to examine its performance, and safety in addition to validating the reliability of the simulation result with real-life experiments. The thesis also highlights the importance of selecting representative scenarios that cover critical edge cases and challenging driving situations to ensure thorough validation.

The thesis concludes by highlighting the advantages of the proposed methodology. The cost-effectiveness and repeatability of this toolkit allow for extensive scenario coverage and early identification of potential issues. By conducting rigorous simulations, developers can enhance AVs' safety and performance, minimizing the risks associated with real-world testing. Furthermore, the findings contribute to the advancement of scenario-based validation techniques and pave the way for safe and reliable AV integration into our transportation ecosystem.

In summary, this work presents a comprehensive toolkit for scenario-based evaluation of AVs using a low- and high-fidelity SiL simulation method. The results obtained from extensive simulations demonstrate the efficacy of the proposed methodology in identifying vulnerabilities, optimizing decision-making algorithms, and validating AV's functionality under diverse scenarios. This research contributes to the development of robust validation techniques and facilitates the realization of safe and reliable autonomous transportation systems.

Kokkuvõte
Autonoomse sõiduki ohutuse ja jõudluse stsenaariumipõhine valideerimine tsüklisimulatsiooni meetodi abil

Autonoomse sõiduki (AV) tehnoloogia nõuab tugevaid ja usaldusväärseid valideerimismeetodeid, et tagada nende ohutus ja optimeerida nende jõudlust. See lõputöö põhineb viiel uurimistööl, mis keskenduvad AV stsenaariumipõhisele valideerimisele tarkvara-in-loopi (SiL) simulatsioonimeetodi rakendamise kaudu. SiL-lähenemine võimaldab ulatuslikku testimist virtuaalses keskkonnas, kopeerida reaalmaailma stsenaariume ja hinnata AV-vastuseid. Kavandatav metoodika on rakendatud praegu TalTechi ülikoolilinnakus töötaval autonoomsel süstikul.

Uurimistöö algab olemasolevate valideerimistehnikate põhjaliku ülevaatega, rõhutades vajadust stsenaariumipõhise testimise järele, et hõlmata paljusid kriitilisi olukordi. Traditsioonilised valideerimismeetodid, nagu rajal põhinev testimine ja tegelikud maanteekatsetused, on piiratud kulude, aja ja ohutuspiirangute tõttu suure hulga stsenaariumide süstemaatilise uurimisega. Stsenaariumipõhine hindamine pakub tõhusamat ja kulutõhusamat lähenemisviisi, võimaldades inseneridel hinnata AV jõudlust ja ohutust erinevates keerulistes olukordades.

Seejärel tutvustatakse lõputöös SiL/HiL madala ja kõrge täpsusega simulatsiooniplatvormi disaini ja juurutamist, mis hõlmab autonoomse sõidukijuhtimistarkvara integreerimist virtuaalse katsealusega. Tarkvara juhtimissüsteem, sealhulgas taju-, otsustus- ja juhtimismoodulid, on ühendatud virtuaalse keskkonnaga, et luua suletud ahela simulatsioon. AV-vastuseid simuleeritud stsenaariumitele hinnatakse ohutusmõõdikute, jõudlusnäitajate ning etteantud reeglite ja eeskirjade järgimise põhjal.

Kavandatava hindamisvahendite komplekti tõhususe ja kehtivuse demonstreerimiseks on läbi viidud palju erinevaid uuringuid. Need uuringud hõlmavad AV-süstiku madala ja kõrge täpsusega simulatsioone, et uurida selle jõudlust ja ohutust, lisaks simulatsioonitulemuste usaldusväärsuse kinnitamisele reaalsete katsetega. Lõputöö rõhutab ka seda, kui oluline on valida esinduslikud stsenaariumid, mis katavad kriitilisi äärejuhtumeid ja väljakutseid pakkuvaid sõiduolukordi, et tagada põhjalik valideerimine.

Lõputöö lõpetuseks tuuakse välja pakutud metoodika eelised. Selle tööriistakomplekti kulutõhusus ja korratavus võimaldavad ulatuslikku stsenaariumi katvust ja võimalike probleemide varajast tuvastamist. Rangete simulatsioonide abil saavad arendajad parandada AV-de ohutust ja jõudlust, minimeerides reaalmaailma testimisega seotud riske. Lisaks aitavad leiud kaasa stsenaariumipõhiste valideerimistehnikate edasiarendamisele ja sillutavad teed ohutule ja usaldusväärsele AV-integratsioonile meie transpordiökosüsteemi.

Kokkuvõttes esitab see töö põhjaliku tööriistakomplekti AV-de stsenaariumipõhiseks hindamiseks, kasutades madala ja kõrge täpsusega SiL-i simulatsioonimeetodit. Ulatuslike simulatsioonidega saadud tulemused näitavad pakutud metoodika tõhusust haavatavuste tuvastamisel, otsustusalgoritmide optimeerimisel ja AV funktsionaalsuse valideerimisel erinevate stsenaariumide korral. See uurimus aitab kaasa tugevate valideerimistehnikate väljatöötamisele ning hõlbustab ohutute ja usaldusväärsete autonoomsete transpordisüsteemide realiseerimist.

Appendix

Paper I

Christopher Medrano-Berumen, **Mohsen Malayjerdi**, Mustafa İlhan Akbaş, Raivo Sell, and Rahul Razdan. "Development of a validation regime for an autonomous campus shuttle." In 2020 SoutheastCon, pp. 1-8. IEEE, 2020.

Development of a Validation Regime for an Autonomous Campus Shuttle

Christopher Medrano-Berumen*, Mohsen Malayjerdi†, Mustafa İlhan Akbaş‡, Raivo Sell† and Rahul Razdan*
*Advanced Mobility Institute, Florida Polytechnic University
{cmedranoberumen2844,rrazdan}@floridapoly.edu
†Department of Mechanical and Industrial Engineering, Tallinn University of Technology
{momala,raivo.sell}@taltech.ee
‡Department of Electrical, Computer, Software & Systems Engineering, Embry-Riddle Aeronautical University
akbasm@erau.edu

***Abstract*—The autonomous vehicles need to be validated with a reliable and repeatable methodology to be accepted by the public. In this paper, we present our methodology to develop a validation regime for the decision making system of an autonomous vehicle operating in a certain road network. The methodology starts with the thorough analysis of the selected roads. Then these roads are divided into atomic units, each of which is unique for testing purposes. The atomic units are modeled in simulation using our existing scenario generation framework, which allows for the stress testing and edge scenario discovery. Then the decision making software of the vehicle under test is taken in the loop to execute the tests. The methodology is applied to the autonomous campus shuttle currently operating at the Tallinn University of Technology campus. The shuttle's route is analyzed and modeled in simulation to create the testing scenarios. The methodology will be a complete validation scheme as the shuttle is tested in the field with a variety of the corner test cases discovered by our methodology.**

***Index Terms*—Autonomous Vehicles, Validation, Verification, Simulation, Testing, Scenario Testing**

I. Introduction

Autonomous Vehicle (AV) technology has the potential to have a fundamental impact on various fields such as the transportation, automotive, farming, and so on. There have been significant advances in the artificial intelligence (AI) engines and perception systems, which form the core technologies of AVs. On the other hand, there are still unsettled challenges in the testing and validation of AVs [1]. The full potential of AV technology cannot be realized and regulators cannot have the tools to install the needed safety regime for broad-based AV proliferation unless these challenges are resolved.

The critical features of a test and validation regime must be based on a clear model upon which the operation of the tested system can be reasoned. Further, this model must be able to receive feedback from the physical world through accidents or physical tests and the state space for the tests must be understood sufficiently to get to a notion of completeness. All of these must follow a procedure that maximizes safety and builds confidence in the public.

We observe that the current AV testing space has none of the aforementioned critical characteristics. The current commercial solutions are using ad-hoc methods such as miles driven to provide some indication of validation [2]. However, no fundamental structure has been offered to demonstrate the robustness of the solutions. In fact, there is no assurance even for the safety impact of various software updates. The use of "real world" testing through the most common "shadow driving" method has various disadvantages both from the point-of-view of verification convergence and also public safety [1].

978-1-7281-6861-6/20/$31.00 © 2020 IEEE

In our previous work [3], [4], we presented our testing and validation approach for AVs. A center point of this approach is a model for a "scenario", which allows a semantic language to specify the AV test environment. Using these scenarios, we build an environment to provide a model for completeness, a flow for constant update from physical feedback, and drive the test generation process for "edge" test cases.

This paper presents the utilization of this methodology for an autonomous campus shuttle at Tallinn University of Technology (TalTech), Estonia (See Fig. 1). TalTech Autonomous vehicles research group is well-known for its AV shuttle - ISEAUTO [5], [6] that is operational in the campus for demonstration purposes. Because of the slower speeds and relatively low complexity of the campus AV path (See Fig. 2), TalTech offers an ideal starting point to hash out and prove the utility of our testing and validation methodology. Additionally, it is complementary for the efforts of the TalTech Autonomous vehicles research group working on the development of AV shuttle.

The overall research project is planned to be executed in three stages. First, the TalTech route and environment is modeled in the validation flow to build a framework for coverage and generate representative "edge" test cases. Second, these scenarios are going to be fed into software-in-the-loop (SiL) simulation to test the decision making system of the shuttle. The progress towards these two phases are given in this paper. The final stage will be a physical demonstration in the TalTech campus. Beyond verifying functionality, key aspects of this demonstration will be to instrument the shuttle in a manner to detect test conditions which were not examined in simulation. This very critical feedback process creates a situation to debug the test and validation system. The most important result of this project will be to prove an AV test and verification methodology which can then be subsequently scaled to other

Fig. 1. TalTech autonomous campus shuttle.

environments.

The remainder of this paper is organized as follows. Related work is given in Section II. The validation approach and the scenario generation process are given in Section III. We present the implementation details and examples in Section IV and conclude in Section V.

II. Related Work

According to the annual reports by the National Highway Traffic Safety Administration (NHTSA), number of traffic fatalities are close to 40,000 every year and an overwhelming portion of these (more than 90% for 2017) are due to human error [7]. AVs are expected to take the responsibilities of perception and decision making of driving from human drivers and it is hoped that this transition will reduce the number of crashes related to human error. Despite the foreseen advantages, the major barrier for wide-scale adoption of AVs is the validation and verification regime to assure safety. To address this barrier, a process, which builds an engineering argument for assuring safety, must be developed. In this process, typically first a conceptual understanding of the problem is built and supported through virtual models. Then using the conceptual model, a test regime is built to test the model and build an argument for correctness. The state space of tests is examined within the modeling environment to develop metrics for completeness. A structure is constructed where field testing feeds back into this flow such that safety is always rising. The classic V paradigm model intertwined with this methodology is used as a mechanism to enable concurrent design and test. In this paradigm, mathematical models, which have been correlated with a bottom-up component level characterization stage, are used early in the design stage. As the design is refined, physical components can be substituted to a point when system level tests can be performed on the whole physical design. Modeling issues are often corrected with a virtual to physical diagnostics flow.

The combination of the conceptual safety regime and the V design process have been effectively used to build robust safe systems in automotive industry. However, the addition of the perception and decision making has increased the complexity in the safety problem. The conceptual models for the AV operation, test regime to build confidence, completeness and coverage of the validation regime and accumulative learning through validation remain to be open research areas with no current solution. Without the solutions to these challenges, regulators have no means to address safety issues.

The current commercial solutions are mostly using shadow driving [8] with ad-hoc measures to provide some indication of safety. There are also testing approaches focusing on modeling and simulation to verify that newly learned maneuvers are tested until they can be performed at a satisfactory rate. For instance, SiL and Hardware-in-the-loop (HiL) testing methods bring the vehicle features into tests. Others use simulations based on scenarios that their vehicles in the real world encountered [9]. Each important scenario from real life is fuzzed into generating more scenarios based on the original to strengthen the coverage. There are also several initiatives for the AV validation by integrating different techniques such as the Intelligent Testing Framework [10] and PEGASUS [11]. Recently, there have been approaches to standardize the AV validation approaches. UL 4600 is a draft standard based on setting scope requirements for an overarching safety case, planned to evolve with the accumulated experience [12]. Open Measurable Scenario Description Language (M-SDL) [13] is another initiative originated in industry to create a high level language that aims to simplify the capture, reuse and sharing of scenarios. Such standardization efforts have the potential to be instrumental in the extension of our approach.

Most of the existing validation solutions try to test the whole vehicle stack, starting from scene perception and understanding all the way to performing the decided action. Our approach follows the ‘separation of concerns’ principle, where we divide the validation into several phases. In this paper, we use our validation approach to test the decision making of an operational AV and demonstrate the application of the approach with SiL testing.

III. Development of the Validation Regime

In this section, we present the methodology to apply the abstract scenario generation system for addressing the critical issues of AV validation and verification of the TalTech autonomous campus shuttle. It is important to note that we separate the concerns for perception and decision making. Therefore, the scenario generation of our system focuses on the test and evaluation of the decision making system of the shuttle independent of the perception components.

In our approach, we use a semantic language for describing driving scenarios that can take random values as inputs and then convert them into a logical driving scenario in simulation [3]. For instance, a road segment can be described by the function of the line that it follows as well as the width at each point. The goal in defining different road pieces is to easily constrain the generated scenario to realistic road networks and situations. We use geometric primitives to generate roadways with every possible curve and number of lanes in the road

Fig. 2. Autonomous shuttle operating on its route at TalTech campus.

topology. By exploring all inputs to these pieces and to this language in general, the behavior of an AV can be tested in any situation.

A. Validation Approach

The goal of the AV shuttle at TalTech campus is to demonstrate the capabilities of the developed vehicle and develop an operational AV shuttle system, which can co-exist with other traffic participants. The biggest concern to operate an AV shuttle system in any environment is safety. In the example of a crowded campus day, many students are populating the university, which has the potential to cause high density of spontaneous road crossings of pedestrians over both marked and unmarked sections.

Although the TalTech campus road stretches over a larger area, the AV shuttle pilot route is around 2 miles with dedicated beginning and end locations (See Fig. 2). Our approach is designed to study in detail the AV interaction scenarios from a safety perspective as well as the traffic impact of using an AV along a campus.

Given the specific circumstances of the pilot validation system, the simulation system will fully model the relevant pieces. These include the following:

- Definition of relevant driving scenarios in the interaction between the AV shuttle and its environment along the route.
- Analysis of sections along the pilot route for safety.
- Simulation of critical safety risk scenarios and identification of improvement options (vehicle side, infrastructure side) with simulated data.
- Software-in-the-loop (SiL) testing of the vehicle using the generated scenarios.
- Use of collected data through a physical AV test vehicle along the specified route to validate safety simulations and to validate suggested improvement measures.

Based on the findings of the simulation and the validation of the simulation model through physical experiments, it should be possible to build an enhanced model to address the operation of AV vehicles on campus in mixed traffic and its impact on traffic flow as well as operational safety risk.

It is important to note that along the suggested TalTech AV shuttle pilot route, a single low speed lane integrated in a shared space with bicycles and pedestrians is experimented before and demonstrated to be feasible.

The validation approach will include use cases that will be considered for both the simulated as well as the actual experiments and some main examples can be listed as follows:

- People movement:
  - Use case 1: Drop-off and pick-up of people at dedicated AV shuttle stations
  - Use case 2: Impact of different vehicle dynamics scenarios (e.g. hard braking) on people inside the shuttle while AV shuttle is moving at different speed levels, both considering normal operation as well as emergency maneuvers.
  - Use case 3: Impact of different AV vehicle driving scenarios on other traffic participants (reaction pattern analysis, e.g. interaction with bicycles drivers or golf karts)
- Goods movement:
  - Use case 1: Loading and unloading of goods into the vehicle at dedicated AV shuttle stations
  - Use case 2: Point to point movement of goods along fixed route without intermediate stops

From a safety perspective, all segments of the AV route need to be carefully explored in terms of potential road layout configurations and vehicle/bicycle/pedestrian configurations. It needs to be determined how a low-speed AV shuttle operation can be implemented both in the existing road configuration and lane layouts as well as how future road modifications and lane configurations should look like. Ultimately a solution needs to be found with the lowest safety risk level. Long term goal is to shift the purpose of the shuttle from demonstration to shared use on campus. For efficient transportation, it is desirable to shift the student movement to AV shuttles as much as possible.

B. Scenario Generation for TalTech AV Route

The focus of this section is breaking down the main road crossing TalTech's campus into major segments and describing the development process. This process is used to generate the segments to be used in our scenario generation framework. The segments define what road pieces would be needed to compose the campus path in the simulation for testing and validation. Since our focus is validating the decision making without the consideration of other factors such as environmental conditions, we utilize MATLAB AD toolbox [14] with its cuboid and low-fidelity simulation capability.

To conduct a thorough verification of an AV that will be operating within the campus, it is important to define road pieces and their properties and be able to alter and randomize the scenarios occurring on them. The process with which the pieces are defined and used to build the validation framework

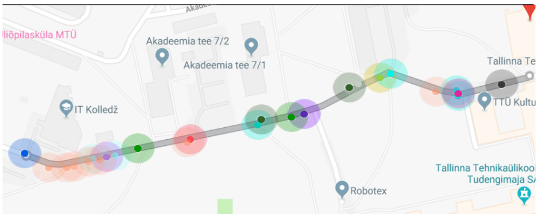

Fig. 3. The unique segments identified on AV shuttle route for modeling.

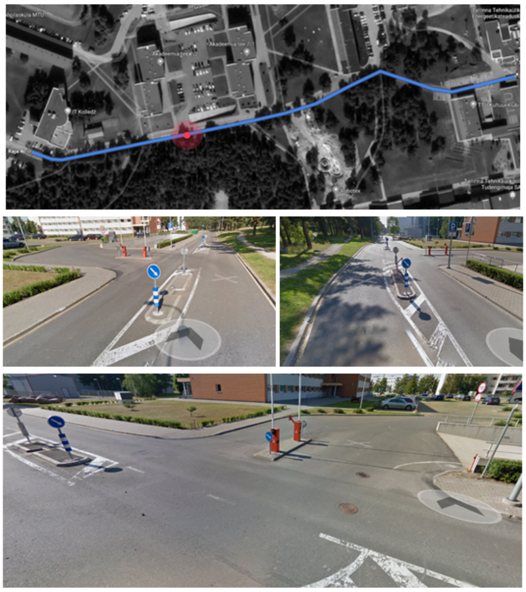

Fig. 4. Side Lot Enter as found in TalTech path

for a vehicle operating there can then be extrapolated to any other location until enough pieces and scenarios are defined to achieve Level 5 Autonomous validation and verification.

We define all of the unique segments found along the TalTech path after analyzing driving footage and Google Maps as given in Fig. 3. Most of the path on the campus can be described as a generic bidirectional road with no set boundaries or lane marker. The parts of the path with no new additions will be constructed using the already existing Multilane Road model. The entire main road has two lanes wide with some extra room and no lane markers. Areas with sidewalk or sidewalk path entries are also described the same way and have higher pedestrian crossing probability. In this section, we describe how our framework is extended to include the segments defined for the TalTech path in terms of updates to parameters, current pieces and new pieces with new parameters.

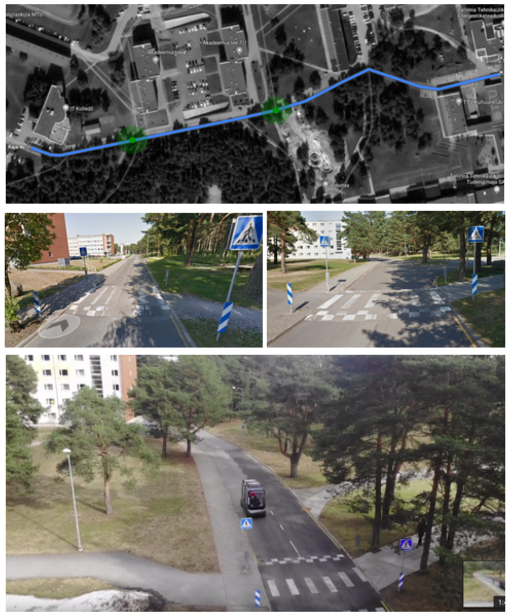

Fig. 5. Single Pedestrian Crosswalk as found in TalTech path

1) Side Lot Enter: The side lot entrance shows up once on the path and is a way for vehicles driving in either direction of the road to enter and exit neighboring parking lot. Along the path, the side lot enter is where most other traffic drives to. Going into the parking lot, there is a divider with specific signage and barriers. In the center of the main path, right next to the lot entrance, is an open area splitting the medians on either side of the road for left turning vehicles.

Since the side lot entrance is an important piece for developing scenarios, it is analyzed thoroughly to model it accurately for TalTech campus. For this piece, the number of lanes in the road and side lot are modeled as attributes so that their properties can be varied. The width of the medians also determine the geometry of the central turning space and it is added as an attribute. The length of the medians follows the length of the road, keeping the current road length parameter functioning as is.

It is important to note that in the test scenarios including this road segment, the emphasis will be put on vehicles entering and leaving whichever end is not connected to the rest of the road network to simulate the traffic a vehicle at this point is dealing with.

2) Single Pedestrian Crosswalk: This piece shows up three times along the path, all with the same design and signage. However, one of the pieces appears to be built before the others due to its wear. All of the crosswalks are elevated to serve as speed bumps as well, causing vehicles to slow down significantly. These will represent a high frequency pedestrian

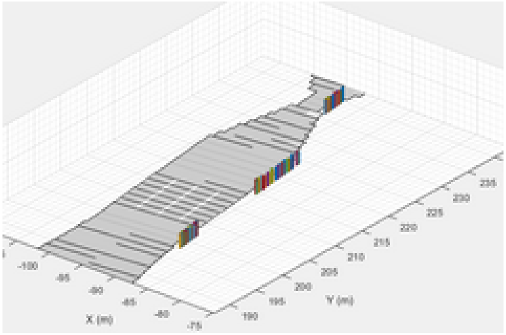

Fig. 6. Simulation model of Single Pedestrian Crosswalk

Fig. 7. Three-Way Intersection as found in TalTech path.

crossing area, so they are expected to be more active than the pedestrian path entries.

Some parameters to consider in this piece are the width of the walkway, which should be at least two meters and go up to five meters representing just about all walkways of this kind. In terms of actors, the number, type (groups or solo) and rate of people crossing are also added attributes to be varied.

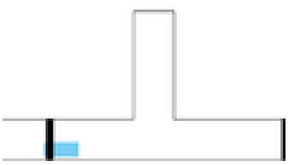

Fig. 8. Simulation model of Three-Way Intersection.

3) Three-Way Intersection with Barrier: At the beginning of the shuttle route, there are two areas where a road exits the main path perpendicularly like a three-way intersection. One of these goes into a parking lot while the other goes into the area surrounding a building with a barrier. The logic that a car follows when arriving at a barrier can be critical for some test scenarios. Hence, by including the barrier as a part of the 3-way intersection, the road piece becomes more versatile.

Parameters for this segment includes the number of lanes, following a similar format for the already implemented four-way intersection of the framework. Other parameters are boolean, representing whether the main road is bidirectional and whether the outgoing road is bidirectional. The properties of the road that connects to the barrier would resemble that of the multi-lane road.

4) Extra Road Area: The shuttle route has an extra stretch of road next to a group of dumpsters. This piece is on the same asphalt plane as the road next to it and not above some curb. Both the dumpster area and this extra area extending from it are separated from the main road simply by a lane marker. This area is distinguished from the dumpster areas for accurate representation, and its geometry and separating lane marker are its varied properties.

5) Open Area: Once the path arrives at the end, it reaches an open area surrounded by TalTech buildings, some parking spaces, and pedestrian pathways. There is an actual road on the other side beneath a sky bridge connecting two of the buildings.

This piece will have the most stochastic properties, as there is no defined path for vehicles to travel and no defined path for people to walk through. With high activity, vehicles will be moving around and parking; and pedestrians will be walking around the space.

6) Garbage Truck Entry: At two points along the path, the road provides some connection to an area where garbage trucks can pick up the dumpsters. On one of the areas, the curb drops as seen above for the garbage truck to drive in, and in the other, the road is simply extended to the side with a lane marker between the road and the area for dumpsters.

Unless the logic for garbage trucks is going to be implemented in some format, either as the ego vehicle or as an actor that the ego vehicle interacts with, this will be part of the previously mentioned extra area (which will be tagged onto the multilane road in the simulation model).

7) New Road Piece Attributes:

a) Median: The median is a unique part of the road pieces that needs to be added to he models. The 'median' shows up in the west half of the path and splits the nonspecific lanes going in either direction. At the start of the median on either side, a sign points to which side the driver should go to.

In the simulation, the median is modeled so that it can be varied in terms of its width, length, as well as how fast it tapers off on either end. The height may or may not be an important metric for the 'median' piece in general as most medians use no walls to split lanes of the same road, and they

Fig. 9. Median as found in TalTech path.

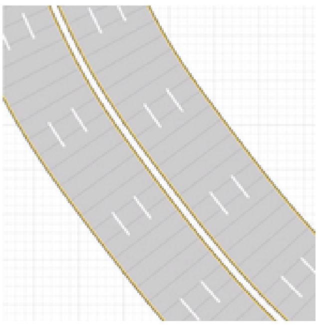

Fig. 10. MATLAB implementation of Median as Attribute.

are not pertinent to how a car should act in a scenario. Rather than creating a separate road piece, median is appended to the current model's multi-lane road to take advantage of the existing logic for creating unique geometries.

b) Outlet: In the TalTech shuttle, there are two outlets that do not go anywhere surrounding the median and leaving empty asphalt rectangles extending from the road. The second outlet goes into a larger area but is blocked off, and therefore serves essentially the same purpose.

In the simulation, if the outlet is pertinent to scenarios, then its width and horizontal height will be varied, determining the size of the rectangle. The position along the road and whether it is on the right or left side will also be set up as a parameter. Rather than make this a unique piece, this will also be supplemented to the multilane road.

c) 'AV Only' Segment: The campus AV path previously had two posts along with a no entry sign mark a point along the road that blocks access. However, the signage changed recently and now an "AV only" sign sits there with no posts. Scenarios that are meant to replicate the real TalTech path will not have other vehicle actors beyond this point, only the vehicle under test (or more AVs).

8) Sidewalk Entry: At a few points along the path, a small sidewalk reaches the road without arriving at a pedestrian walkway. This implies people are crossing through here, but not enough to justify building another walkway.

This is implemented as an extension of the multilane road to mark a place of higher frequency pedestrian crossing. Since there is no walkway, it is important to note that the path the pedestrian takes might not always be straight across, but rather going out at some angle from the starting point. To define the probability, either any pedestrians placed are placed on one of these or the distribution of starting points for pedestrians lean in favor of these points. A high probability of pedestrian crossing must be defined at these points.

a) Grass Path Entry: Same as the sidewalk path entry, these grass paths were identified due to their obvious wear from use when crossing. These may have an equal or smaller amount of people crossing than the sidewalk equivalent.

The grass path entry exists as an extension/property of the multilane road and will be placed in the same category as the sidewalk path entry. Because it shares all properties except visual with the previous piece, there's no real need to also account for it except for differing the frequency of pedestrian crossings, which will be accounted for by making that property variable. See previous entry for frequency definitions.

C. Software-in-the-Loop Testing with Generated Scenarios

It is time-consuming, costly and risky to use an untested control algorithm directly in a road test of an AV. Accordingly, sufficient and reliable simulation tests should be performed to examine the reaction of the algorithms in different scenarios before deploying them to the real vehicle. Therefore, SiL has become a critical and indispensable part of self-driving technology development.

ISEAUTO design and development process follows early stage design approach [15] supported by SysML technique [16] developed jointly by Estonian-Finnish research offering high modularity and fast interface creation process. The autonomy is achieved by running Autoware [17], a ROS based open-source software, enabling the self-driving functionality. Autoware platform provides an environment to associate all sensor data such as cameras and lidars for creating a unified observation and then make decisions based on this information by its control algorithms.

In our aprroach, MATLAB AD toolbox is used as the simulation software to create a virtual scene based on the real specified path in TalTech campus. On the other hand, the decision-maker algorithm, which runs on the vehicle side in the Autoware software, was applied to control the velocity and steering of the simulated vehicle. However, using the control and lane follower algorithms standalone needs some modification in the Autoware software. All necessary information that is provided by real sensors for the control algorithms must be replaced by new virtual sensor data from the simulation side. Additionally, actuation command issued by Autoware that was sent to the motor controller hardware must be sent to the MATLAB side. In this simulation, a two way communication is established between ROS and MATLAB to subscribe and publish data to and from specific nodes. Fig. 11 shows the simulation loop, which is now complete as the Autoware is connected to MATLAB.

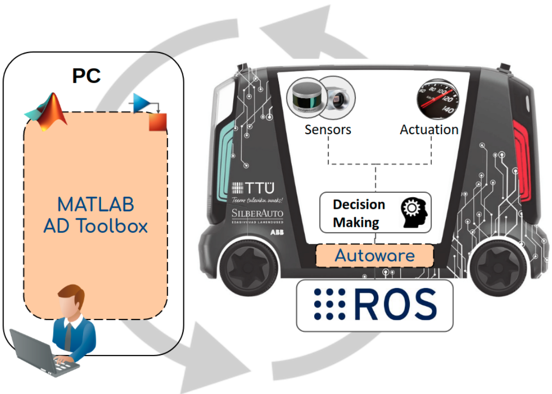

Fig. 11. Schematic view of the SiL testing with simulation loop.

MATLAB scenario generation enables the examination of the TalTech AV shuttle's decision-making algorithm in different cases and interaction with other vehicles, bicycles and pedestrians. Fig. 12 demonstrates how control algorithms are developed through simulation of different scenarios. When the algorithms successfully pass a test, the next scenario is created. The algorithms are modified according to the results of simulation scenarios that the AV fails. This process continues until an adequate number of cases are examined. These modified algorithms are then deployed to the real vehicle to perform a test ride.

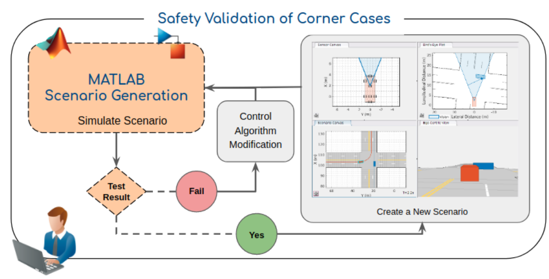

Fig. 12. SiL is used with scenario generation to develop control algorithms.

IV. Implementation

The first step in our work flow is analyzing the path that the AV operates on and breaking it down into the types of atomic

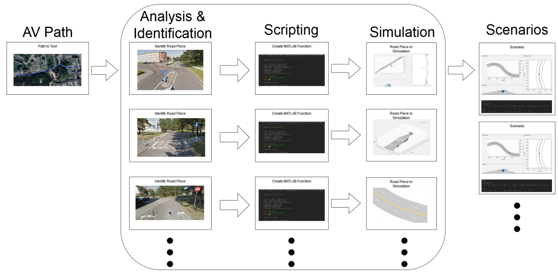

Fig. 13. Overview of the scenario generation workflow

Fig. 14. Entrance to a parking lot at TalTech AV route

pieces that make up the path. These pieces are then modeled in MATLAB to be able to generate scenarios that directly reflect the analyzed path as well as variations on it. The developed pieces are also collected in a database to be able to generalize the methodology to all paths that AVs operate on. An overview of the workflow is given in Fig. 14.

The first AV path that our methodology is applied on is the AV testing path at TalTech. The route includes turns, medians, pedestrian crossings, and roads that connect it to parking lots, each of which had to be integrated into the scenario generation either as an individual road piece or as a property of theirs. Once the segmentation and programming process was complete, each section of the AV path could be tested with, adjusting its properties as necessary using our existing validation language [3].

An important implementation example from the main segments of the TalTech path that we focused on programming into our scenario generator as an individual piece was one of their buildings' entrances which led into its parking lot. This entrance (See Fig. 14) included a barrier leading to the parking lot and medians going either way into the main road. On top of cars entering and exiting the parking lot, there is a high frequency of pedestrian traffic, making this a complex interaction for an AV.

Once it was decided that this segment would become a

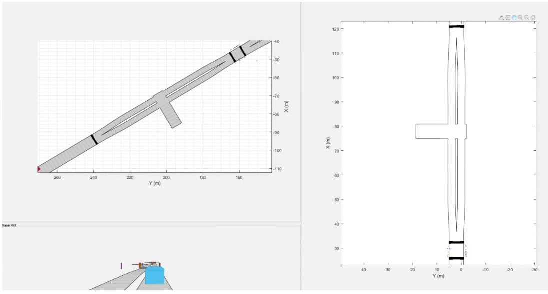

Fig. 15. 'Side Lot Enter' road piece in the scenario generator

piece, all of its attributes that could affect a logical scenario would have to be parameterized, allowing for testing in the space around the case this piece was based on. Each of these parameters exists as a column in our matrix-based language for describing road networks. Each row defines an individual road piece that is stitched to the piece described by the row before it if it does not break the scenario, and so the space around what type of road piece one of them comes from or continues to can be tested as well. For each of the new parameters defined during the analysis and design process, a new parameter is created and consequently a new column is added into the road matrix unless a previous parameter covers it directly or gives us the same type of a usable value such as a floating point number that intersections use.

The new road piece that we dubbed 'sideLotEnter' now exists as a possible road type that can be generated, with each of its properties able to be varied (See Fig. 15). This way, the same basic logic navigating this road can be tested with a different number of lanes, a different sized median, a different sized road, more or less pedestrians, and so on.

The same process is conducted for each of the segments discovered during the path analysis. This enables us to do stress testing and edge case finding by randomly generating matrices to be explored by the ego vehicle. The implementation process is dynamic, which allows implementation of different segments as needed and ensuring each of them has complete coverage over all the logical variations possible. In this paper, we focused on showing those that are already implemented.

The next phase of our research will be to implement SiL with TalTech's AV program to run through the randomly generated scenarios, and also to identify critical scenarios along the way. This phase takes advantage of computation power by running as many scenarios as possible in parallel and consecutively for stress testing.

We also work on a technique to identify corner cases. The rare, corner cases will be identified and stored in a database. Since the matrices used to in the scenes generate the scenarios in a deterministic manner, the tests are repeatable. Therefore, the corner cases can be reused exactly as when they were caught. These scenarios make it to the final phase as we perform testing using HiL and ViL.

V. Conclusion

The testing and validation of an AV is critical for its deployment in the transportation system. The validation approach in this paper takes the real problem of AV validation at TalTech campus and builds a workflow for it. This paper presents the completed phases of the project, where simulation approach is integrated with test generation methodology. The overall project combines simulation with SiL testing and finally a physical demonstration. The successful operation of this testing scheme will enable the generalization of the approach and removal of current ad-hoc and ineffective methods of validation.

References

[1] R. Razdan, M. I. Akbas, A. Sargolzaei, A. J. Alnaser, S. Sahawneh, S. Alsweiss, and J. Vargas, "Unsettled Technology Areas in Autonomous Vehicle Test and Validation," *SAE (Society of Automotive Engineers) EDGE Research Report EPR2019001*, June 2019.

[2] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[3] C. Medrano-Berumen and M. I. Akbas, "Abstract Simulation Scenario Generation for Autonomous Vehicle Verification," in *Proceedings of the IEEE SoutheastCon*. IEEE, April 2019.

[4] A. J. Alnaser, M. I. Akbas, A. Sargolzaei, and R. Razdan, "Autonomous vehicles scenario testing framework and model of computation," *Society of Automotive Engineers (SAE) International Journal of Connected and Automated Vehicles*, October 2019.

[5] R. Sell, M. Leier, A. Rassõlkin, and J.-P. Ernits, "Self-driving car ISEAUTO for research and education," in *IEEE International Conference on Research and Education in Mechatronics*, 2018, pp. 111–116.

[6] A. Rassõlkin, L. Gevorkov, T. Vaimann, A. Kallaste, and R. Sell, "Calculation of the Traction Effort of ISEAUTO Self-Driving Vehicle," in *IEEE International Workshop on Electric Drives: Optimization in Control of Electric Drives*, 2018.

[7] N. S. C. Statistics Department. (2017, Dec.) NSC Motor Vehicle Fatality Estimates. http://www.nsc.org/NewsDocuments/2017/12-month-estimates.pdf.

[8] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in california," *PLoS one*, vol. 12, no. 9, p. e0184952, 2017.

[9] D. Dolgov, "Google self-driving car project-monthly report-September 2016-on the road," Google, Tech. Rep., 2016.

[10] L. Li, W.-L. Huang, Y. Liu, N.-N. Zheng, and F.-Y. Wang, "Intelligence testing for autonomous vehicles: a new approach," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 158–166, 2016.

[11] H. Winner, K. Lemmer, T. Form, and J. Mazzega, "PEGASUS—First Steps for the Safe Introduction of Automated Driving," in *Automated Vehicles Symposium 2018*. Springer, 2018, pp. 185–195.

[12] P. Koopman, U. Ferrell, F. Fratrik, and M. Wagner, "A safety standard approach for fully autonomous vehicles," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2019, pp. 326–332.

[13] Foretellix Ltd., "Measurable Scenario Description Language Reference," Tech. Rep. 0.9, September 2019.

[14] Mathworks, *MATLAB and Automated Driving System Toolbox Release R2019a*. Natick, Massachusetts, US: The MathWorks Inc., 2019.

[15] R. Sell, E. Coatanea, and F. Christophe, "Important Aspects of Early Design in Mechatronic," in *6th international conference of DAAAM Baltic industrial engineering, Tallinn*, 2008, pp. 177–182.

[16] F. Christophe, R. Sell, and E. Coatanéa, "Conceptual design framework supported by dimensional analysis and system modelling language," *Estonian Journal of Engineering*, vol. 14, no. 4, pp. 303–316, 2008.

[17] 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 *ACM/IEEE International Conference on Cyber-Physical Systems*, 2018.

Paper II

Mohsen Malayjerdi, Vladimir Kuts, Raivo Sell, Tauno Otto, and Barış Cem Baykara. "Virtual simulations environment development for autonomous vehicles interaction." In ASME International Mechanical Engineering Congress and Exposition, vol. 84492. American Society of Mechanical Engineers, 2020.

Proceedings of the ASME 2020
International Mechanical Engineering Congress and Exposition
IMECE2020
November 16-19, 2020, Virtual, Online

IMECE2020-23362

VIRTUAL SIMULATIONS ENVIRONMENT DEVELOPMENT FOR AUTONOMOUS VEHICLES INTERACTION

Mohsen Malayjerdi[1], Vladimir Kuts[1], Raivo Sell[1], Tauno Otto[1], Barış Cem Baykara[1]

[1]Mechanical and Industrial Engineering Department, Tallinn University of Technology, Tallinn, Harju, Estonia

ABSTRACT

One of the primary verification criteria of the autonomous vehicle is safe interaction with other road users. Based on studies, real-road testing is not practical for safety validation due to its time and cost consuming. Therefore, simulating miles driven is the only feasible way to overcome this limitation. The primary goal of the related research project is to develop advanced techniques in the human-robot interaction (HRI) safety validation area by usage of immersive simulation technologies. Developing methods for the creation of the simulation environment will enable us to do experiments in a digital environment rather than real. The main aim of the paper is to develop an effective method of creating a virtual environment for performing simulations on industrial robots, mobile robots, and autonomous vehicles (AGV-s) from the safety perspective for humans. A mid-size drone was used for aerial imagery as the first step in creating a virtual environment. Then all the photos were processed in several steps to build the final 3D map. Next, this mapping method was used to create a high detail simulation environment for testing an autonomous shuttle. Developing efficient methods for mapping real environments and simulating their variables is crucial for the testing and development of control algorithms of autonomous vehicles.

Keywords: Virtual environment, interaction, simulations, safety, Autonomous Vehicle

1. INTRODUCTION

Geospatial data has an essential role in an estimated 80% of our daily decisions [1] and various urban planning activities. Aerial or satellite imagery collected through remote sensing or earth observation is used as a data source for many base map activities and creating virtual environments for different purposes. Previous research has demonstrated the use of satellite and aerial imagery as a way of extracting information for creating and updating maps [2], [3] as well as to provide input for urban models [4]. Essential features of the urban environment, such as roads and buildings, may then be digitized in the imagery either by experts [5] or in participatory mapping projects by a wider public [6].

During the last two decades, substantial work has also centered on automatic feature extraction from high-resolution satellite and aerial photos [7], [8], [9]. However, the temporal resolution of standard sensors is limited by the restricted availability of aircraft platforms and the orbit characteristics of satellites [10]. Another drawback is the cloud cover, which prevents the acquisition of images via these platforms. These restrictions hinder the use of satellites or manned aircraft for map updating purposes, as this may increase the cost and time. To provide the high-quality and up-to-date information required to support urban governance and informed decision-making, Van der Molen [11] calls for land surveyors to make use of the potential of new affordable, geospatial technologies.

Unmanned Aerial Vehicles (UAVs), which are proving to be a successful data acquisition tool designed to operate without a human pilot onboard, are an appropriate example of such emerging technology. The term UAV is commonly used, but other terms, such as drones, Unmanned Aerial Systems (UAS), Remotely Piloted Aircraft (RPA) or Remotely Piloted Aerial Systems (RPAS), have also been frequently used in the geomatics community [12].

The primary aim of the research project is to use gathered from real-world terrain point-cloud data and Virtual Reality (VR) technologies to develop the Digital Twins (DT) for robotics simulation [13]. The central hypothesis of the project is that the test method and metrics for human-robot teaming should be developed first in DT to increase the safety level of physical industrial robotic systems. However, the first step toward it is the creation of a feasible environmental model for this task, which is being aimed at the related paper.

The main aim of the paper is to develop an effective method of creating a virtual environment for performing simulations on industrial robots, mobile robots, and autonomous vehicles from the safety perspective for humans.

2. MATERIALS AND METHODS

The advancement of drone technology made it possible to take sensors into new heights. LiDAR equipped drones, aerial imagery have all become mainstream. Today, the capabilities of these machines are used in a vast spectrum from crop assessment to archeological discoveries. Now, more than ever, there is a need to map and digitalize a physical environment. The mapping of an environment can be performed for various reasons. It could be performed to enter places such as caves where it was not possible before, or it could be performed for simulation purposes.

In this research, the map was created based on the aerial images taken by a mid-size drone from the real specified path for running an autonomous shuttle in the Tallinn University of Technology campus. This process was carried out in three main steps. Fig. 1 shows these steps, which started from capturing images by the drone, then processing the data, and classifying them into different objects and finally created in the Unity environment.

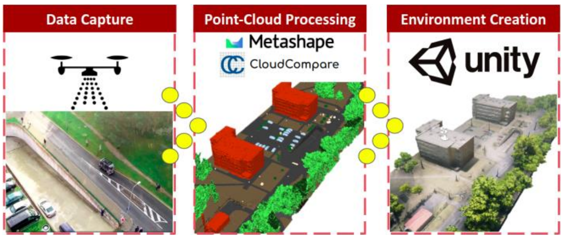

FIGURE 1: THREE MAIN STEPS TO CREATE A MAP BASED ON CAPTURING PHOTOS

2.1 MAPPING STEPS WITH UAV-S

In the photogrammetric approach of mapping, the data is captured by a camera-equipped drone, flying over the area to be mapped. Images are captured in a grid-patterned flight path in different camera angles at a determined altitude to ensure most coverage of the area. Capturing images is one of the most challenging steps in the mapping process due to the significant effect of the pictures on the final work. On the other hand, the weather conditions and the sunlight made it a little harder to get decent photos for processing. The taken images are geotagged thanks to the onboard Global Positioning System (GPS) module, and their orientation is determined by the drone's Inertial Measurement Unit (IMU). If both the orientation and the position of the images are known, they can be stitched together to form a 3D representation of the area. Fig. 2 shows a real captured picture and the corresponding object in the created map.

FIGURE 2: (LEFT): DRONE CAPTURED PHOTO, (RIGHT): 3D CREATED OBJECT

The taken pictures can be processed with photogrammetric software to generate a rough point cloud (see Figure 3).

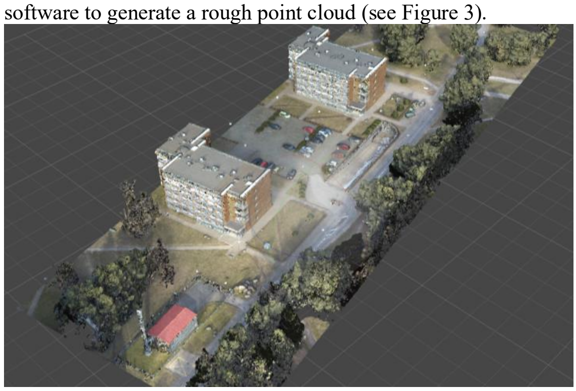

FIGURE 3: DENSE POINT CLOUD CREATED BY METASHAPE PRO

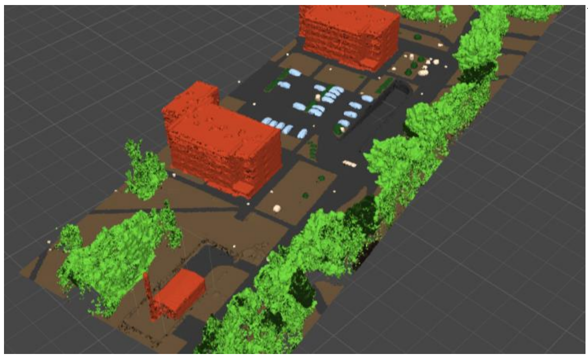

FIGURE 4: CLASSIFIED POINT CLOUD

Depending on the application, in complicated data sets, there can be some imperfections in the point cloud, such as power transition lines, trees, moving objects, etc. These may be classified or removed from the point cloud to ensure a clean and segmented end model of the area. The classified map was shown in figure 4. Segmentation is useful to label different objects and can be used to train self-driving vehicles in a simulated environment. Segmentation is also useful when generating details back on the terrain in unity (see Fig.6). Classified points can be read individually and turned into a grayscale mask to act as a template for where to generate terrain details such as trees, grass, etc. The classification can also be used in order to ignore and remove imperfections while creating the terrain in unity. Fig.5 shows a terrain created from the classified point cloud with vegetation and buildings removed. This approach is fast and cost-effective compared to other ones, as the drone does not have to carry significant payloads in order to complete its missions. It is also worth mentioning the convenience of RGB color data provided by the drone's camera (see Fig. 2).

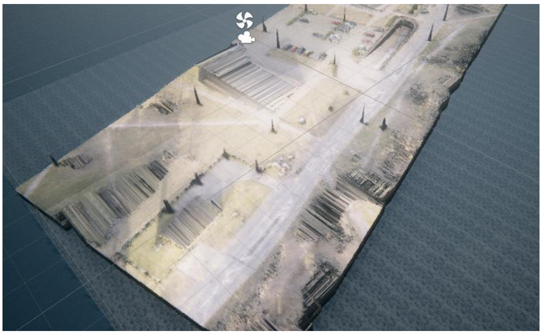

FIGURE 5: UNITY TERRAIN

The digitalization of an environment proves to be very useful in many scenarios. It can be applied to the construction industry to analyze ground characteristics to agricultural fields to optimize crop growth, and among many other things, it can be used in the development of autonomous vehicles to simulate and test the characteristics of a vehicle on a determined terrain. Not only would this make testing of sensory behaviors and control algorithms easy, but it can also reduce the risk of error to a minimum and reduce the costs of equipment. For an autonomous vehicle to be approved and certified for general operation, its software must be tested and validated as safe or more stable than a human equivalent. This requirement can be quantified statistically. Research by RAND [14] showed that it would require between 8 billion and 11 billion miles of road testing over a span of 400 to 500 years to show the reliability of autonomous cars to adequate levels of statistical trust in typical cases. Clearly, this is not practical. The only realistic way to overcome this constraint is simulating miles driven in virtual environments that were created based on the real ones.

FIGURE 6: UNITY TERRAIN WITH DETAILS

2.1 SIMULATION - USE CASE WITH SELF-DRIVING CARS

Safety verification and validation of AGVs (autonomous ground vehicles) are performed based on two main approaches. One is the online test, which refers to testing real self-driving vehicles in real scenes. On the contrary, simulations were used to perform tests in virtual scenarios, which is known as an offline test. Due to the fast development of AVs, exploring a structure for testing and verifying algorithms that can be applied quickly and safely is essential. It is expensive, time-consuming, and risky to use the untested algorithm directly on a real automated vehicle to make a road test. Accordingly, sufficient reliable offline tests should be performed to examine the algorithms and the HRI system before deploying them to the real vehicle.

FIGURE 7: ISEAUTO AUTONOMOUS SHUTTLE

This phase has become a critical and indispensable part of self-driving technology development. Mapping the environment is one of the first steps to define various scenarios as similar as possible to real terrains. This study is based on an experimental platform ISEAUTO that is an autonomous shuttle as a last-mile vehicle [15] (see Fig. 7). ISEAUTO is controlled by Autoware, an open source framework which is based on the Robotic Operating System (ROS).

Autoware software has many features, from processing raw data of sensors to creating motion commands. In addition to working with hardware in the loop processes, Autoware supports simulations. Simulation enables us to assess the control algorithms and new custom features before deploying them on the hardware, such as interactions with other road users or reaction of the vehicle on different scenarios. Maps play a vital role in Autoware operation, especially in navigation and localization, and should be considered carefully. Autoware itself can work with LiDAR point-cloud maps to navigate the vehicle. However, for utilizing the 3D graphical map like Fig. 6, Autoware needs to be connected with another simulator that works based on a graphical engine like unity. Fig. 8 shows the difference between these two maps, which are used in the simulations. In this figure, the left picture is a vector map merged with a LiDAR point-cloud map that is used in the Autoware software. The right one is the same area but in a 3D game engine, which is created by the method described. In the game engine terrain, we have more options to customize.

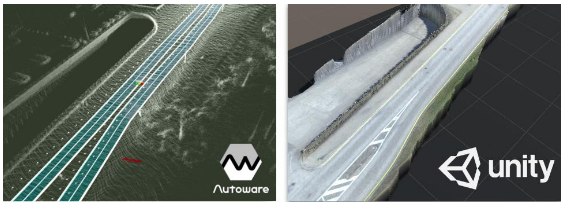

FIGURE 8: TWO DIFFERENT MAPS USED FOR SIMULATION IN DIFFERENT ENVIRONMENTS

In our simulation, LGSVL simulator was utilized in order to use the created unity map. it is a simulator that facilitates the testing and development of autonomous driving software systems. It enables developers to simulate billions of miles and arbitrary edge case scenarios to speed up algorithm development and system integration. This simulator gives us the ability to get virtual sensor data from the simulated environment, such as LiDAR point-cloud or cameras and receive the vehicle commands from the vehicle control software to navigate the vehicle inside the virtual environment. It is also fully compatible with our self-driving vehicle software that increases the simulations validity.

To validate the safety and develop the control algorithms, our simulation follows the diagram, which is shown in Fig. 9. Different scenarios, including high potential risk situations, will be examined through this method, such as overtaking, pedestrian crossing, and mix conditions.

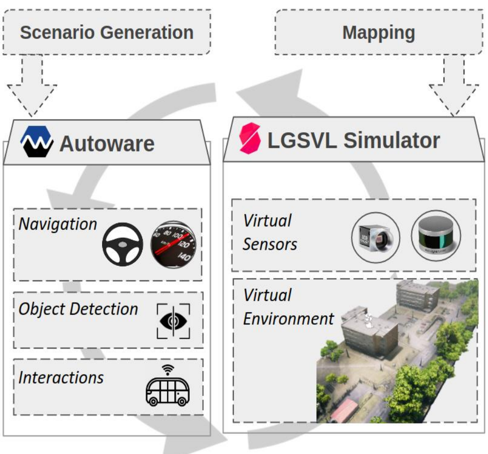

Figure 9: PROCEDURE OF THE SIMULATIONS THAT EVALUATE DIFFERENT SCENARIOS.

In the top layer of the simulation, the map and desired scenarios were generated. The base layer is where the simulations were performed by connecting Autoware to the LGSVL simulator. Autoware uses its features like navigation and object detection in the process, while the LGSVL simulator creates a virtual environment with virtual sensors. Virtual LiDAR and camera perception provide all the required data for Autoware control algorithms, such as localization and navigation, to operate in the simulation environment like the real one.

Through these simulations, algorithms will be modified and developed according to the test result of the simulated scenarios. If they meet the safety requirements, then a new scenario will be created, and the process continues until adequate cases are examined. These simulations also led to adding new HRI systems to the shuttle, such as LED displays, which results in increased safety and reliable interaction approaches [15].

3. RESULTS AND DISCUSSION

Now, it is clear that with a low-cost UAV and photogrammetric techniques, it is possible to obtain high-quality virtual environments which are adequate to use in different simulation scenarios. Compared to the time and cost of traditional photogrammetric surveys, this strategy represents a promising alternative. Creating maps from places, which can be the edge cases for HRI system scenarios in the AGV-s simulation, helps us to understand what is going on under our autonomous shuttle control software in the high-risk scenario of interactions. Furthermore, creating a virtual environment in a game engine like unity enables us to customize the environment and set any external variable (such as, weather simulation, friction on the road surface, etc.) to create realistic test scenarios.

Although, mapping with UAVs is an interesting undertaking but not without challenges. Most of them are related to weather conditions, battery management, and logistics. For capturing high-quality pictures, the amount of sunlight that is related to cloud cover is a key factor because of the reflection effect on them. Also, wind speed, rainy or snowy weather can affect the flight and photos quality. It is normal in some areas to wait for a good flight and data capturing condition for several months. In large-scale mapping projects, computer processing requirements are one of the main challenges faced by the application; however, there are some technological developments, such as GPU processing that are continuously reducing the computational bottleneck. However, with a little planning, these issues can be solved.

Another big challenge was, places where there were no point-cloud data which would cause holes in the generated terrain. This may happen due to dense vegetation in the area or poor coverage of the scanned terrain. It can also arise during post processing of the point-cloud, especially when ground points are separated from the rest. In order to mitigate these issues, the simplest solution is to interpolate the nearby pixels of the holes when creating necessary textures and images.

This research makes use of an RGB camera for creating a map based on real terrains. The obtained map is accurate enough to use as a virtual environment for our simulation purposes. However, a LiDAR sensor can be utilized to scan from the ground as well to improve the quality of the produced result.

The main future objectives of the related research:

1) To develop test methods, and Key Performance Indicators (KPIs) developed for HRI to evaluate autonomous robotic systems in regards to collaboration with the human in a standardized approach.

2) Import those test methods and metrics, to the Virtual test bench for various autonomous robotic systems that are enabled with the Virtual Reality (VR) toolkit for the human presence simulations.

3) Perform experiments with integrated metrics, UAV models integrated into a digital, realistic environment, and recap results for the optimization of the physical AGV-s.

4. CONCLUSION

Prototype Virtual Environment for safety experiment simulations was developed by usage of point cloud data of the real physical environments. This data was obtained by employing a surveyor UAV on a specific area. Moreover, simulations of the AGV safety concept were introduced for future development. In these simulations, the edge case scenarios which have more potential to cause errors in the HRI system will be reviewed. Future research will address mainly the integration of the HRI metrics and the AGV simulation model to the virtual environment for efficient and safe for human being's experimental work.

ACKNOWLEDGMENTS

The research has been supported by the "Smart specialization" project LEP18082EM “Applied research on the system of sensors and software algorithms for safety and driver assistance on remotely operated ground vehicles for off-road applications” and by the Estonian Research Council under grant PSG453 “Digital twin for propulsion drive of an autonomous electric vehicle”.

REFERENCES

[1] Heipke, C., Woodsford, P. A., and Gerke, M., Updating geospatial databases from images. In: Advances in photogrammetry, remote sensing and spatial information sciences: 2008 ISPRS congress book. Boca Raton: CRC Press, 355–362, 2008.

[2] Alexandrov, A., et al., Application of QuickBird satellite imagery for updating cadastral information. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVPart B2, 386–391, 2004.

[3] Ali, Z., Tuladhar, A., and Zevenbergen, J., An integrated approach for updating cadastral maps in Pakistan using satellite remote sensing data. International Journal of Applied Earth Observation and Geoinformation, 18, 386–398, 2012. doi: 10.1016/j.jag.2012.03.008

[4] Herold, M., Goldstein, N.C., and Clarke, K.C., The spatiotemporal form of urban growth: measurement, analysis and modeling. Remote Sensing of Environment, 86 (3), 286–302, 2003. doi: 10.1016/S0034-4257(03)00075-0

[5] Ottichilo, W. and Khamala, E., Map updating using high resolution satellite imagery – A case of the kingdom of Swaziland. International Archives of Photogrammetry and Remote Sensing, 34 (6/W6), 89–92, 2002.

[6] Mourafetis, G., et al., Enhancing cadastral surveys by facilitating the participation of owners. Survey Review, 47 (344), 316–324, 2015. doi: 10.1179/1752270615Y.0000000009

[7] Babawuro, U. and Beiji, Z., Satellite imagery cadastral features extractions using image processing algorithms: a viable option for cadastral science. International Journal of Computer Science Issues, 9 (4(2)), 30–38, 2012.

[8] Gruen, A., Baltsavias, E., and Henricsson, O., eds., Automatic extraction of man-made objects from aerial and space images (II). Boston: Birkhäuser, 2012.

[9] Awad, M.M., A morphological model for extracting road networks from high-resolution satellite images. Journal of Engineering, 2013, 1–9, 2013. doi: 10.1155/2013/243021

[10] Turner, D., Lucieer, A., and Watson, C., An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (UAV) imagery, based on structure from motion (SFM) point clouds. Remote Sensing, 4 (5), 1392–1410, 2012. doi: 10.3390/rs4051392

[11] Van der Molen, P., Rapid urbanisation and slum upgrading: what can land surveyors do? Survey Review, 47 (343), 231–240, 2015. doi: 10.1179/1752270614Y.0000000125

[12] Nex, F. and Remondino, F., UAV for 3D mapping applications: a review. Applied Geomatics, 6 (1), 1–15, 2014. doi: 10.1007/s12518-013-0120-x

[13] Kuts, V.; Sarkans, M.; Otto, T.; Tähemaa, T.; Bondarenko, Y., Digital Twin: Concept of Hybrid Programming for Industrial Robots - Use Case. ASME 2019 International Mechanical Engineering Congress and Exposition, 2B: ASME 2019 International Mechanical Engineering Congress and Exposition, IMECE2019, November 8-14, Salt Lake City, Utah, USA. ASME, 2019. doi: 10.1115/IMECE2019-10583.

[14] RAND Corporation, Driving to Safety: How Many Miles of Driving Would it Take to Demonstrate Autonomous Vehicle Reliability?, Kalra and Paddock, 2016.

[15] Rassõlkin, A., Sell, R., & Leier, M., Development case study of the first estonian self-driving car, iseauto. Electrical, Control and Communication, 2018. doi: 10.2478/ecce-2018-0009

Paper III

Mohsen Malayjerdi, Bariş Cem Baykara, Raivo Sell, and Ehsan Malayjerdi. "Autonomous vehicle safety evaluation through a high-fidelity simulation approach." Proceedings of the Estonian Academy of Sciences 70, no. 4, pp. 413-421. 2021.

Proceedings of the Estonian Academy of Sciences,
2021, **70**, 4, 413–421
https://doi.org/10.3176/proc.2021.4.07
Available online at www.eap.ee/proceedings

MANUFACTURING ENGINEERING

Autonomous vehicle safety evaluation through a high-fidelity simulation approach

Mohsen Malayjerdi[a*], Barış Cem Baykara[a], Raivo Sell[b,a] and Ehsan Malayjerdi[a]

[a] Mechanical and Industrial Engineering Department, Tallinn University of Technology, Tallinn, Estonia
[b] FinEst Twins Smart City Center of Excellence, Tallinn University of Technology, Tallinn, Estonia

Received 15 June 2021, accepted 13 July 2021, available online 2 November 2021

Abstract. The autonomous vehicle (AV) industry aims to design strategic plans to ensure the safety of the developed systems before their mass deployment. Real-road testing is shown to be impractical for validating these systems as it requires many years if not decades of testing in different environmental conditions. For solving this issue, the method should be complemented with simulation. The primary goal of this research was to develop advanced techniques in the safety validation area by using end-to-end simulation technologies. In this study, we present a simulation approach for safety evaluation of an AV shuttle, iseAuto, currently operating at the Tallinn University of Technology campus. We created a virtual environment by using geospatial data from the specified path on the university campus that includes all relevant features. Then, we converted the map to a 3D format applicable for the SVL simulator. Also, we provided the AV 3D model to use in the simulation and equipped it with the SVL virtual sensors to provide data for the Autoware perception algorithms, which is the control software of the shuttle. To show the efficiency of the proposed method, we designed two overtaking scenarios and observed the AV behaviour under the test. Finally, we demonstrate how the system enables us to evaluate AV's decision-making performance and safety in different situations.

Key words: autonomous vehicle, simulation, safety validation, high-fidelity simulator.

1. INTRODUCTION

Development of autonomous vehicles is one of the top trends in the automotive industry and the technology has been evolved to make them safer. Thus, engineers are facing new challenges, especially in moving toward Levels 4 and 5 of the Society of Automotive Engineers (SAE). To place autonomous vehicles (AVs) on roads and evaluate the reliability of their technologies, they have to be driven billions of miles [1]. It would take a long time to achieve this, unless with the help of simulation. Furthermore, due to the past real crash cases of AVs, a high-fidelity simulator has become an efficient and alternative approach to provide different testing scenarios for controlling these vehicles, also enabling safety validation before real-road driving [2–5]. Different high-resolution virtual environments can be developed for simulators by using cameras or lidars to simulate the scenarios as close to the real world as possible [6]. Also, virtual environment development enables us to customize and create various urban backgrounds for testing the vehicle. Creating a virtual copy of an existing intelligent system is a common approach nowadays, called a digital twin [7,8]. Extensive research and development, such as in [9,10] or [11], has been performed on AVs in recent years involving simulation. However, most of that has employed a low-fidelity simulator that cannot be a reliable reference for safety validations.

In this paper, we focus on the utilization of a high-fidelity simulator for an AV shuttle at Tallinn University

* Corresponding author, momala@taltech.ee

of Technology (TalTech), Estonia. The TalTech AV research group is well known for its AV shuttle, iseAuto [12], which is operational on the campus for experimental research purposes (Fig. 1). The vehicle was designed and developed from scratch by implementing the previously proposed mechatronic design methodology [13–15] with a special focus on early design stages. The first prototype development was a joint venture with TalTech and the local industry Silberauto [16]. This shuttle is controlled by Autoware [17], a Robotic Operating System (ROS) based platform for self-driving vehicles.

The overall research project was planned to be executed in two stages. First, the virtual environment was built based on the campus AV road area, where most of our real experiments take place, to create the simulation framework. We used geospatial images to generate the environment as a Unity terrain. Among different modern AV simulators such as CARLA [18], LGSVL (in 2021 the name was changed to SVL) [19] and Gazebo, we opted for SVL to be our simulator due to its compatibility with our control software (Autoware) and our terrain generation platform Unity. Another reason was to create different scenarios and perform software-in-the-loop (SIL) simulation by connecting Autoware with SVL. This enables us to find a better sensor configuration and settings in addition to the verification of the decision-making system that leads to safety assessment.

2. SIMULATOR

Simulation has been widely used in vehicle manufacturing, particularly for mechanical behaviour and dynamical analysis. However, AVs demand more due to their specific nature. Simulation in various complex environments and scenarios involving other road users with different sensor combinations and configurations enables us to verify their decision-making algorithms. One of the most popular robotic simulator platforms is Gazebo. It is based on ROS and utilizes physics engines and various sensor modules suitable for autonomous systems. Nevertheless, Gazebo lacks modern game engine features such as Unreal and Unity, which give the power to create a complex virtual environment and realistic rendering.

CARLA and SVL, on the other hand, are modern open-source simulators based on these game engines, Unreal and Unity respectively, which also have good compatibility with our AV stack Autoware. However, comparing these two is beyond the scope of our discussion, but we selected SVL as our simulator mainly because of its compatibility with our terrain generator Unity.

Figure 2 shows a full map of the simulation workflow and different layers in the simulator as well as the control software (Autoware). Vehicle 3D model and the virtual environment, which were built inside Unity, were imported to the simulator. The simulator allows cus-

Fig. 1. TalTech iseAuto – an AV shuttle.

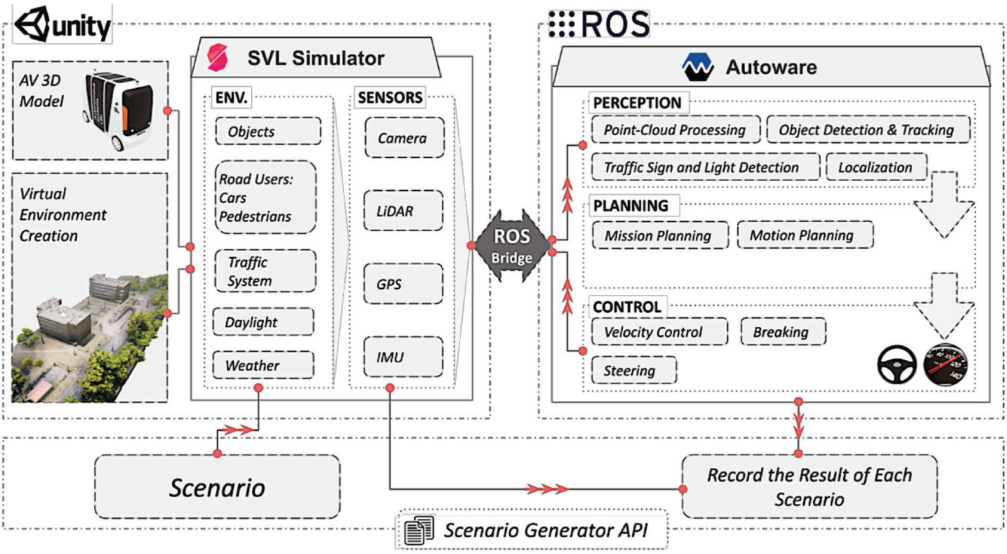

Fig. 2. High-level architecture of the simulation and the AV system.

tomizing the environment to create different scenarios such as adding/removing other road users, inserting traffic systems, adjusting the time of day and the weather of the scene. There is a scenario generator API that connects to the simulator and creates various scenarios according to the user definition. Then, the virtual sensors used in the AV provide information for the perception of the environment. This information is transferred via a ROS bridge to our control software platform to use in the perception algorithms for the localization and detection. Perception results are used in the Autoware planning section which makes the control commands for the AV. These control commands are sent back to the simulator via the ROS bridge to navigate the vehicle inside the simulator. Furthermore, in the case of any failure in any scenario, some sensor data and vehicle navigation commands are recorded for further study.

The iseAuto 3D model and its lidar sensors are illustrated in Fig. 3. A Velodyne VLP-32 was installed at the top front of the shuttle and a VLP-16 at the top back. Two Robosense Bpearl were installed at the left and right sides of the vehicle. Furthermore, to cover the blind zone in front of the vehicle, a RS-LiDAR-16 was installed in the front bumper. This lidar configuration creates a good point-cloud coverage around the vehicle for perception purposes.

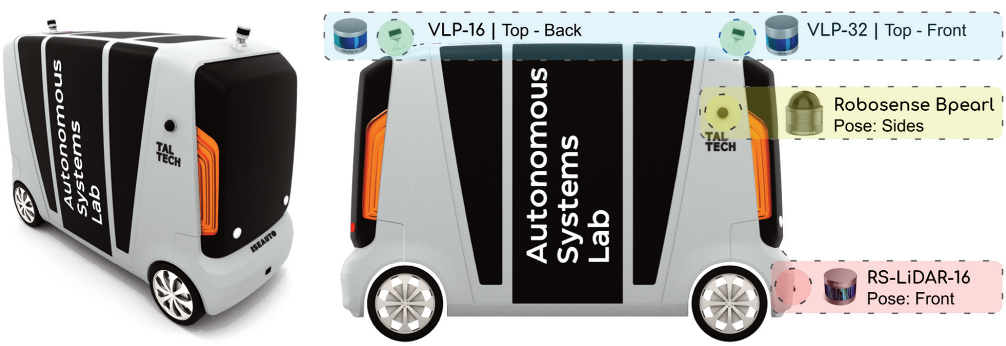

Fig. 3. iseAuto simulated model with different lidars installed.

3. VIRTUAL ENVIRONMENT CREATION

The fierce competition in the gaming industry nowadays has generated many features for game engines. These engines can simulate physics and thus be exploited as simulators aside from game development. SVL and others have already taken advantage of the aforesaid engines and created a framework for testing autonomous vehicles within such physics simulators. Even though these simulators provide some basic tools and assets to get started, it is still not sufficient. To make it more realistic, we need to have real-world terrains simulated.

3.1. Workflow

In order to create a terrain for simulation, the area to be simulated has to be mapped. There are certain steps to follow:

- Data Collection and Processing;
- Terrain Generation.

Data is collected by aerial photography and processed further to obtain a dense point-cloud of the area to be mapped. The point-cloud is then processed through a process called segmentation. Lastly, it is fed into Unity as an input for terrain generation.

3.2. Data collection and processing

Aerial imagery of the area to be mapped has to be captured with a camera drone. The images are captured at a grid flight path, which ensures that the captured images cover different sides of a subject. In order to make sure that the images have maximum coverage, the flight path is followed three times from different camera angles but at a constant altitude. Taking aerial photos is one of the most important steps in the mapping process as it will significantly affect the outcome of the process and the amount of work to be done to process those images. There are also external factors that may affect the quality of the pictures taken off the ground. Weather conditions and scene lighting may create artifacts on the pictures, which may disturb the photogrammetric process. The images taken are georeferenced by the drone and if necessary, a stationary Real Time Kinematic (RTK) device can be utilized to mitigate errors and shift the positioning data stamped on the pictures. The onboard IMU provides the pictures with orientation, so that later they can be stitched together and used for photogrammetric processing. Third party software aligns and creates the dense point-cloud from the pictures that were captured. Once the dense point-cloud is created, the segmentation and classification of the points is needed in order to separate unwanted objects and vegetation from the point-cloud data. However, removing is not to be performed in the point-cloud as the positional information they provide for their respective objects will aid terrain generation to spawn details. Figure 4 shows the three main steps to generate the Unity train from geospatial data.

3.3. Terrain generation

Digitalization of a real-life environment can be used for simulating AVs in countless different scenarios without taking the vehicle out for once. Terrain generation from point-cloud is performed right in Unity. In-house developed plugin reads a pre-classified point-cloud file, and based on chosen parameters it creates a normal map, a heightmap and a colour map to utilize in conjunction with the Unity's terrain engine to create realistic environments.

4. SIMULATION AND SAFETY ASSESSMENTS

Based on the simulation architecture illustrated in Fig. 2, the AV can be run inside the virtual environment. In collaboration with Florida Polytechnic University and Embry-Riddle Aeronautical University, we developed a regime for creating edge-case scenarios for safety validation of the shuttle working on our campus pilot road

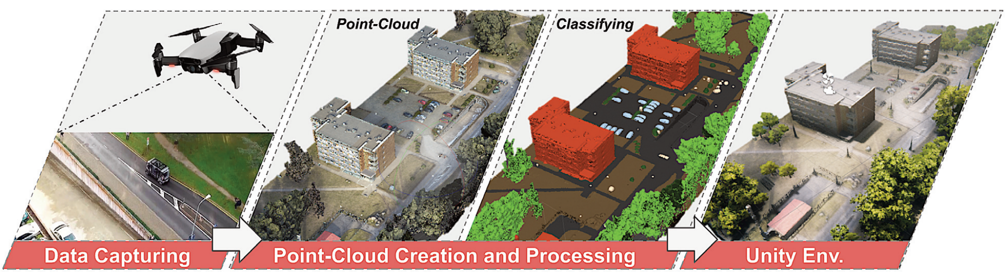

Fig. 4. Steps for virtual environment generation.

[20]. Now, by using a high-fidelity simulator we can simulate different scenarios close to real life in order to evaluate the control algorithm performance and safety. In terms of defining these scenarios, SVL provides a Python API for spawning different objects such as cars and pedestrians inside the virtual environment with different motion plans.

Figure 5 shows iseAuto facing a stopped Non-Player Character (NPC) vehicle that is spawned in front of the AV. Picture (a) is inside the SVL environment while picture (b) illustrates the lidar perception of the environment in RViz visualization tool. There is no filtering applied on this point-cloud; therefore, everything is mixed together and it is hard to distinguish objects for later processing. One of the challenging topics of self-driving development is overtaking. The way that the AV should decide for this mission and the risks that it faces are under study. Our experience with the vehicle trying to pass a stopped NPC or an object has led us to focus on this topic more. In this way, simulations can help first to improve our perception and detection system, and then to improve the mission and motion planning for a safe overtake. The first steps for detection are filtering and clustering the point-cloud. Autoware has some predefined features for them. One common point-cloud filtering is ground removal, in which some part of the point-cloud defined as ground will be separated. Each lidar point-cloud can be filtered separately or once after concatenation with other lidars. Filtering parameters have an intensive effect on the detection result. Sometimes losing 10 to 20 points due to the improper filtering will result in the object not to be detected.

Filtering and clustering are illustrated in Fig. 6. Filtering was applied to Fig. 5b. As a result, the ground, which can be seen in the figure, is almost removed from the point-cloud (see Fig. 6a). However, the NPC points remained and they were clustered as an object in Fig. 6b. Filtering accuracy results in high-performance object detection and safe decision making [21]. Figure 7 illustrates how different ground filtering parameters can change maximum distance for detecting a stopped NPC in front of the AV shuttle, although both cases have similar clustering parameters. Figure 7b shows that the NPC is

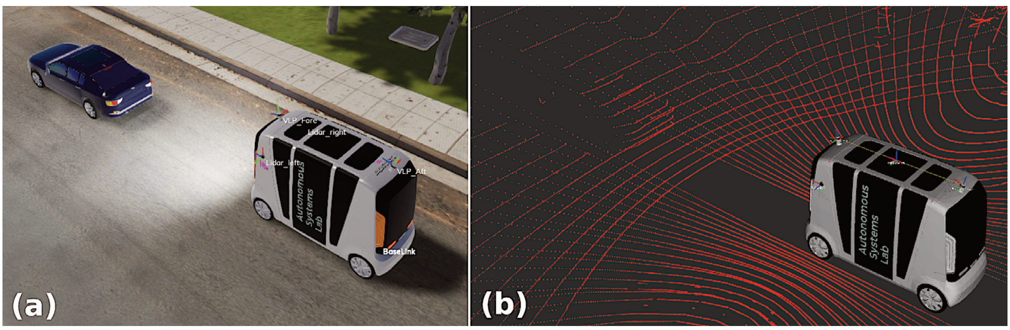

Fig. 5. (a) SVL environment versus (b) Rviz point-cloud visualization.

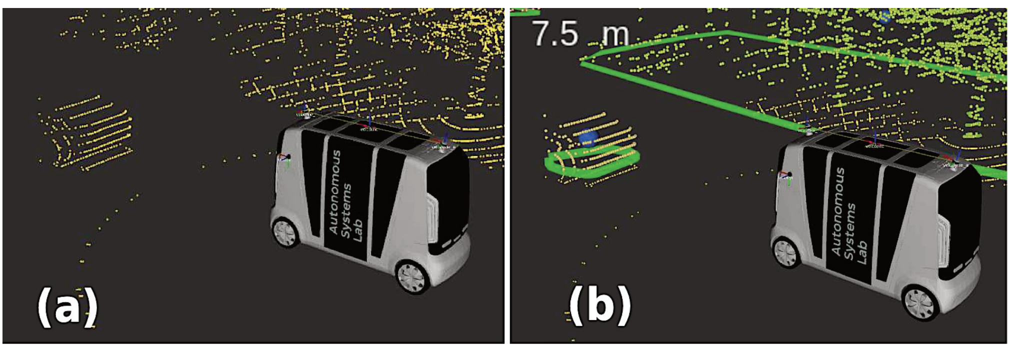

Fig. 6. (a) The ground filtering of the point-cloud and (b) applying of Euclidean clustering.

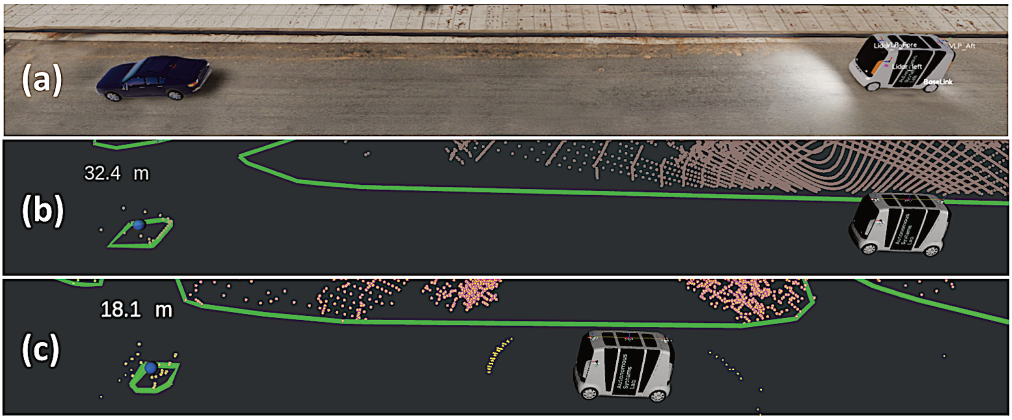

Fig. 7. Maximum distance for detecting a stopped NPC after filtration with different filtering parameters.

detected by the AV shuttle from the distance of 32 metres but picture (c) demonstrates that the maximum distance enabling to detect an object has decreased to 18 metres. The more distance we have for detection, the more time we have for making a smooth control decision. In AVs with multiple lidars, filtering accuracy can be improved by performing it before point-cloud concatenation.

4.1. Scenario definition

Scenarios are plans for studying simulations effectively. A good scenario generator can help to validate the whole control system faster in a more reliable way, guaranteeing to cover all the corner cases that might cause failure in the system. There are several methods for generating the scenarios such as human designed, grid search and optimized searching. For example, in [22], the authors implemented a learning method to find safety-critical scenarios for specific tasks. In this paper, for showing the simulation workflow, two main and simple overtaking scenarios were studied. Figure 8 demonstrates two different situations in overtaking: scenario A shows a stopped car that is overtaken by our shuttle while scenario B shows the same mission with an additional car, already starting to overtake the two others.

4.2. Running simulation

In this section, the two described scenarios are simulated inside the simulator and shuttle behaviour is monitored.

- Scenario A

In this scenario, the shuttle is passing a stopped vehicle by generating an alternative local waypoint. The overtaking algorithm is enabled after the shuttle has detected an obstacle in its path. Five different frames of this scenario simulation are shown in Fig. 9. First, the AV follows the way and detects the obstacle (step 1), then stops 15 metres before the object (step 2) and generates a new waypoint (step 3). Then, it starts to follow the new waypoint, and finally, after passing the obstacle, it changes the lane back to the initial path (step 4) and continues its former route (step 5).

By simulating scenario A several times in different areas, the overtaking algorithm for passing a static object was initially evaluated and verified. But to investigate more challenging situations, various road users such as other vehicles and pedestrians should be involved in the scenario. For this, another scenario was designed by adding another vehicle driving forward from behind in the opposite lane.

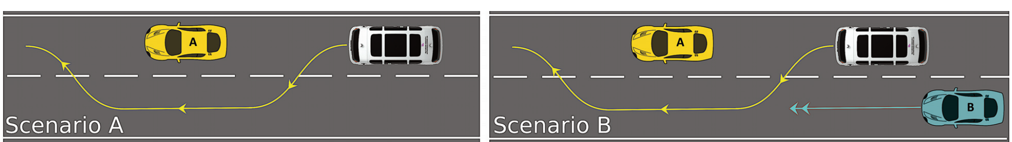

Fig. 8. Two different scenarios for overtaking.

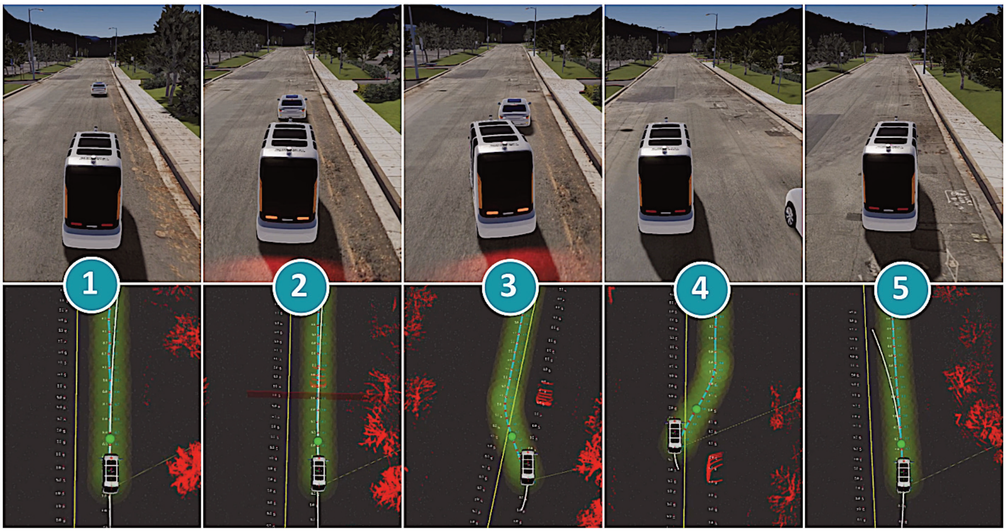

Fig. 9. Five different steps of the scenario A simulation in the SVL simulator (top) and in the Rviz (bottom).

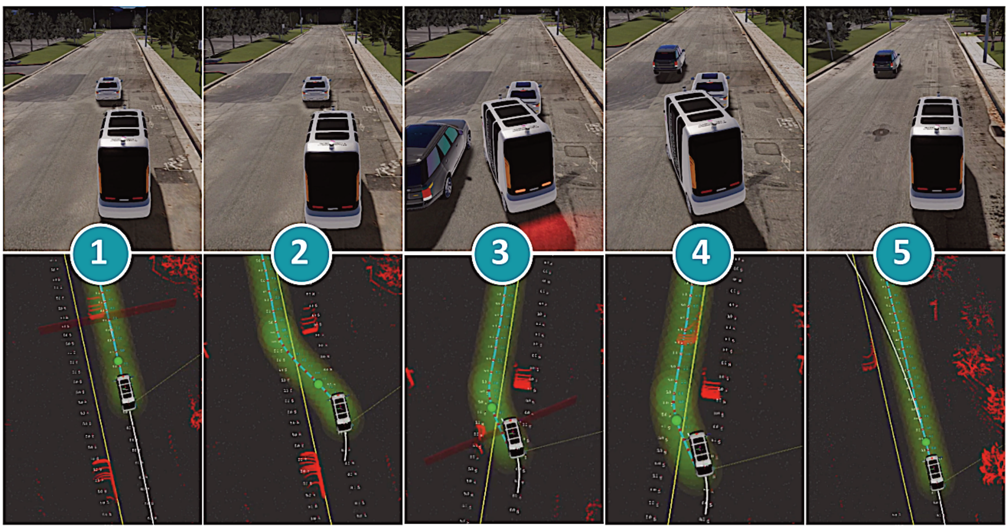

Fig. 10. Different steps of an overtaking process.

- Scenario B

Figure 8 shows the scenario B scheme, that a third vehicle is overtaking the shuttle and the stopped vehicle. It is expected that the shuttle prevents collision and considers the opposite lane traffic. Similar to the former scenario, five steps of scenario B are recorded in Fig. 10. As seen in the simulation, the AV reaches the static object and stops to prepare for overtaking (step 1). The moving vehicle is visible in the Rviz software (frame 1 image below) as a red point-cloud cluster. It is expected that the shuttle prevents collision and considers the opposite lane traffic while overtaking. In step 2 the shuttle starts to overtake and the new path is generated. Before the shuttle changes the lane, it meets the moving vehicle in the green area (collision area, any object inside it is an obstacle), then the shuttle stops before the collision happens. Finally, after the moving vehicle drives more than 15 metres along the green area, the shuttle starts to follow the route and changes the lane back to its initial path.

This scenario was simulated with a different value for variables such as the speed of the moving vehicle and the lateral position of each vehicle on the road. The results recorded collision in some cases and investigations showed that due to the limited size of the green area and lack of an efficient motion prediction while shifting lanes, the AV can collide with other road users that are not considered. Therefore, using the current overtaking algorithm without any added prediction feature is rejected and it is not safe to be implemented in the real shuttle.

5. CONCLUSIONS

Safety validation is crucial for most of the AV developments and deployments. The simulation as a validation approach presented in this paper offers a practical and effective way to evaluate the safety in different levels. This paper provides the simulation architecture of iseAuto with SIL testing, which shows how the virtual environment and vehicle model are used in combination with Autoware to simulate different scenarios. As an illustration, two overtaking scenarios were studied and the control algorithm was examined based on its safe performance. In conclusion, the development and utilization of this testing scheme will enable the development of safety improvement and autonomous vehicle performance.

ACKNOWLEDGEMENTS

This research received funding from two grants: the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No. 2014-2020.4.01.20-0289. The publication costs of this article were covered by the Estonian Academy of Sciences and Tallinn University of Technology.

REFERENCES

1. Kalra, N. and Paddock, S. M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract.*, 2016, **94**, 182–193.
2. Chao, Q., Jin, X., Huang, H. W., Foong, S., Yu, L. F. and Yeung, S. K. Force-based heterogeneous traffic simulation for autonomous vehicle testing. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 20–24, 2019*. IEEE, 8298–8304.
3. Aeberhard, M., Rauch, S., Bahram, M., Tanzmeister, G., Thomas, J., Pilat, Y., Homm, F., Huber, W. and Kaempchen, N. Experience, results and lessons learned from automated driving on Germany's highways. *IEEE Intell. Transp. Syst. Mag.*, 2015, **7**(1), 42–57.
4. Anderson, S. J., Peters, S. C., Pilutti, T. E. and Iagnemma, K. Design and development of an optimal-control-based framework for trajectory planning threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In *Robotics Research* (Pradalier, C., Siegwart, R. and Hirzinger, G., eds). Springer, Berlin, Heidelberg, 2011, 39–54.
5. Razdan, R., Lumina, J., Balachandran, A., Cheng, C., Sreenivas, S., Fernando, X., Taiber, J., Kalia, A., Keel, N., Zuby, D., Krishnan, K., Langer, D. and Sell, R. Unsettled technology areas in autonomous vehicle test and validation. SAE Technical Paper Series, 2019.
6. Malayjerdi, M., Kuts, V., Sell, R., Otto, T. and Baykara, B. C. Virtual simulations environment development for autonomous vehicles interaction. In *Proceedings of the ASME 2020 International Mechanical Engineering Congress and Exposition, Portland, OR, USA, November 16–19, 2020.* ASME, **2B**, IMECE2020-23362.
7. Kuts, V., Modoni, G. E., Otto, T., Sacco, M., Tähemaa, T., Bondarenko, Y. and Wang, R. Synchronizing physical factory and its digital twin through an IioT middleware: a case study. *Proc. Est. Acad. Sci.*, 2019, **68**(4), 364–370.
8. Kuts, V., Cherezova, N., Sarkans, M. and Otto, T. Digital Twin: industrial robot kinematic model integration to the virtual reality environment. *J. Mach. Eng.*, 2020, **20**(2), 53–64.
9. Lu, B., He, H., Yu, H., Wang, H., Li, G., Shi, M. and Cao, D. Hybrid path planning combining potential field with sigmoid curve for autonomous driving. *Sensors*, 2020, **20**(24), 7197.
10. Andersen, H., Schwarting, W., Naser, F., Eng, Y. H., Ang, M. H., Rus, D. and Alonso-Mora, J. Trajectory optimization for autonomous overtaking with visibility maximization. In *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, October 16–19, 2017*. IEEE, 1–8.
11. Easa, S. M. and Diachuk, M. Optimal speed plan for the overtaking of autonomous vehicles on two-lane highways. *Infrastructures*, 2020, **5**(5), 44.
12. Sell, R., Leier, M., Rassõlkin, A. and Ernits, J. Self-driving car ISEAUTO for research and education. In *Proceedings of the 2018 19th International Conference on Research and Education in Mechatronics (REM), Delft, Netherlands, June 7–8, 2018*. IEEE, 111–116.
13. Christophe, F., Sell, R., Bernard, A. and Coatanéa, E. Opas: Ontology processing for assisted synthesis of conceptual design solutions. In *Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering* Conference*, San Diego, USA, August 30–September 2, 2009*. ASME, **49026**, 249–260.

14. Sell, R., Coatanéa, E. and Christophe, F. Important aspects of early design in mechatronic. In *Proceedings of the 6th International Conference of DAAAM Baltic Industrial Engineering, Tallinn, Estonia, April 24–26, 2008*, 177–182.
15. Wang, R., Sell, R., Rassõlkin, A., Otto, T. and Malayjerdi, E. Intelligent functions development on autonomous electric vehicle platform. *J. Mach. Eng.*, 2020, **20**(2), 114–125.
16. Rassõlkin. A., Sell, R. and Leier, M. Development case study of the first estonian self-driving car, iseauto. *Electr. Control Commun. Eng.*, 2018, **14**, 81–88.
17. Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y. and Azumi, T. Autoware on board: Enabling autonomous vehicles with embedded systems. In *Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, April 11–13, 2018*. IEEE, 287–296.
18. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. and Koltun, V. Carla: An open urban driving simulator. 2017, *arXiv:1711.03938*.
19. Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S. et al. LGSVL simulator: A high fidelity simulator for autonomous driving. 2020, *arXiv:2005.03778*.
20. Medrano-Berumen, C., Malayjerdi, M., Akbaş, M. I., Sell, R. and Razdan, R. Development of a validation regime for an autonomous campus shuttle. In *Proceedings of the 2020 SoutheastCon, Raleigh, NC, USA, March 28–29, 2020*. IEEE, 1–8.
21. Habermann, D., Hata, A., Wolf, D. and Osório, F. S. 3d point clouds segmentation for autonomous ground vehicle. In *Proceedings of the 2013 III Brazilian Symposium on Computing Systems Engineering, Rio de Janeiro, Brazil, December 4–8, 2013*. IEEE, 143–148.
22. Ding, W., Chen, B., Xu, M. and Zhao, D. Learning to collide: An adaptive safety-critical scenarios generating method. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, October 25–29, 2020.* IEEE, 2243–2250.

Autonoomse sõiduki turvalisuse hindamise suure täpsusega simulatsiooni meetod

Mohsen Malayjerdi, Barış Cem Baykara, Raivo Sell ja Ehsan Malayjerdi

Autonoomsete sõidukite tööstus planeerib strateegilisi lahendusi, et kindlustada turvalisus enne, kui autonoomsed sõidukid viiakse masstootmisse. Turvalisuse saavutamiseks on vajalik läbi viia väga erinevaid teste. Kõikide testide tegemine reaalse sõidukiga reaalses linnaruumis on pigem ebapraktiline ja võtaks aega aastaid. Selle probleemi vältimiseks kasutatakse simulatsioone. Antud artikli eesmärgiks on välja pakkuda metoodika ja tehnoloogia turvalisuse valideerimise simulatsioonideks autonoomsete sõidukite testimisel. Artiklis on välja pakutud turvalisuse hindamise meetod, mis on realiseeritud TalTechi linnakus tegutseva TalTechi iseauto autonoomse sõiduki platvormil. On loodud virtuaalne mudel linnaku testalast, mis sisaldab eri objekte ja mis on konverteeritud 3D-kaardiks Unity keskkonnas. Loodud virtuaalne mudel on omakorda sisendiks SVL-simulaatorile, mis ühendab endas virtuaalsete andurite simulatsiooni ning Autoware algoritmid, mis juhivad TalTechi iseautot. Demonstratsioonlahendusena on kirjeldatud simulatsioonijuhtu, kui isejuhtiv sõiduk peab tegema möödasõidu seisvast autost, mis blokeerib sõidurea. Lõpuks on näidatud, kuidas antud lahendus võimaldab hinnata isejuhtiva sõiduki otsuste tegemise võimekust ja turvalisust eri situatsioonides.

Paper IV

Mohsen Malayjerdi, Andrew Roberts, Olaf manual Maennel, and Ehsan Malayjerdi. "Combined Safety and Cybersecurity Testing Methodology for Autonomous Driving Algorithms." In Proceedings of the 6th ACM Computer Science in Cars Symposium, pp. 1-10, 2022.

Combined Safety and Cybersecurity Testing Methodology for Autonomous Driving Algorithms

Mohsen Malayjerdi
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Tallinn, Estonia
mohsen.malayjerdi@taltech.ee

Andrew Roberts
FinEst Centre for Smart Cities
Tallinn University of Technology
Tallinn, Estonia
andrew.roberts@taltech.ee

Olaf Maennel
Centre for Digital Forensics and Cybersecurity
Tallinn University of Technology
Tallinn, Estonia
olaf.maennel@taltech.ee

Ehsan Malayjerdi
Department of Mechanical and Industrial Engineering,
Tallinn University of Technology
Tallinn, Estonia
ehsan.malayjerdi@taltech.ee

ABSTRACT

Combined safety and cybersecurity testing are critical for assessing the reliability and optimisation of autonomous driving (AD) algorithms. However, safety and cybersecurity testing is often conducted in isolation, leading to a lack of evaluation of the complex system-of-system interactions which impact the reliability and optimisation of the AD algorithm. Concurrently, practical limitations of testing include resource usage and time. This paper proposes a methodology for combined safety and cybersecurity testing and applies it to a real-world AV shuttle using digital twin, software-in-the-loop (SiL) simulation and a real-world Autonomous Vehicle (AV) test environment. The results of the safety and cybersecurity tests and feedback from the AD algorithm designers demonstrate that the methodology developed is useful for assessing the reliability and optimisation of an AD algorithm in the development phase. Furthermore, from the observed system-of-system interactions, key relationships such as speed and attack parameters can be used to optimise testing.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

automotive cybersecurity, safety testing, autonomous driving

ACM Reference Format:

Mohsen Malayjerdi, Andrew Roberts, Olaf Maennel, and Ehsan Malayjerdi. 2022. Combined Safety and Cybersecurity Testing Methodology for Autonomous Driving Algorithms. In *Computer Science in Cars Symposium (CSCS '22), December 8, 2022, Ingolstadt, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3568160.3570235

CSCS '22, December 8, 2022, Ingolstadt, Germany

ACM ISBN 978-1-4503-9786-5/22/12…$15.00
https://doi.org/10.1145/3568160.3570235

1 INTRODUCTION

Testing autonomous driving (AD) algorithms for performance under safety test cases is a predominant focus for developers to assess the reliability of the algorithm and for optimisation. AD algorithms are also susceptible to manipulation from cyber threats which target the advanced hardware technologies sensor telemetry which serves as an essential input for perception, detection, and control decisions [2, 12, 20]. Existing methods [3, 8] for testing are challenged by the complexity of evaluating system-of-system interactions to identify key relationships and parameters, and limitations of testing inherent to real-world AV programs, resource usage and time. The main idea of this paper is to establish a method for combined safety and cybersecurity testing of developmental AD algorithms to evaluate system-of-system interactions to identify and investigate parameters that impact safety and the effect of cyber attacks, and to develop future ideas for optimisation of testing. To this end, the paper focuses on three research questions aligned with the challenges of combined safety and cybersecurity for AD algorithms.

- RQ1 How can AD algorithm designers evaluate the reliability and optimisation of the AD algorithm to both safety and cybersecurity test cases?
- RQ2 Cybersecurity testing is predominantly conducted on well-established AD algorithms. How can combined safety and cybersecurity testing be conducted on a developing AD algorithm?
- RQ3 What key relations and parameters can we identify that can optimise safety and cybersecurity testing?

To evaluate these research questions, we apply our methodology to a developing AD algorithm in a digital twin, software-in-the-loop (SiL) simulator and real-world AV testing environment. Cybersecurity testing and safety testing are often conducted separately, reducing our understanding of the relationship between failures of the algorithm caused under normal safety scenarios and failures caused by the impact of cyber attacks. For AD algorithms in the development stage, where the reliability and optimisation of the AD algorithm to safety scenarios have not been established, this exploration of the relationship between safety and cybersecurity can offer novel insights to improve the awareness of the AD algorithm designer to shortcomings in the algorithm.

The major contributions of this paper are the following:

- Methodology for combined safety and cybersecurity testing
- Safety and cybersecurity test cases conducted on an AD algorithm under development, and with feedback from the AD algorithm designer
- An analysis of the combined safety and cybersecurity test cases that identifies key relations and the sensitivity of parameters.
- All the code, our AV simulation configurations and research data used in the combined safety and security testing will be available for the research community on GitHub.

2 TARGET SYSTEM

2.1 Low-Speed AV Shuttle for Public Transportation

The target AV for this study, iseAuto (see Fig. 1), is a real-world AV shuttle for public transportation, operating in numerous EU countries.

Figure 1: iseAuto autonomous shuttle

The shuttle was developed as part of a project at Tallinn University of Technology's AV research group. The objective of this project is to build an open-source AV shuttle that provides a smart city test bed within the university campus, enabling different types of urban mobility research. Currently, this SAE level 4 and 5 shuttle is operating on the campus for experimental and study purposes. iseAuto uses a multi-LiDAR sensor system for perception and localisation. Two Velodyne LiDARs are mounted at the top front (VLP-32) and the 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.

2.2 Autonomous Driving Algorithm

The AV uses Autoware.ai [11] autonomous software stack which is an open-source AD software. This software enables us to employ different algorithms for each main part of the autonomous system including localization, sensing, detection, and navigation. Open-Planner navigation planning algorithm.

In this study, we focused on OpenPlanner as one of the most widely used path-planner modules in the AD software. In the latest version of this algorithm, which is currently 2.5, the module has become noticeably more advanced in terms of supporting various high-definition map formats, predicting the trajectories of other actors, and using a kinematics-based trajectory generator [5]. This version is compatible with Autoware.ai 1.15. Open-planner combines global and local planners that jointly utilize the road network map to generate local waypoints based on a global route and manage discrete behaviours such as avoiding dynamic obstacles and following traffic lights.

The local planner module generates tracks parallel to the main path defined by the global planner. These tracks are named rollouts (see Fig. 2). The trajectory evaluator assesses all possible rollouts in case an obstacle blocks the path. Then, the behaviour selector will lead the AV to the new safe rollout. Figure 2 shows how open-planner selected rollout number 6 in order to pass the non-player character (NPC). It also detects the curb lines and avoids those rollouts which intersect the curbs.

The algorithm uses the output of the kf_contour_track algorithms to consider all the perceived objects based on the LiDARs point cloud in its local path planning. Earlier, the euclidean clustering algorithm received the filtered point cloud data and prepared point clusters, which is the input of the kf_contour_track. This combination of cluster and contour tracking is done in each sequence for the open-planner to evaluate possible trajectories and create the behaviour based on that. Figure 3 shows the diagram of how the open-planner module works under the AD software package.

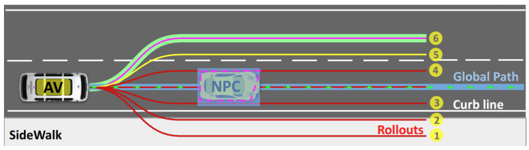

Figure 2: How open-planner generates different trajectory to pass an object

3 COMBINED SAFETY AND CYBERSECURITY TESTING METHODOLOGY FOR AD ALGORITHMS

The architecture of the proposed combined testing methodology is presented in Figure 3. This method takes advantage of a high-fidelity software in the loop (SiL) simulation [16] approach to validate and verify the performance of a AD software under critical cyber security conditions. This method consists of three main following elements:

- Attack script: which simulates a critical security condition.
- High-fidelity simulator: It is a game engine environment that provides the physics for modeling sensors and motion.
- AD software: It is the autonomous driving software that controls the AV.

The combined safety and cybersecurity methodology consisted of the following iterative steps:

- **Scenario Selection**
- **Analysis of the scenario to extrapolate the safety evaluation criterion applicable**
- **Safety Test Case Setup**
  - Initialisation of the SiL high-fidelity simulator and configuration to the real-world AV

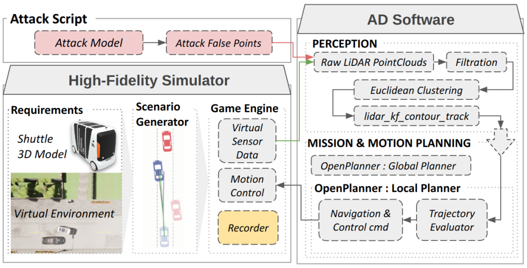

Figure 3: Architecture of the testing platform

  - Initial scenario testing using the safety test cases to assess the reliability of the algorithm and the quality of the test data
  - Optimisation of the safety test cases to select a subset of the scenario tests to assess the reliability of the algorithm
  - Run of the safety test case scenarios
  - Selection of distinct safety test case scenarios which provide most stable results in terms of success of mission and safety violation
- **Cybersecurity Test Case Setup**
  - Analysis of the scenario to determine cyber attack strategy for test cases
  - Development of the code for adversary generation in the SITL high-fidelity simulator
  - Selection of attack parameters
  - Optimised the cybersecurity test cases
  - Evaluate cybersecurity test cases in SiL high-fidelity simulator
  - Real-World AV Testing for safety and cybersecurity
- **Results Analysis**
  - Analysis of the performance of AD algorithm to safety criteria
  - Analysis of sensitivity of attack parameters and driving parameters

3.1 Testing Environment

All tests are conducted in a virtual environment powered by the "Unreal game engine" (Unreal) [4]. Carla simulator [6] is one of the open-source high-fidelity vehicle simulators capable of connecting to different AD software and scenario generator applications. In this study, we use Carla 0.9.13 as the high-fidelity simulator. Figure 3 illustrates the requirements for the high-fidelity simulator to conduct simulation testing which are two components, the digital twin of our AV and the virtual replication of our target environment. These replicated components help us to gain more accurate results of the proposed platform [14]. The AV digital twin is a 3D model of our real-world world AV shuttle, designed in Blender, a graphical 3d modelling software, and imported and built in Unreal for deployment in Carla. This model uses the same dimension and sensor configuration (model, position, and orientation) from the real AV shuttle. The environment digital twin, in our case, is identical to the location where we are testing and operating our shuttle, this includes the urban details and vegetation. The next module in the simulator is a scenario generator that produces the desired scenario based on the user input specification. Finally, the simulator engine generates sensor data from sensors, including LiDARs, cameras and others and publishes it for other blocks (see Fig. 3 the simulator block). Then, the AD software receives this data as raw LiDAR point-cloud information and processes the data as mentioned in the diagram (Figure 3).

This simulation setup was implemented on a desktop computer with the following configuration:

- Intel® Core™ i7-11700K @ 3.60GHz × 16 cores
- NVIDIA GeForce RTX 3080 10 GB
- RAM: 128 GB

3.2 Scenario Selection

To evaluate the combined safety and cybersecurity testing, we chose a simple overtaking maneuver, which is one of the most safety challenging operations [13]. Figure 4 shows the functional level of the planned scenario. To generate a variety of distinct scenarios, we opt for the initial relative distance to the NPC D_x and the NPC constant speed S_{NPC} as the distinct scenario parameters.

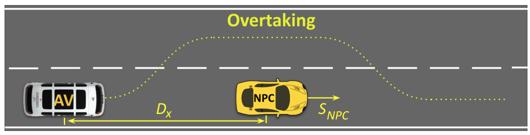

Figure 4: Dx **and** S_{NPC}**, define the initial relative distance to the NPC and the constant NPC speed in each scenario**

Table 1: Target scenarios definition

Actor	Speed	D_x	Goal
AV	[0:6]m/s	0 (m)	overtake the NPC safely
NPC	[1 1.4 1.8 2.1 2.5]	[15 20 25](m)	keep moving

3.3 Safety Evaluation Criteria

In determining the evaluation criteria for AV safety we considered two conditions, 1) mission success and 2) safety violations. A safety violation consists of a collision and dangerous driving behaviour. In determining which criteria to apply, we considered the EuroNCAP [1] and ISO26262 [10] standards as well those used in composite studies [3, 7, 8]. We derived that the safety goal of the AD algorithm is to execute the overtaking mission without colliding or interfering with other ego vehicles or objects and without exhibiting driving behaviour which is dangerous to the AV passengers. Table 2 details the safety criteria applied in our experiments.

3.4 Safety Test Case Setup

To evaluate the reliability and optimisation of the AD algorithm for the overtaking manoeuvre, we, firstly, initiated a run of 50 distinct scenarios in the high-fidelity simulator, repeating 6 times. Each scenario was repeated 6 times to ensure the reproducibility

Table 2: Safety Evaluation Criteria

Safety Condition	Data Label	Description	Metric
Succeed	Suce	AV Successful complete the mission	Pass/Fail
Not Finished	NotF	Failure to finish the mission	Pass/Fail
Distance-to -Collision	DTC	Violation of the safe distance between AV and NPC	AV within 0.5m of other vehicle
Break on Driving Lane	BrD	AV initiates emergency break on driving lane	Pass/Fail
Break on Passing Lane	BrP	AV initiates emergency break on passing lane	Pass/Fail
Collision	Col	AV collides with NPC	Pass/Fail
Violation	V	Safety Violation	

of the outcome. With the mentioned desktop configuration, it took approximately 100 *sec* for each scenario and, in total, 8.3 hours for 300 runs. The purpose of the first scenario run was to provide a general overview of the performance of the algorithm. We targeted a range of 1 to 3 m/s for the NPC speed and 15 to 30 m for the initial relative distance to the NPC for selecting the 50 distinct scenario parameters. The results showed that the AD algorithm could not safely overtake the NPC at an NPC speed higher than 2.5 m/s and a distance (D_x) of more than 25 m.

Although a high number of scenario variations shows better coverage in the scenario space to find corner cases, it will lead to an increase in the time duration of the runs. Furthermore, the number of each scenario repetitions was not sufficient to statistically explain the occurrence of each safety violation. Finally, it is worth mentioning that, as our primary study focus is not just the validation of the AV performance, we need to use an optimum number of trials for both safety and cyber test cases. Due to this, we limited the scenario parameters space to the intervals listed in Table 1 that regressed the test set to 15 distinct cases in a full factorial setup. This enabled us to repeat the simulation of these test cases 50 times and apply the full set of safety criteria: collision, DTC, break in passing lane, break in driving lane, failure to finish, and mission success.

Each scenario is generated by the Carla scenario runner utilizing the Python behaviour trees to handle series and parallel events in the scenario. Figure 5 depicts the scenario scheme starting with the main sequence behaviour. This series begins with transforming the actors into the environment and finishes by destroying the actor block. A parallel behaviour (Driving Toward Intersection) is defined to run the attack and the scenario stop block while the NPC follows the defined waypoint. For safety test case scenarios, the attack block is skipped, and the scenario waits till the stop criteria are satisfied.

3.5 Cyber Test Case Setup

To determine the cyber attack strategy for implementation in this test scenario, we analysed the overtaking scenario and its applicability to state-of-the-art attacks on AD algorithms. We selected

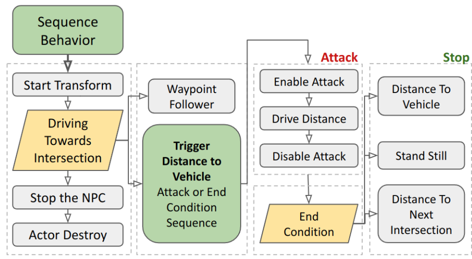

Figure 5: Flow-graph of how each scenario is processed in the simulation platform

LiDAR spoofing as it is a realistic attack in the driving environment of our real-world AV shuttle [3] and its impact is relevant to safety outcomes due to the likelihood that the manipulated driving behaviour will result in collisions, emergency breaking, and lane violations [20]. Attacks on LiDAR perception predominantly focus on spoofing LiDAR 3D point-clouds through the following means: 1) injection of adversarial LiDAR 3D point cloud data to add adversarial objects to the driving environment inducing a *false positive result* of the AD perception [3, 17] 2) removal of LiDAR 3D point cloud data to perturb the ability of the perception algorithm to detect objects in the driving environment, also known as a *false negative result* [8, 9] 3) manipulating LiDAR 3D point cloud data to obfuscate the true distance of environmental objects (Other road vehicles, pedestrians, other road objects) from the AV, causing the perception to *fail translation* 4) implementation of adversarial mesh in the driving environment to introduce manipulated points into the LiDAR 3D point cloud and create unpredictable perception events [19]. The aim of the attacker, in adversarial LiDAR threat models, is to induce the victim AV to perform dangerous driving maneuvers, which include; emergency breaking, collisions, and exceeding the limits of the driving lanes. Variables that have been shown to influence attack success include; angle of attack of the adversarial point cloud vector, density of the spoofed points, duration of the broadcast of spoofed points, distance of the point cloud to the target [3, 8, 17, 20]. We implemented a variation of the attack suggested by Yang et al. [20], where the adversary creates an adversarial roadside object to inject spoofed, malicious LiDAR point clouds into the target AV LiDAR. In our attack, an adversary has configured a LiDAR on the roadside to inject malicious point cloud data into the AV as it is conducting the overtaking manoeuvre. Figure 6 demonstrates the implementation of our attack.

Using the knowledge gained from literature [8, 17, 20], the parameters we chose to generate our attack are: density of the LiDAR point clouds, frequency (the publishing rate of the fake points), duration of the adversarial point cloud broadcast, and location, which is the relative location between the target vehicle and NPC. As an infinite number in the range of each of the parameters can be chosen, we decided to limit our testing to parameter values that had demonstrated utility to investigate the impact of cyberattacks on AD algorithms. For example, Hallyburton et al. [8] found that the success of cyber attacks increased when spoofed point density were over 80. Therefore we chose a range for spoof point density from 50 to 300.

3.5.1 Taguchi Analysis. In this study, we use the Taguchi method for statistical evaluation [18] of the attack parameters effect on each safety criterion. The number of tests with four parameters and 3 levels for each in full factorial mode would become unrealistic to perform, noting that each experiment should repeat 50 times (81x50 = 4050 distinct scenarios). A design of the experiment is recommended in order to avoid full factorial tests and reduce the number of tests without compromising accuracy [18].

A Taguchi design of experiment (DOE) technique [18] was applied to quantify the influence of four proposed attack parameters; the false points (FP) density, the FP frequency, the attack duration, and the attack location. In total, 9 experiments were designed with 3 different values for the four parameters. The analyses hence possess four factors and three levels for the Taguchi L9 matrix. Table 3 lists the configuration for each run conducted for cybersecurity tests.

Table 3: Taguchi L'9 matrix for study of factor influence

Num.	*Density*	*Frequency*	*Duration*	*Location*
1	50	5	3	3
2	50	7	6	6
3	50	10	9	9
4	150	5	6	9
5	150	7	9	3
6	150	10	3	6
7	300	5	9	6
8	300	7	3	9
9	300	10	6	3
	[50 150 300]	[5 7 10]	[3 6 9]	[3 6 9]

Figure 6 demonstrates the cyber attack setup within the overtaking scenario (Please note, the Figure only depicts the overtaking frame and not the entire overtaking sequence.). The proposed attack model will start by generating spoof points from the designated place on the roadside. At the starting point, P_1, the AV has relative distance to NPC that defines the attack location. After a specific duration (Attack Duration), the AV reaches, P_2. While the attacker keeps the malicious LiDAR pointing toward the AVs front LiDAR. Overall, the spoofed point direction changes from θ_1 to θ_2.

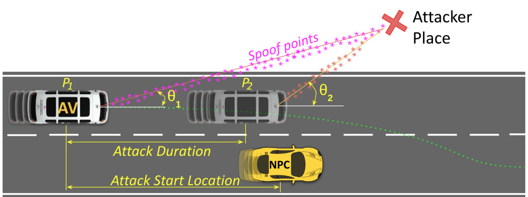

Figure 6: Attack scheme

Code was created for the generation of the adversarial LiDAR fake points to be run in the digital twin, high-fidelity simulation environment. This is available on the GitHub site [15].

4 RESULTS AND ANALYSIS

In this section, we present the results of the safety and cybersecurity testing of the end-to-end AD algorithm. The purpose of the safety test case results is to evaluate the reliability and optimisation of the algorithm.

4.1 Safety Test Case

The aim of the testing is to assess the utility of the methodology to evaluate the relationship between the reliability of the AD algorithm to safety and the impact of cybersecurity. As the testing is based on a real-world AV, we were motivated to establish what results could be gained from an amount of tests that took into account the requirements for CPU and GPU resources and the time involved in running high-fidelity simulations. For instance, 50 distinct scenarios run 3 times expends x amount of resources, and takes x amount of time. Therefore, we, firstly, performed a baseline evaluation test where we ran 50 distinct scenarios of the overtaking manoeuvre, 3 times. Each scenario is distinct based on changes to parameters such as NPC speed and initial distance to NPC.

In our proposed simulation platform, we perform 15 distinct scenarios, run 50 times; in total, 750 consecutive simulation runs were conducted. Table 4 shows the parameters of the distinct scenarios evaluated against the safety criteria. Using our configuration for testing, the AD algorithm shows the performance for the overtaking manoeuvre with a success rate of 43.9% of the simulated scenarios, whilst, 66.1% are safety violations.

In Figure 7 is the performance of the AD algorithm.

Table 4: Summary of the safety simulation

	D_x	S_{NPC}	V_{Col}	V_{DTC}	V_{BrP}	V_{BrD}	V_{NotF}	V_{Suce}
1	15	1	18%	22%	0%	10%	24%	26%
2	20	1	18%	40%	8%	6%	18%	10%
3	25	1	4%	20%	32%	8%	20%	16%
4	15	1.4	6%	32%	16%	2%	12%	32%
5	20	1.4	22%	26%	14%	6%	2%	30%
6	25	1.4	4%	12%	22%	8%	0%	54%
7	15	1.8	36%	34%	8%	2%	6%	14%
8	20	1.8	22%	12%	2%	2%	0%	62%
9	25	1.8	18%	6%	0%	4%	0%	72%
10	15	2.1	4%	0%	4%	2%	4%	86%
11	20	2.1	8%	10%	0%	0%	0%	82%
12	25	2.1	24%	0%	0%	4%	0%	72%
13	15	2.5	14%	6%	0%	6%	2%	72%
14	20	2.5	44%	22%	14%	0%	2%	18%
15	25	2.5	64%	18%	0%	0%	6%	12%
		mean	20.4%	17.3%	8.0%	4.0%	6.4%	43.9%
		STD	16.8%	2.3%	9.8%	3.2%	8.1%	28.3%
		min	4%	0%	0%	0%	0%	10%
		max	64%	40%	32%	10%	24%	86%

NPC speed is an important parameter as it influences the decision control for the critical cut-in manoeuvre of the overtaking mission. In the context of the results of the simulations, we can see that NPC speed impacts certain safety criteria.

The first such relation that can be seen, is that more collisions are caused at high speeds, > 2.1 m/s. This can be the effect of a poor trajectory evaluator that doesn't consider the prediction of the other actors motions in the process of the waypoint generation. In

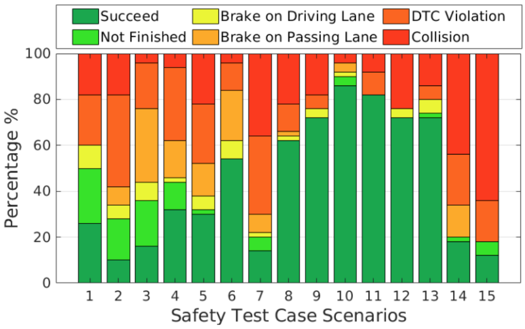

Figure 7: The 15 distinct scenarios

most collision cases the AV tried to perform a cut-in while the NPC collided from the right side. The probability of this safety violation will be increased as the NPC speed increases.

NPC speed also impacts the likelihood of a DTC safety violation. In the range of the NPC speed parameter, 1 m/s to 1.8 m/s, it can be observed that AV Shuttle violates the safe distance to the NPC. This can be due to the AV speed adjusting relative to the NPC speed and the cut-in is attempted at low-speed, whilst acceleration is required to safely attempt the cut-in. This low-speed cut-in firstly causes a DTC violation and if the overtaking manoeuvre progresses it causes a collision. DTC and collision correlate based on the relative speed. A low-speed NPC will likely result in a DTC violation, whilst in a higher-speed scenario, a collision is more likely to happen.

In the lowest speed range, 1 m/s to 1.4 m/s, it is more likely that the AV will initiate an emergency break in the passing lane. This is due to the relationship of the NPC speed to the AV Shuttle speed. The emergency break on the passing lane at low speeds is caused by a failure of the open-planner trajectory evaluator to effectively plan the overtaking trajectory. Figure 8 demonstrates the AV emergency break in the passing lane, for a scenario with an NPC Speed of 1 m/s. The upper rectangle represents the AV and the lower rectangle is the NPC. The two rectangles closest to the left represent the frame that the first emergency break on the passing lane safety violation occurs. The most right rectangles represent the end of the mission. The AV speed and the acceleration verify two hard brakes in the mission while it was in the passing lane. The failure of the trajectory planning of the open-planner algorithm is apparent.

The failure to finish the overtaking mission is most prominent at the lowest speed, 1 m/s, this is due to the time the AV Shuttle is taking to perform the cut-in process and therefore cannot enact the overtaking manoeuvre within the simulation timeout which is 40 s. It was observed that for the proposed configuration, for the lower speed of the NPC, the open-planner trajectory evaluator is not reliable as it suggests waypoints that are not within safe navigation and this is due to the lack of firm decision-making of which roll-out to choose. Ultimately, this causes collision and DTC safety violations. Furthermore, the failure to finish the simulation

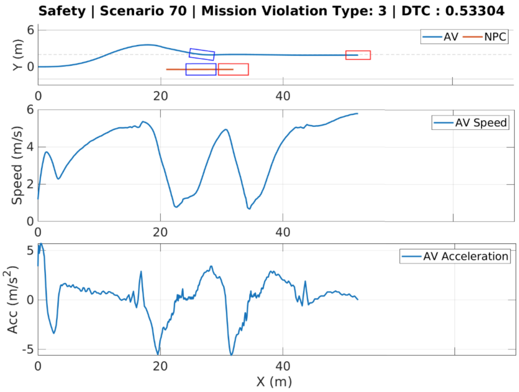

Figure 8: A Brake on Passing Lane safety violation

results, we see the low-speed delays in the overtaking manoeuvre decision making which results in the breach of the 40 s time-out.

The success rate of the safety test cases increases as the NPC drives from 1.4 to 2.1 m/s speed. This focal success point around scenario 10 with an NPC speed of 2.1 m/s can be a sign of matching the current configuration of perception and open-planner with the scenario situation.

The safety metrics results are shown in Figure 10 based on the initial relative distance from the AV to NPC. It shows that the rate of collision safety violations for longer initial distances from NPC slightly increased while the success rate decreased. This is the only trend that can be identified from results for initial relative distance, so it can be concluded that speed is a more determining parameter for the safety testing of our AV.

Overall, the results in Figure 7 indicate that speed is a critical parameter for our AV safety testing platform.

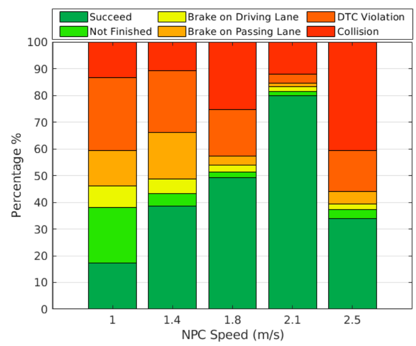

Figure 9: Test Results based on NPC Speed

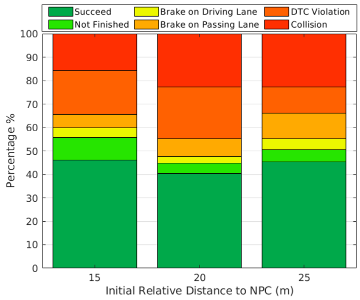

Figure 10: Results based on Initial Relative Distance to NPC

4.2 Cybersecurity Test Case

For the cybersecurity test cases we chose 2 of the 15 distinct scenarios (Figure 7). This was to allow a greater scale of testing to be conducted on a select number of relevant scenarios. Scenario 10 was chosen as it demonstrated the most reliable performance, in terms of the most successful overtaking manoeuvres. Scenario 2 was chosen as it demonstrated the least successful results for overtaking. These two scenarios were run 50 times each, as had been conducted with the safety scenario runs. Figure 11 shows the performance of cybersecurity testing, conducted on scenario 2 and scenario 10, in comparison to safety test cases.

Scenario 10 results reveal a discernible impact of the cyber attack. The LiDAR spoofing attack causes an increase in safety violations, prominently, in collisions and emergency breaking in the passing lane. This is also a concurrent result of the Scenario 2 test cases. Figure 3 shows the control level view, that incorporates sensor perception and mission and motion-planning. In the safety violation cases, we noticed that the euclidean clustering and kf_countour detect the spoofed LiDAR injection as an object and this false positive detection impacts the local-planning to force the AV to make the cut-in, in the overtaking manoeuvre process. Specifically, as the placement of the adversarial LiDAR device is on the left of the AV, the roll-outs of the left-side are blocked by the trajectory-evaluator. This forces the AV to veer right and attempt the cut-in process that causes predominantly collision, DTC safety violations.

Cao et al. [3] and Hallyburton et al. [8] identify density of the spoofed points to be one of the key variables affecting cyber attack success rate. Figure 12 and figure 13 present the sensitivity of each attack parameter according to the cyber attack test cases. From evaluating the raw data of the test sets, and the sensitivity analysis for the cyber attack test cases of scenario 10, we concur with these assessments. We find the rate of collisions is influenced by the density of the point cloud and the location of the attack. We can also see the influence the point of attack and duration have on causing a break on passing lane safety violation. As the duration of transmitting of the LiDAR point clouds increases and the location of the attack is further from the NPC, the likelihood of the AV initiating its breaks is higher.

In comparison, Scenario 2 cyber attack test case results show that safety violations are less sensitive to attack parameters. This can be due to the difficulty in interpreting the impact of cybersecurity on this scenario due to the already high rate of safety violations of the algorithms exhibited in the safety test case.

Table 5: Results of Cyber Attack applied to Scenario 10

Num.	V_{Col}	V_{DTC}	V_{BrP}	V_{BrD}	V_{NotF}	V_{Suce}
1	54%	20%	2%	0%	6%	18%
2	38%	38%	6%	2%	6%	10%
3	30%	28%	22%	2%	4%	14%
4	24%	28%	16%	6%	2%	24%
5	26%	16%	12%	6%	4%	36%
6	4%	4%	6%	4%	0%	82%
7	32%	14%	14%	6%	0%	34%
8	50%	24%	8%	2%	0%	16%
9	50%	30%	2%	2%	0%	16%
mean	34.2%	22.4%	9.8%	3.3%	2.4%	27.8%
std	15.9%	10.1%	6.7%	2.2%	2.6%	22.2%
min	4.0%	4.0%	2.0%	0.0%	0.0%	10.0%
max	54.0%	38.0%	22.0%	6.0%	6.0%	82.0%

Table 6: Results of Cyber Attack applied to Scenario 2

Num.	V_{Col}	V_{DTC}	V_{BrP}	V_{BrD}	V_{NotF}	V_{Suce}
1	16%	34%	28%	8%	14%	0%
2	26%	34%	20%	0%	8%	12%
3	20%	42%	20%	4%	6%	8%
4	26%	34%	16%	0%	14%	10%
5	22%	36%	16%	0%	20%	6%
6	22%	32%	20%	0%	18%	8%
7	0%	0%	0%	0%	0%	0%
8	0%	0%	0%	0%	0%	0%
9	0%	0%	0%	0%	0%	0%
mean	14.7%	23.6%	13.3%	1.3%	8.9%	4.9%
std	11.4%	17.9%	10.6%	2.8%	7.9%	4.9%
min	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
max	26.0%	42.0%	28.0%	8.0%	20.0%	12.0%

4.3 Real-World AV Testing

The real-world AV testing was conducted on a private road environment using our AV Shuttle, and an NPC vehicle (turquoise Mitsubishi iMIEV). The NPC vehicle is stationary during the tests as a safety assessment deemed it was too dangerous to conduct the experiment with a moving vehicle. This is due to the experiment being within a road environment where pedestrians and other vehicles are present. We conducted 3 test cases; a safety test case, cybersecurity test case and an optimised cybersecurity test case. The first test was an overtaking safety scenario. Two repetitions of the safety test case were conducted. The first test demonstrated a successful execution of the overtaking mission. The second test resulted in a DTC safety violation. The AV motioned to within 0.42 m of the NPC. The DTC violation is evident in Frame 3 of Figure 14, which details the second overtaking safety test case. Frame 4 demonstrates the eventual overtake after the DTC safety violation. Whilst the number of repetitions in the real-world pale in comparison to those conducted in the simulator, the real-world results

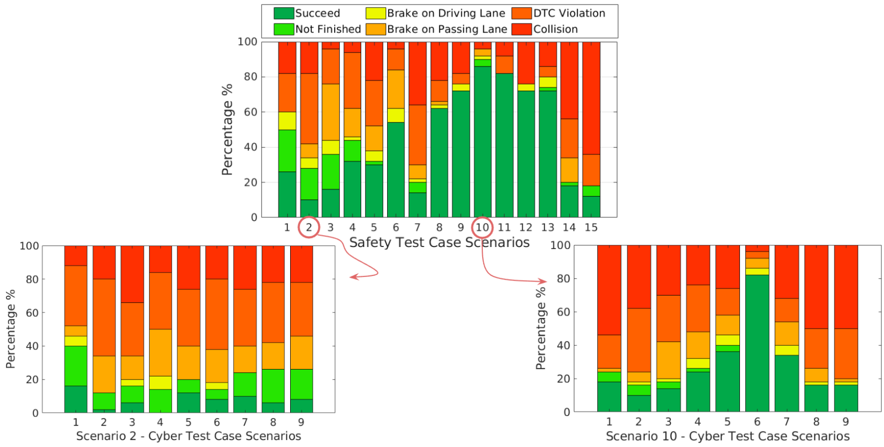

Figure 11: Performance Results Comparing Cyber Vs Safety Test Cases

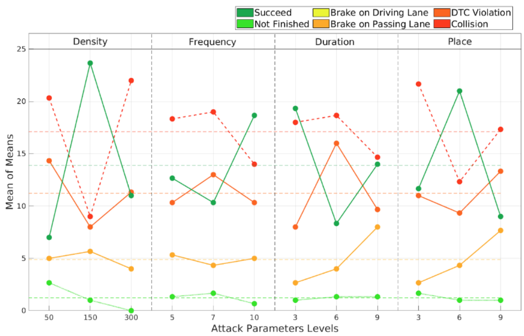

Figure 12: Scenario 10 - Cyber Attack Test Cases - Parameter Sensitivity

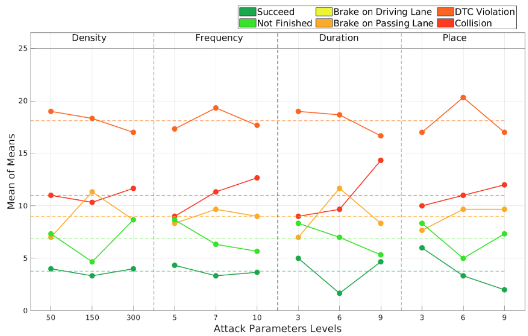

Figure 13: Scenario 2 - Cyber Attack Test Cases - Parameter Sensitivity

Table 7: Result of the 3 real-world test cases

Test Type	Num. of repeats	success	Safety Violations
Safety Tests	2	1	1 DTC=0.42m
Cyber Tests	2	1	1 DTC=0.38m
Optimised Cyber Tests	1	0	1 DTC=0.32m

concur with simulation results, that the AD algorithm does not have enough reliability for the deployment in real-world missions.

The cybersecurity test was conducted 3 times. Table 7 lists all the real-word experiments and their results. The first cybersecurity test demonstrated no impact from the spoofed LiDAR points and the overtaking manoeuvre was successful. The second cybersecurity test resulted in a DTC violation, the AV motioned to within 0.38 m of the NPC. After these two tests, we optimised the target angle of the spoofed points in relation to the attack scheme in Figure 6, to reduce the attack starting angle of θ_1. We did this because during the real-world test we observed that the reduced angle would provide assist the spoofed points to be closer to the AV trajectory and would cause the AV to detour from its intended route. It can be seen that this did work as the DTC decreased to 0.32 m. Figure 15 depicts the real-world cybersecurity test. Frame 2 represents the moment the attack was generated and perceived by the AD algorithm.

The videos and images related to the real-world tests are found on GitHub site.

5 DISCUSSION

From the analysis of the results we interpreted that different safety violations are connected to different modules of the AD algorithm.

Perception Module) We interpreted the cause of safety violations of the emergency break in the passing lane and emergency break in the driving lane to be related to the quality of the ground filtration. As we observed, ground filtering outcome changes during the AV maneuvers (turns) because the shuttle body is tilted because of suspension and this results in the lidar reference frame orientation

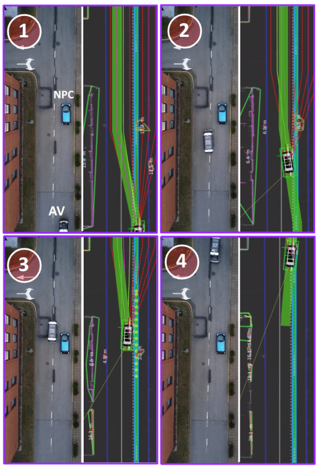

Figure 14: Real-World AV Test - Safety Test Case

Figure 15: Real-World AV Test - Cyber Attack Test Case

changing. Then some part of the ground point cloud as an unfiltered perception can be seen in the detection algorithms as an obstacle. This fake sudden obstacle might stop the AV during the motion. The spoofed LiDAR point cloud threat model is likely to make this condition worse. Optimisations for this: New body designs to rectify or limit the issues of LiDAR with the physics of the AV Shuttle are being developed. To focus specifically on these corner and edge cases and look at optimisation of the filtering of the perception algorithm. The latter recommendation is complicated by the fact it may include trade-offs; if the LiDAR perception algorithm is specifically tuned for this corner/edge case it could lead to over-filtration in normal driving scenarios, therefore this is one of the optimisation options to resolve the perception for the algorithm.

Open-Planner Module) We interpret the cause of safety violations for DTC and collision as due to an issue of the open-planner in predicting the trajectory of the NPC during the process of performing a cut-in, in front of the NPC. The optimisation would involve incorporation of features that would enable the prediction of the trajectory of the NPC and for perception improve the perception of the side-lidar to accurately perceive the NPC. We found that optimising all the perception and open-planner parameters for our shuttle model would significantly improve the reliability of the AD algorithm.

5.1 Open-Planner Developer Feedback

We sent a presentation of our results to the developers of the open-planner AD algorithm. In response, they acknowledged that it is a developing algorithm and we are engaged in more detailed discussions with them on how to optimise the algorithm. They also announced they are transitioning from Autoware.ai to Autoware.universe which is a more developed and advanced platform. Amongst their responses, they also pointed to the novelty of receiving feedback on the reliability of cybersecurity test cases in addition to safety test cases.

6 RELATED WORK

The closest contributions to our work are Yang et al. [20], Hallyburton et al. [8], Cao et al. [3] and Zhu et al. [21]. Each of these papers utilises a LiDAR spoofing threat model that varies based on the method for delivering the attack, adversarial generation and the type AD algorithm. Hallyburton et al. [8] target camera and LiDAR sensor fusion. They identify a blind spot between the camera and LiDAR sensor at the rear of the target AV. They use a malicious, 3D LiDAR point cloud array to inject malicious spoof points into the rear angle of the target AV. The attack was tested in a high-fidelity simulation and real-world against multiple perception algorithms. The results revealed a high rate of success utilising this attack. Cao et al [3], Yang et al [20], and Zhu et al [21] developed LiDAR spoofing attacks based on a threat model of a malicious LiDAR 3D point cloud injection in the road environment and by the roadside. Each of these contributions demonstrated that cyber attack results from AV simulation testing can be used to identify key parameters such as point cloud density, attack location and duration and that these parameters can be optimised to test the robustness of perception algorithms. We chose to extend from the related literature, in our work, in three areas; simulation testing configuration, safety criteria evaluation and target AD algorithm is in the developmental phase and is used within a real-world AV program. A feature of the selected work is that simulation testing often selected only one frame or a limited amount of frames and therefore the full driving mission was not observed. Whilst this is useful for reducing testing resource usage, running massive scale of tests and applicable to the scope of their work, as our study evaluates the end-to-end AD algorithm and combines safety, our study focused on conducting simulation testing for the entire driving mission. Secondly, the evaluation of cyber attacks focused on attack success rate and attack parameters whilst the safety impact on the AV as a result of cyber attacks was not as clearly elaborated. In our study, we evaluate the cyber attack test cases with the same criteria as the safety case to derive the category of safety violation. Lastly, most of the simulations use default AV configurations and evaluate well-established algorithms. Our study uses a simulator configured for a real-world AV and evaluates an AD algorithm in the developmental stage where reliability and optimisation are required to be assessed under safety, non-cyber test cases before the impact of cyber attacks can be understood.

7 CONCLUSION

We developed a combined methodology for safety and cybersecurity utilising a digital twin, high-fidelity simulation environment and a real-world AV shuttle for public transportation. We evaluated our approach on a developing AD algorithm consisting of open-planner, as the mission and motion-planning module. We evaluated the reliability of the AD algorithm on an overtaking scenario using test cases for safety and cybersecurity based on a LiDAR spoofing attack. The combined safety and cybersecurity testing enabled us to assess the outcome of the cyber attack in comparison to the ground truth of the reliability of the AD algorithm established in the safety testing. This clearly demonstrated the effect of cyber-attacks regardless of the reliability of the algorithm. We were also able to assess, from the performance of the AD algorithm, that the algorithm is not optimised for the overtaking manoeuvre. In our research, we discovered several sensitive parameters that play a significant role in the safety outcome of the AV and the success rate of the cyber attack. Furthermore, we provided the results of our testing platform to the designer of the open-planner algorithm. Based on their feedback a process has been initiated to optimise the AD algorithm. All test scripts and software resources including our AV simulation configurations and research data used in the combined safety and security testing will be available for the research community on GitHub.

7.1 Future Work

Future work consists of diversifying the safety scenarios to include a more complex and broader range of scenarios. Cybersecurity testing will be evolved to develop black-box testing models. Furthermore, we will continue to develop methods for optimising testing to factor in real-world limitations such as resource usage and time.

ACKNOWLEDGMENTS

This research has received funding from the following grants: the European Union's Horizon 2020 Research and Innovation Programme, under grant agreements No. 856602 and No 883321 (CityScape), and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No 2014-2020.4.01.20-0289.

REFERENCES

[1] Euro NCAP Working Group on Automated Driving. 2019. *Euro NCAP's First step to assess automated driving systems.* Technical Report. European New Car Assessment Programme.
[2] Adith Boloor, Karthik Garimella, Xin He, Christopher Gill, Yevgeniy Vorobeychik, and Xuan Zhang. 2020. Attacking vision-based perception in end-to-end autonomous driving models. *Journal of Systems Architecture* 110 (2020). https://doi.org/10.1016/j.sysarc.2020.101766 arXiv:1910.01907
[3] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. 2019. Adversarial Sensor Attack on LiDAR-Based Perception in Autonomous Driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 2267–2281. https://doi.org/10.1145/3319535.3339815
[4] CARLA Simulation Project. 2022. *Combined Safety and Cybersecurity Testing Methodology for Autonomous Driving Algorithms.* Technical Report. CARLA.
[5] Hatem Darweesh, Eijiro Takeuchi, and Kazuya Takeda. 2021. OpenPlanner 2.0: The Portable Open Source Planner for Autonomous Driving Applications. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. 313–318. https://doi.org/10.1109/IVWorkshops54471.2021.9669253
[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*. 1–16.
[7] Junyao Guo, Unmesh Kurup, and Mohak Shah. 2020. Is it Safe to Drive? An Overview of Factors, Metrics, and Datasets for Driveability Assessment in Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems* 21, 8 (2020), 3135–3151. https://doi.org/10.1109/TITS.2019.2926042
[8] R. Spencer Hallyburton, Yupei Liu, Yulong Cao, Z. Morley Mao, and Miroslav Pajic. 2022. Security Analysis of Camera-LiDAR Fusion Against Black-Box Attacks on Autonomous Vehicles. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 1903–1920. https://www.usenix.org/conference/usenixsecurity22/presentation/hallyburton
[9] Zhongyuan Hau, Kenneth T Co, Soteris Demetriou, and Emil C Lupu. 2021. Object Removal Attacks on LiDAR-based 3D Object Detectors. In *Third International Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*.
[10] ISO/TC 22/SC 32 Electrical and electronic components and general system aspects. 2018. *ISO 26262-1:2018 Road vehicles — Functional safety.* Technical Report. International Standards Organization.
[11] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. 2018. Autoware on board: Enabling autonomous vehicles with embedded systems. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 287–296.
[12] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and X. Zhang. 2018. Trojaning Attack on Neural Networks. In *NDSS*.
[13] Ehsan Malayjerdi, Raivo Sell, Mohsen Malayjerdi, Andres Udal, and Mauro Bellone. 2022. Practical path planning techniques in overtaking for autonomous shuttles. *Journal of Field Robotics* 39, 4 (2022), 410–425.
[14] Mohsen Malayjerdi, Vladimir Kuts, Raivo Sell, Tauno Otto, and Barış Cem Baykara. 2020. Virtual Simulations Environment Development for Autonomous Vehicles Interaction. In *ASME International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers.
[15] Mohsen Malayjerdi, Andrew Roberts, Olaf Maennel, and Ehsan Malayjerdi. 2022. *Combined Safety and Cybersecurity Testing Methodology for Autonomous Driving Algorithms.* https://github.com/momala/Safety_Cyber_Testing.git
[16] Raivo Sell, Ehsan Malayjerdi, Mohsen Malayjerdi, and Baris Cem Baykara. 2022. Safety Toolkit for Automated Vehicle Shuttle -Practical Implementation of Digital Twin. In *2022 International Conference on Connected Vehicle and Expo (ICCVE)*. 1–6. https://doi.org/10.1109/ICCVE52871.2022.9742881
[17] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. 2020. Towards robust LiDAR-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. *Proceedings of the 29th USENIX Security Symposium* (2020), 877–894. arXiv:2006.16974
[18] KWOK-LEUNG TSUI. 1992. AN OVERVIEW OF TAGUCHI METHOD AND NEWLY DEVELOPED STATISTICAL METHODS FOR ROBUST DESIGN. *IIE Transactions* 24, 5 (1992), 44–57. https://doi.org/10.1080/07408179208964244 arXiv:https://doi.org/10.1080/07408179208964244
[19] James Tu, Huichen Li, Xinchen Yan, Mengye Ren, Yun Chen, Ming Liang, Eilyan Bitar, Ersin Yumer, and Raquel Urtasun. 2021. Exploring Adversarial Robustness of Multi-sensor Perception Systems in Self Driving. In *5th Annual Conference on Robot Learning*. https://openreview.net/forum?id=m5k1XdK5nI2
[20] Kaichen Yang, Tzungyu Tsai, Honggang Yu, Max Panoff, Tsung-Yi Ho, and Yier Jin. 2021. Robust Roadside Physical Adversarial Attack Against Deep Learning in Lidar Perception Modules. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (Virtual Event, Hong Kong) *(ASIA CCS '21)*. Association for Computing Machinery, New York, NY, USA, 349–362. https://doi.org/10.1145/3433210.3453106
[21] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. 2021. Can We Use Arbitrary Objects to Attack LiDAR Perception in Autonomous Driving? *Proceedings of the ACM Conference on Computer and Communications Security* (2021), 1945–1960. https://doi.org/10.1145/3460120.3485377

Paper V

Mohsen Malayjerdi, Quentin A. Goss, M. Ilhan Akbas, Raivo Sell, and Mauro Bellone. “A Two-Layered Approach for the Validation of an Operational Autonomous Shuttle”, in IEEE Access, vol. 11, pp. 89124-89137, 2023.

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

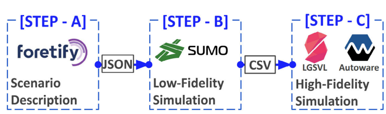

FIGURE 2. Three main steps of the proposed validation method. The format of each signal passing among these steps is annotated.

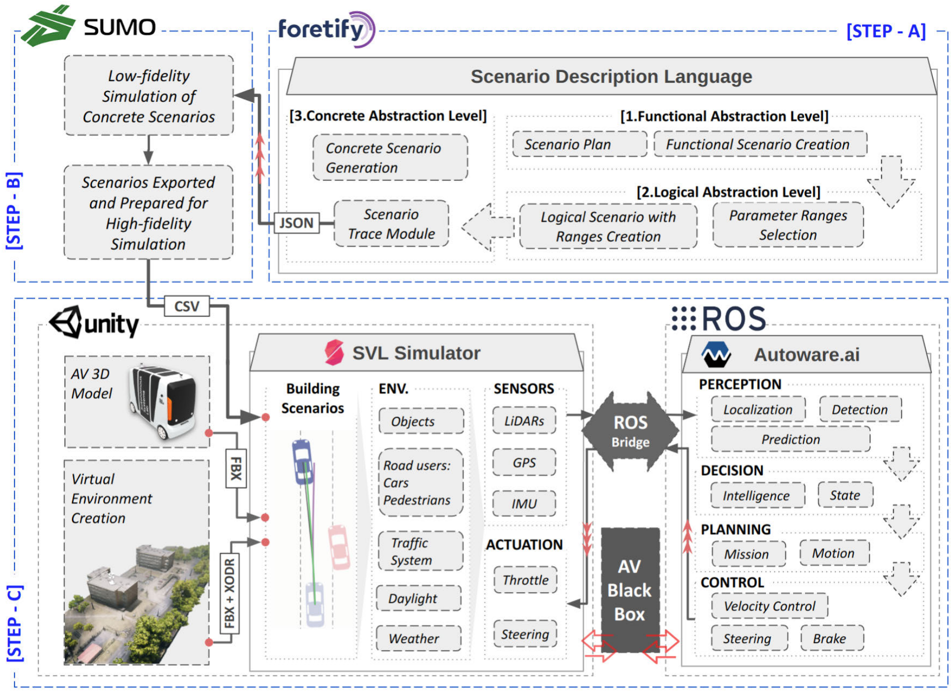

FIGURE 3. High-level architecture of the scenario generation, simulator, and AV control system. The low-fidelity simulation is represented by the SUMO block, while the SVL simulator and the Autoware control system can be considered as one high-fidelity simulation block. Please refer to Section III for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

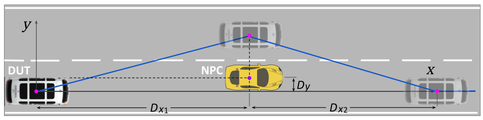

FIGURE 4. Two relative positions are given, Dx_1 and Dx_2, define the NPC position in each scenario. Dy is determined by Foretify.

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 $Y = [\text{Pose}(x, y), \text{Collision}]$, 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

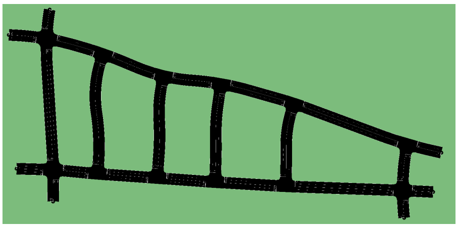

FIGURE 5. The street network.

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.

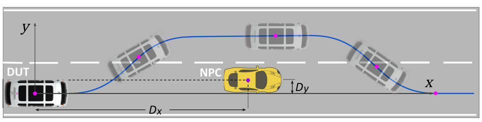

FIGURE 6. Two relative coordinates, ***Dx*** and ***Dy***, define the NPC position in each scenario.

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 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.

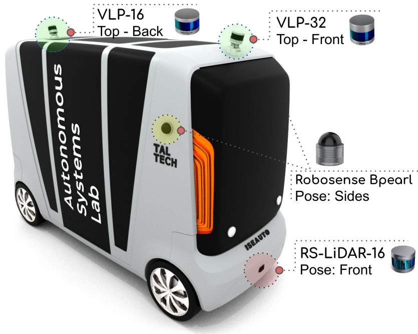

FIGURE 7. iseAuto simulated model with different LiDARs installed.

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 = $ [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

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

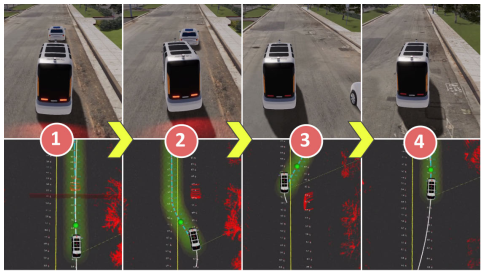

FIGURE 8. One simulated scenario shown in different frames. The top frames represent the SVL simulator and the below one displays the RViz visualization software receiving simulated data.

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.

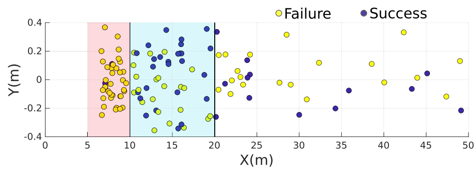

FIGURE 9. Points representing all initial relative NPC locations in 120 scenarios that are marked based on their simulation result.

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.

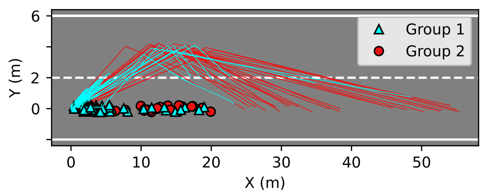

FIGURE 10. **All routes traveled in the Foretify simulation.**

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.

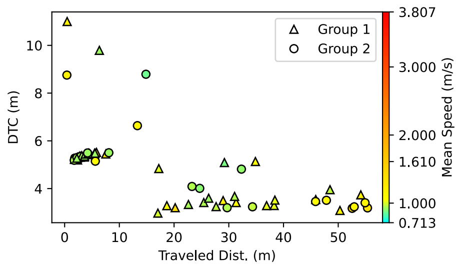

FIGURE 11. **Results of the low-fidelity simulations.**

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.

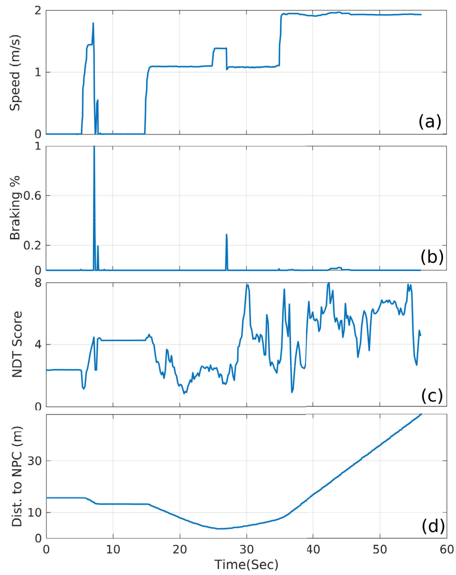

FIGURE 12. **The result of a filtered scenario simulation; (a) Speed of the DUT, (b) normalized braking pressure, (c) NDT score for localization performance, and (d) distance measured from Ego to NPC.**

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.

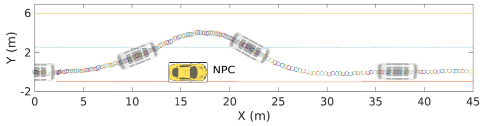

FIGURE 13. Simple representation of the Ego traveling route and the NPC position during the scenario.

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.

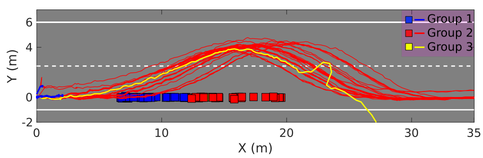

FIGURE 14. All traveled routes in the 87 selected scenarios are shown and divided into three groups as described.

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).

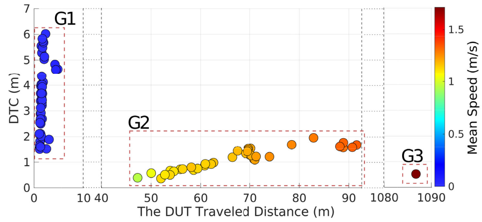

FIGURE 15. Result recorded from the high-fidelity simulation of selected scenarios. Each point on the chart represents the traveled distance and the minimum distance to collision (DTC). The color bar also shows the mean speed of the DUT in the mission. G1, G2, and G3 represent each group's scenarios.

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.

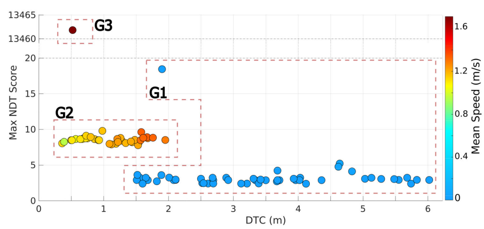

FIGURE 16. Scenarios represented by DTC and NDT score. The color bar also shows the mean speed of the DUT in the mission. G1, G2, and G3 represent each group's scenarios.

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.

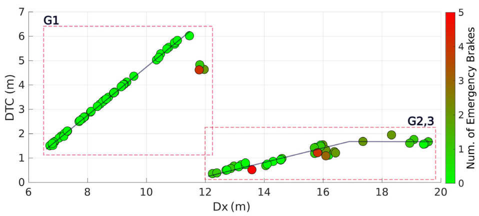

FIGURE 17. Scenarios represented by the initial longitudinal distance to the NPC and the minimum DTC reached during the simulation. Also, the color bar demonstrates the normalized brakes magnitude through each scenario.

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.

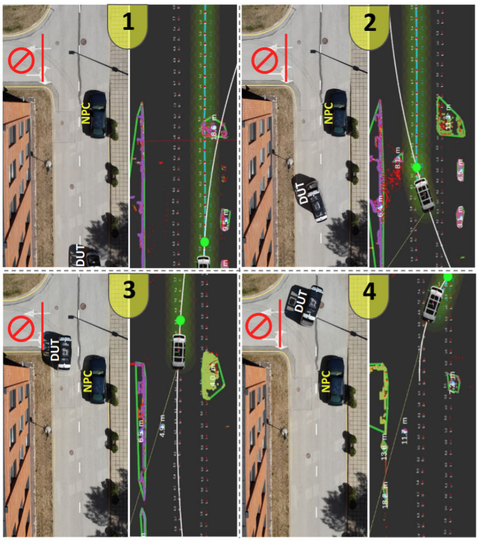

FIGURE 19. Four different time frames of the passing experiment are shown in the real test environment beside the RViz visualizer. The T intersection entrance was closed during the test.

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

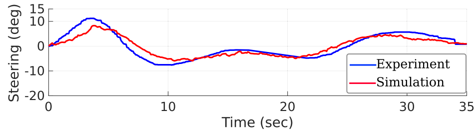

FIGURE 20. Values of Steering angle on the wheels recorded during the same scenario in the high-fidelity simulation and real experiment.

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] A. Roberts, M. Malayjerdi, E. Malayjerdi, and O. M. Maennel, "Combined safety and cybersecurity testing methodology for autonomous driving algorithms," in *Proc. Comput. Sci. Cars Symp. (CSCS)*. New York, NY, USA: Association for Computing Machinery, 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.

• • •

List of Publications Not Included in This Thesis

VI **Mohsen Malayjerdi**, Bariş Cem Baykara, Raivo Sell, and Ehsan Malayjerdi. "Safety Assessment and Simulation of Autonomous Vehicle in Urban Environments." In IOP Conference Series: Materials Science and Engineering, vol. 1140, no. 1, p. 012032. IOP Publishing, 2021.

VII Raivo Sell, Ehsan Malayjerdi, **Mohsen Malayjerdi**, and Baris Cem Baykara. "Safety toolkit for automated vehicle shuttle-Practical implementation of digital twin." In 2022 International Conference on Connected Vehicle and Expo (ICCVE), pp. 1-6. IEEE, 2022.

VIII Ehsan Malayjerdi, Raivo Sell, **Mohsen Malayjerdi**, Andres Udal, and Mauro Bellone. "Practical path planning techniques in overtaking for autonomous shuttles." Journal of Field Robotics 39, no. 4 (2022): 410-425.

IX Andrew Roberts, **Mohsen Malayjerdi**, Mauro Bellone, Olaf Maennel, and Ehsan Malayjerdi. "Analysing Adversarial Threats to Rule-Based Local-Planning Algorithms for Autonomous Driving." NDSS Symposium on Vehicle Security and Privacy (VehicleSec), 2023.

X Rahul Razdan, Mustafa İlhan Akbaş, Raivo Sell, Mauro Bellone, Mahesh Menase, and **Mohsen Malayjerdi**. "PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration." IEEE Access, vol. 11, pp. 28343-28354, 2023.

XI **Mohsen Malayjerdi**, Gemb Kaljavesi, Frank Diermeyer, and Raivo Sell. "Scenario-based validation for autonomous vehicles with different fidelity levels." In 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6, 2023.

Curriculum Vitae

1. Personal data

Name	Mohsen Malayjerdi
Date of birth	22 September 1988
Nationality	Iran

2. Contact information

Email	mohsen.malayjerdi@taltech.ee
LinkedIn	mohsen-malayjerdi

3. Education

2018–2023	Tallinn University of Technology, Estonia Mechanical Engineering, Ph.D. student
2011–2014	Ferdowsi University of Mashhad, Iran Mechanical Engineering, M.Sc.
2006–2010	Azad University of Mashhad, Iran Mechanical Engineering, B.Sc.

4. Language competence

Persian	native
English	fluent
Estonian	basic

5. Professional employment

2019 - Present	Early Stage Researcher, Tallinn University of Technology
2011 - 2019	CEO and R&D Engineer, Robot Makers (H2M) Co.

6. Field of research

- Autonomous Vehicle system V&V
- Simulation

Elulookirjeldus

1. Isikuandmed

Nimi	Mohsen Malayjerdi
Sünniaeg	22. september 1988
Kodakondsus	Iraani

2. Kontaktandmed

E-post	mohsen.malayjerdi@taltech.ee
LinkedIn	mohsen-malayjerdi

3. Haridus

2018–2023	Tallinna Tehnikaülikool, Eesti Mehhaanotehnika, Ph.D. tudeng
2011–2014	Ferdowsi Mashhadi ülikool, Iraan Mehaanika, M.Sc.
2006–2010	Azad Mashhadi ülikool, Iraan Mehaanika, B.Sc.

4. Keelteoskus

Pärsia	emakeel
Inglise	kõrgtase
Eesti	algtase

5. Teenistuskäik

2019- . . .	Tallinna Tehnikaülikool, Doktorant-nooremteadur
2011-2019	Tegevjuht ning teadus- ja arendusinsener, Robot Makers (H2M) Co.

6. Teadustöö põhisuunad

- Autonoomsete sõidukite valideerimine ja verifitseerimine
- Simulatsioonid

ISSN 2585-6901 (PDF)
ISBN 978-9916-80-040-9 (PDF)