Challenges in Virtual Testing of Autonomous Vehicles

Piazzoni, Andrea ; Vijay, Roshan ; Cherian, Jim ; Chen, Lyu ; Dauwels, Justin

DOI
10.1109/ICARCV57592.2022.10004249

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)

2022 17th International Conference on
Control, Automation, Robotics and Vision (ICARCV)
December 11-13, 2022. Singapore

Challenges in Virtual Testing of Autonomous Vehicles

Andrea Piazzoni[1,2] Roshan Vijay[2], Jim Cherian[2],
Lyu Chen[3], *Senior Member, IEEE*, Justin Dauwels[4], *Senior Member, IEEE*.

2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV) | 978-1-6654-7687-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICARCV57592.2022.10004249

***Abstract*— The worldwide development of Autonomous Vehicles (AVs) has also encouraged the use of software simulators for virtual testing of AVs. However, the effectiveness of the AV simulators is constrained by numerous challenges, such as their computational cost and lack of fidelity in specific areas. In this paper, we describe the modality of virtual testing and its benefits for AV development and validation. Moreover, we summarize and provide an overview of the state-of-the-art AV simulators, their limitations, and the current directions toward improvement.**

I. Introduction

As Autonomous Vehicles (AVs) are becoming more prevalent around the world, many startups and experienced vehicle manufacturers alike are deeply involved and invested in developing them. Ensuring the safety of AVs operating and testing in the real world is challenging, and requires a rigorous and joint testing, verification, and validation efforts between the AV developers, government regulators and third-party regulatory agencies.

Some estimations [1] have concluded that to demonstrate with 95% confidence and 80% power that an AV's failure rate is 20% better than an optimistic human driver failure rate of 1.09 fatalities per 100 million miles, it may need to be tested over a distance of at least 11 billion miles. Surely, this is impossible to achieve even for the most well funded and staffed AV developers. Simulation based virtual testing is a viable and pragmatic alternative to make AVs safer simply by allowing them to be driven across longer distances and test against more conditions.

Vehicles with varying degrees of automation have been prevalent in the market since the end of the 20th century. Automated features, e.g., cruise control, were first introduced to reduce driving stress and improve driver comfort. Today, they have become part of a car's active safety systems. More recently, automotive OEMs have started to introduce conditional SAE L3 [2] semi-automated driving under low speed stop-and-go traffic conditions on expressways [3].

OEMs have been using simulation testing extensively in the pursuit of software quality and to ensure safety of ADAS systems and it is an important step in the product development lifecycle as it provides a safe and relatively inexpensive avenue to test safety critical vehicle systems before they are deployed in the real world. There are a variety of general purpose and task-specific tools, test methodologies and modalities to enable in-depth virtual testing of all vehicular electronic, electrical, electromechanical and mechanical systems.

Even though the vast majority of these tools were conceived with ADAS testing in mind, many of them have evolved to tackle the challenges specific to simulation-based virtual testing of AVs. Most of these tools are highly modular and allow for extensive modification by the end user to enable communication with automated driving system (ADS) software or other task-specific simulation tools.

Considering the many advantages of virtual testing as well as the numerous free, open source and commercial AV and ADAS testing tools available in the market today, virtual testing has proven to be a valuable asset, making it possible to test AVs in a safe, convenient and cost-effective manner.

However, significant technological and scientific challenges hinder AV simulators' effectiveness and widespread employment. In this paper, we aim to provide an overview of AV simulation tools, their applications, and their limitations and we make the following contributions:

- In section II, we illustrate standard methodologies and the benefits of employing AV virtual testing. This section explains why the effort to improve AV simulators is valuable and needed.
- In section III, we present notable AV simulators, both commercial and open source. Moreover, we identify their strengths and weaknesses.
- In section IV, we focus on the major scientific limitation, i.e., the fidelity and validation of the simulator components.
- In section V, we introduce upcoming solutions aimed at improving various aspects of AV simulators.

II. Virtual Testing

Virtual testing is a crucial step in assessing the performance and safety of Autonomous Vehicles (AV). Virtual testing of AVs can be achieved using a variety of different modalities. These include, commonly, Software-in-the-Loop (SiL), Hardware-in-the-Loop (HiL), and Vehicle-in-the-Loop (ViL). Other testing modalities may also be considered, such as Driver-in-the-Loop (DiL). Still, these may serve different testing purposes, such as studying the response and reactions

This work was supported in part by the Centre of Excellence for Testing & Research of AVs - NTU (CETRAN).

[1]ERI@N, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore andrea006@ntu.edu.sg

[2]Centre of Excellence for Testing & Research of AVs, Nanyang Technological University, Singapore rvijay@ntu.edu.sg, jcherian@ntu.edu.sg

[3]School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore lyuchen@ntu.edu.sg

[4]TU Delft, Dept. of Microelectronics, Fac. EEMCS, Mekelweg 4 2628 CD, Delft. j.h.g.dauwels@tudelft.nl

of a human driver rather than validating the functional safety of the Automated Driving System (ADS) software.

A. Virtual testing modalities

While different testing modalities have a different scope and offer different advantages, they usually share common disadvantages and limitations:

- Environment modeling is a time-consuming process;
- Complex simulations of physically accurate sensors are computationally expensive;
- In some situations, real-time performance may not be guaranteed, and this may affect the dynamic response of the ADS under test;
- Perfect fidelity is almost impossible to achieve, and there will always be a mismatch between the simulated world and its physical. counterpart.

X-in-the-loop (XIL) refers to the virtual simulation-based testing setup where the reaction (response) of X being tested (in the virtual environment) will influence the virtual environment itself. X refers to the exact entity (e.g., the Models, Software, Hardware, Vehicle, Human driver etc.) which is in a closed loop with the environment around, and the environment responds to with appropriate actions, leading to a *closed loop test*. Few of the most common XiL paradigms are listed below.

1) Software-in-the-Loop (SiL): Software-in-the-Loop is a method of testing and validating an ADS's various subsystems, modules, and software packages in a simulated environment. The ADS, its modules and/or subsystems, and the simulated environment run as separate processes on the same computing hardware or distributed across various systems connected over a common network protocol.

A typical SiL simulation environment will contain the following modules as seen in Fig. 1:

1) Environment model – A 3D of the test environment and road networks with high fidelity [4].
2) Environmental effects – The ability to generate various environmental conditions which may affect the functioning of the sensors of the AV, such as rain, fog, haze, and so on.
3) Sensor models – A physically accurate representation of the sensors installed on the VUT
4) Traffic model – An accurate and configurable simulation of other road users.
5) Vehicle model – A 3D recreation of the Vehicle-Under-Test (VUT) with multiple degrees of freedom.

These modules directly communicate with the various modules of the ADS under test such as the sensing & perception and control.

SiL simulation is an important aspect of virtual AV testing due to its many advantages, such as:

- SiL simulation can be used to test the ADS with automated DevOps pipelines on standard computing hardware.
- Different types of system-level tests can be performed, including but not limited to scenario-based testing, functional safety tests, tests of specific subsystems such as sensing and perception, or the decision-making module of the ADS.
- Multiple instances of simulations can be tested together, even faster than real-time provided the hardware is powerful enough.

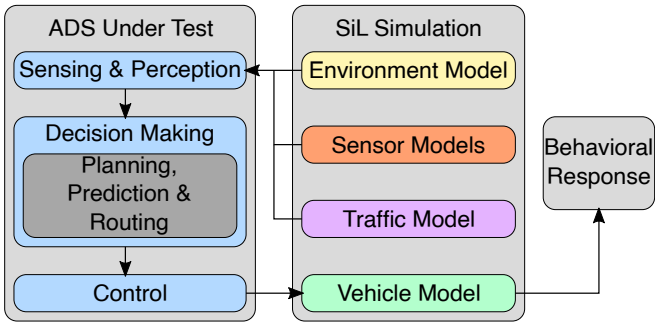

Fig. 1: SiL simulation setup for testing an ADS

2) Hardware-in-the-Loop (HiL): HiL is similar to SiL but with the added complexity of the process-actuator system added into the loop. This may include various hardware submodules handling ADS components such as sensing and perception. This may even include the actual sensors being fed synthetic data such as the camera system being shown a simulated world on a large display. HiL can be considered the next step of the ADS development process after SiL testing. However, even HiL is not a perfect testing solution and may exhibit issues such as those shown by SiL testing, including problems with real-time performance, environment modeling, and fidelity. HiL does not include all the process-actuator systems in the loop and as such may be limited depending on which hardware components are included in the simulation chain.

3) Vehicle-in-the-Loop (ViL): ViL solves the inherent problems faced by HiL by integrating the entire vehicle and all its subsystems and modules in the simulation loop. A physical vehicle with its many different ECUs, CAN interfaces, sensors and computing hardware is integrated directly with the simulation toolchain which generates synthetic information required by the ADS allowing it to operate as if installed in the actual vehicle. ViL can be considered the final step in the virtual testing and validation process before the AV is tested physically, either through controlled or uncontrolled testing on real test circuits or private/public roads. Using this testing modality, we can implement a hard real-time solution which tests the temporal response of the ADS as well.

There are other test modalities such as Driver-in-the-Loop (DiL) but these may not be directly relevant for AV testing. DiL refers to when a human driver is in control of the vehicle under test and the world around them is synthesised using simulation tools. For the purposes of this review, we will be focusing solely on Software-in-the-Loop (SiL) testing.

B. Test types

Tests for virtual safety validation of an ADS can be classified based on if they are controlled or uncontrolled.

- Controlled testing – Typically conducted using predefined scenarios in a controlled space such as a virtual test track with closed boundaries and where sensory conditions are controlled.
- Uncontrolled testing – Typically conducted on virtual roads modelled after real-world locales with randomized but realistic traffic models and sensory conditions.

For both types of testing, specific test scenarios can be defined and implemented in the Virtual Testing Toolchain (VTT). Scenario based testing is an important method within the context of AV virtual testing and validation and helps the tester to study the behavioural response, safety and determinism of the ADS under test. Scenarios can be defined based on the work done in [5] as:

1) Functional scenarios / Scenario categories: Functional scenarios, otherwise referred to as scenario categories, give a high level description of the scenario to be tested. For example, these may give an abstract and verbal description of the road network, its structure, the different stationary and movable objects in a scene, overall weather conditions of the scene and so on.

2) Logical scenarios / Scenario variants: Logical scenarios attempt to parameterize functional scenarios by assigning a range of parameter values to quantifiable aspects of the scenario. For example, a logical scenario may describe a range of lane widths, curve radii and positions of traffic signs for the road network. It may also specify a range of values for the travelling velocities of the various dynamic objects in the scene. However, a logical scenario is still not enough to describe a specific test case which can be used to test an AV's behavioural response.

3) Concrete scenarios / Test cases: A test case or concrete scenario is when specific parameter values are selected from the range of parameters values described in a logical scenario/scenario variant. Each test case may be different based on if any of the individual parameters differ. Test cases are the final step in defining a scenario to be tested in a virtual testing toolchain where it can be implemented based on the decided parameters. For example, the test case would have well defined parameters such as the location and/or dimensions of the road network, position of the various static and dynamic objects, their velocities among other values.

III. Simulators

In this section, we report the most common AV simulators employed in the AV community. Each simulator has pros and cons from economic, ergonomic, and technological points of view. In this paper, we focus on technological features and omit the rest.

A. Open Source

There are two notable Open Source AV Simulators available: CARLA and SVL. They are respectively developed in the Unreal Engine [6], and in the Unity Engine [7]. This approach has a high potential for custom extensions and innovative solutions that expand their functionalities.

- **CARLA** [8] – CARLA is the primary choice for academic research. It offers a powerful API to control all aspects of a simulation and free assets (maps and vehicles). Moreover, it has ROS integration and supports co-simulation with many external modules.
- **SVL** [9] – SVL has been discontinued by the original developer (June 31st, 2022), but it remains Open Source and is available to the community. The main strength is the integrated bridge towards Baidu Apollo [10] or Autoware [11], as well as providing a framework for sharing assets such as maps, vehicles, and sensors.

B. Commercial

Many commercial vehicle simulators are available that may be used to facilitate the testing of ADAS or AV modules. Most if not all of these tools were initially conceived specifically for automotive OEMs to develop and refine the driving assistance systems of their vehicles. These tools are meant to be end-to-end solutions incorporating many of the modules integral to SiL testing as mentioned in Section II-A.1, while also being able to support other testing modalities.

- **MSC VIRES Virtual Test Drive (VTD)** [12] – Originally developed by VIRES Simulationstechnologie GmbH, VTD is a powerful end-to-end vehicle simulation tool that integrates environment design, vehicle dynamics, high-quality weather and environmental effects along with sensor simulation. Even though it is a standalone software, it is quite modular. It can integrate well with other simulation plugins, e.g., those dedicated to vehicle dynamics, such as MSC's own Adams. VTD supports multiple testing modalities such as SiL, HiL, ViL and DiL (Driver-in-the-loop).
- **IPG CarMaker** [13] – CarMaker is another general-purpose end-to-end vehicle simulation tool focusing slightly more on accurate vehicle dynamics. CarMaker promises real-time capable, realistic vehicle dynamics models. As a result, it is frequently used as a dedicated vehicle dynamics simulator in addition to being used as a general-purpose vehicle simulator.
- **Siemens Simcenter Prescan** [14] – Prescan is a general purpose vehicle simulator with specific emphasis on delivering accurate sensor simulation. It is able to accurately model and simulate sensors such as LiDAR, radar and cameras. It has deep integration with MathWorks MATLAB and Simulink. MATLAB/Simulink are well defined and highly documented software with many ways to read and write data from a simulation model, thereby making Prescan quite modular and open to integrate with other simulation tools.
- **AB Dynamics rFPro** [15] – rFPro was originally known as rFactor Pro and is widely used by many of the top racing constructors around the world for driver improvement and testing. It is also used extensively by automotive OEMs for the development of ADAS assistance features with a specific focus on simulating accurate vehicle dynamics with physically accurate road models. rFPro's in-built rigid body vehicle dynamics

Simulator	Accessibility	Sensor simulation	Vehicle dynamics	Traffic simulation	Scripting / modularity	OpenDRIVE support
CARLA	Open source	✓	✓	Limited	Python	✓
SVL	Open source	✓	✓	Limited	Python	✓
VTD	Commercial	✓	✓	Limited	C++, SCP commands	✓
CarMaker	Commercial	✓	✓	✗	C	✓
Prescan	Commercial	✓	✓	Limited	Simulink and C++	✓
RFPro	Commercial	✓	✓	✓	MATLAB/Simulink	✓
Nvidia Drive Sim	Commercial	✓	✓	✓	Omniverse kit	✓
SUMO	Open source	N/A	N/A	✓	Python	✓
VISSIM	Commercial	N/A	N/A	✓	Python	✓
Ansys AVxcelerate	Commercial	✓	N/A	N/A	Python	N/A
Adams	Commercial	N/A	✓	N/A	C++	N/A
CarSim	Commercial	N/A	✓	N/A	MATLAB, VB, C/C++	✓

TABLE I: A comparison of different simulation tools

samples various vehicle systems such as the suspension and the drivetrain at very high refresh rates. It also includes a C++ API as well as integration with Simulink.

- **NVIDIA DRIVE Sim** [16] – NVIDIA, being at the forefront of computer graphics, has led the massive adoption of Graphics Processing Unit (GPU) and CUDA, which has enabled large-scale machine learning. Automated driving is one of its important application focus areas, for which NVIDIA has made a general-purpose ADS solution called DRIVE and a general-purpose AV simulator called DRIVE Sim. This simulator uses NVIDIA's own Omniverse graphical rendering and simulation backend. It supports distributed computing across many nodes and GPUs for multi-sensor simulations. DRIVE Sim also leverages NVIDIA's hardware expertise by utilizing their RTX real-time raytracing solution in order to render physically accurate sensor models. The Omniverse kit allows for DRIVE Sim to be modular and extensible to integrate with many other tools.

C. External Tools

Beside the general purpose simulators previously mentioned, some tools have been developed to focus on specific areas on the simulation pipeline.

- **Eclipse SUMO** [17] – SUMO stands for Simulation of Urban MObility and it is an open-source microscopic traffic simulator. SUMO allows for the modelling of various kinds of traffic systems such as public transport vehicles and other road users including pedestrians. Many of the end-to-end simulation tools previously mentioned only possess simplistic traffic models which may not be representative of certain real-world road networks. External traffic simulators are useful in these cases whereby an AV has to be simulated in a realistic environment with significant entropy from other road users.
- **PTV VISSIM** [18] – VISSIM is a commercial microscopic traffic simulator with similar functionalities to SUMO but with some notable differences [19]. The car following model used is different along with more realistic vehicle dynamics. There is also support for two-wheeled vehicles along with trams and pedestrians.
- **Ansys AVxcelerate Sensors** [20] – AVxcelerate is a sensor simulation software which enables realistic physics-accurate modelling of AV sensors such as LiDAR, camera and RADAR. Sensors can be modelled according to the respective manufacturer's specifications and AVxcelerate will be able to simulate their response when placed in a 3D environment However, AVxcelerate is not a standalone tool and will need to be used in conjunction with a main simulator such as CARLA.
- **MSC Adams** [21] – Adams is a multibody dynamics simulation tool. Unlike many rigid body vehicle dynamics tools which are commonly available and/or integrated with AV simulation tools, Adams allows for complex multibody physics. It is more general purpose than most vehicle simulators and hence is quite powerful from a system engineering standpoint. Being under the MSC software umbrella, Adams integrates well with VTD.
- **Mechanical Simulation CarSim** [22] – CarSim uses parametric math models to reproduce system-level vehicle dynamics behavior for passenger cars and light trucks. It is a standalone vehicle dynamics simulation tool which can integrate with other simulators such as CARLA. CarSim supports built-in scripting apart from integration with MATLAB, Visual Basic and C/C++.

IV. Fidelity

The primary scientific challenge that simulators face is their fidelity. A virtual simulation can provide an insight into the real world if, and only if, the tools employed are high-fidelity surrogates of the real counterpart.

A. Components

AV simulators are complex systems, and many of their different components are affected by this challenge.

a) HD Maps: Digital twins are beneficial for a few applications, such as testing the AV deployment in specific areas and ViL testing. Nevertheless, many other testing modalities are still feasible in maps not based on real roads.

b) Vehicle Dynamics - Road surface: The interaction of car momentum, tire rolling resistance coefficient, engine power, and asphalt surface is a complex system. An accurate model is required to properly validate any component that

interacts with the vehicle controls, i.e., any active safety feature and AVs.

c) Sensors: Sensor models are employed to generate Synthetic Data (e.g., images or point clouds). This component is also exploited in many applications to augment the learning procedure of the Perception Algorithm.

d) 3D Models and Materials: Sensors generate data by reacting to the physical properties of the surrounding. For example, cameras generate images by employing photosensitive sensors that respond to light. Similarly, Virtual Sensors react to the Virtual Environment. It follows that the resulting synthetic data's quality, fidelity, and value inherently rely on the virtual environment's quality.

B. Sensor Physics

Each sensor detects different physical aspects of the surrounding. Thus, each sensor type presents unique challenges due to its physics. In this subsection, we briefly present the phenomena that differentiate the most common AV sensors.

a) Camera: Camera images are affected by many artifacts and alterations. Lens distortions, chromatic aberrations, and vignetting are some of the optical effects. Moreover, since images provide direct visualization of the objects in the environment, even minor details are captured. 3D models, environmental lighting, shadows, and surfaces' textures and materials are crucial elements that determine the quality of an image.

b) LiDAR: LiDAR generates point clouds by measuring depth via the time of flight of laser pulses. Thus, the ray tracing framework is a natural solution to model LiDARs. LiDARs are very sensitive to weather conditions such as rain or fog [23].

c) RADAR: RADAR in the physical domain relies on radio waves, but a real-time virtual implementation of the RADAR equation is computationally expensive. Thus, ray tracing is often adopted to emulate radar physics. The most relevant physical phenomena are multipath reflections and interference. These weaknesses may lead to ghost objects and clutter. Moreover, the Radar cross-section (RCS) of the targets, i.e., a measure of how detectable they are, is affected by many physical properties of each target: the material, the size, the shape, and the relative angles of the wave [23].

C. Validation

While developing a high fidelity component is an ongoing challenge, their validation is also not trivial. A virtual simulation always has to deal with a trade-off between optimization and realism. The abstractions, models, and equations that govern the unfolding of simulation steps are simplifications of the real world. Thus, a degree of approximations is expected, accepted, and understood.

In literature, this problem is often approached by statistically comparing the output of the real component against the model's output. [24], [25]. Sensor specific examples can be found for RADAR [26] and LiDAR [27], [28].

V. Recent trends

In this section, we present solutions that are not yet fully adopted by mainstream simulators.

A. Open interoperability standards

Given the fact that different AV simulators and their modules have unique strengths and weaknesses that can be collectively exploited, purpose-driven co-simulation solutions are increasingly being adopted. However, such approaches often require significant engineering effort to achieve good interoperability.

The Association for Standardization of Automation and Measuring Systems (ASAM) is a non-profit organization which promotes standardization for toolchains in automotive development and testing. The ASAM Open standards span the major areas where work still needs to be done for interoperability between simulation tools. The standards are described in detail below:

- **OpenDRIVE** [29] – OpenDRIVE is a format that semantically describes the road networks of the AV's static driving environment, with an extensible markup language (XML) schema. OpenDRIVE files (`.xodr`) can be generated using various tools, both commercial and open-source, and can be used as input to simulation tools.
- **OpenSCENARIO** [30] – OpenSCENARIO defines a file format to describe complex scenarios involving multiple actors. These may include their movements, trajectories, velocities, and so on. The data for maneuver descriptions are organized in a hierarchical structure and serialized in an XML file format (`.xosc`).
- **OpenCRG** [31] – CRG stands for Curved Regular Grid which is a format to describe road surfaces. CRG files (`.crg`) can store high-precision elevation data from road surface scans and allows for the realistic rendering of 3D road surfaces. This is particularly useful for virtual simulations requiring a high fidelity of vehicle dynamics.
- **OpenODD** [32] – OpenODD aims to provide a format which can describe the Operational Design Domain (ODD) of an AV. It is a concept and not a finalized standard yet.
- **OpenLABEL** [33] – OpenLABEL is a standard currently under development, that proposes a standardized JSON schema for multi-sensor data labeling as well as for scenario tagging. It is expected to include definitions for labeling different kinds of data such as 2D and 3D bounding boxes, the rotation of 3D bounding boxes, semantic segmentation of images and point clouds.
- **Open Simulation Interface (OSI)** [34] – OSI aims to provide direct compatibility between simulation tools and automated driving software and/or functions. It defines standard interfaces for the connection of sensors and exchange of data. OSI utilizes an object-based environment description using the message format of the Protocol Buffers [35] library developed and maintained by Google.

B. Perception Error Models

Perception Error Models (PEMs) [36] are models that describe, and generate, the perception errors that affect an AV. This tool can be integrated into a simulation pipeline to inject perception errors without requiring physics-based sensor models, high-fidelity 3D environments, and object models. Thus, this approach is less computationally expensive and allows more control over the injected perception errors. PEMs model errors at the object list level, i.e., object detections and the noise on their parameters. Hoss et al. [37] provide an extensive review of methodologies suitable for the modeling task. This field is still under-explored, but a few relevant examples are [38], [39] for the modeling part. Instead, we can find a demonstration of their employment in a simulation pipeline for virtual testing in [36], [40], [41].

PEMs, however, do not substitute the other primary application of sensor models and synthetic sensor data, i.e., augmenting the learning task of perception algorithms.

VI. Conclusions

In this paper, we have discussed the major challenges in AV virtual testing. To focus on challenges unique to virtual testing, we omit discussion on the well-known general testing challenges, such as test outcome evaluation, safety metrics, or acceptance thresholds. AV simulators provide many advantages, but their limitations reduce their convenience and effectiveness in AV validation. However, availability of open standards and improvements in fidelity can potentially facilitate their larger-scale adoption. In fact, enhanced and easier interoperability may help overcome the weaknesses of individual simulators, so that we can employ different high-fidelity tools dedicated to specific system components.

References

[1] N. Kalra and S. M. Paddock, *Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?* Santa Monica, CA: RAND Corporation, 2016.

[2] SAE, "J3016 standard: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," 2021.

[3] M. Benz, "Mercedes Drive Pilot 2, group.mercedes-benz.com/innovation/case/autonomous/drive-pilot-2.html," 2021.

[4] D. C. Gross *et al.*, "Report from the fidelity implementation study group," in *Fall Simulation Interoperability Workshop Papers*, 1999.

[5] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1821–1827.

[6] Epic Games, "Unreal Engine 4, unrealengine.com," 2020.

[7] Unity Technologies, "Unity Engine, unity.com," 2022.

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

[9] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," *arXiv:2005.03778*, 2020.

[10] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu Apollo EM Motion Planner," *arXiv: 807.08048*, 2018.

[11] 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 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 287–296.

[12] MSC, "Vires VTD, vires.mscsoftware.com," 2022.

[13] IPG, "IPG CarMaker, ipg-automotive.com/en/products-solutions/software/carmaker/," 2022.

[14] Siemens, "Siemens Simcenter Prescan, plm.automation.siemens.com/global/en/products/simcenter/prescan.html," 2022.

[15] rFPro, "rFPro, rfpro.com," 2022.

[16] NVIDIA, "NVIDIA DRIVE Sim and Omniverse, nvidia.com/en-sg/self-driving-cars/simulation/," 2022.

[17] Eclipse, "Eclipse SUMO, eclipse.org/sumo/," 2022.

[18] PTV, "PTV VISSIM, myptv.com/en/ptv-vissim," 2022.

[19] M. Maciejewski, "A comparison of microscopic traffic flow simulation systems for an urban area," *Transport Problems : an International Scientific Journal*, vol. 5, 01 2010.

[20] Ansys, "Ansys AVxcelerate Sensors, ansys.com/products/av-simulation/ansys-avxcelerate-sensors," 2022.

[21] MSC, "MSC Adams, mscsoftware.com/product/adams," 2022.

[22] Mechanical-Simulation, "CarSim, carsim.com/products/carsim/index.php," 2022.

[23] E. D. Marti, M. A. de Miguel, F. Garcia, and J. Perez, "A Review of Sensor Technologies for Perception in Automated Driving," *IEEE Intelligent Transportation Systems Magazine*, 2019.

[24] A. Schaermann, A. Rauch, N. Hirsenkorn, T. Hanke, R. Rasshofer, and E. Biebl, "Validation of vehicle environment sensor models," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 405–411.

[25] P. Rosenberger, J. T. Wendler, M. F. Holder, C. Linnhoff, M. Berghöfer, H. Winner, and M. Maurer, "Towards a generally accepted validation methodology for sensor models-challenges, metrics, and first results," 2019.

[26] M. Holder, P. Rosenberger, H. Winner, T. D'hondt, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, Z. Slavik, O. Bringmann *et al.*, "Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving," in *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2616–2622.

[27] P. Rosenberger, M. Holder, S. Huch, H. Winner, T. Fleck, M. R. Zofka, J. M. Zöllner, T. D'hondt, and B. Wassermann, "Benchmarking and functional decomposition of automotive lidar sensor models," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 632–639.

[28] P. Rosenberger, M. F. Holder, N. Cianciaruso, P. Aust, J. F. Tamm-Morschel, C. Linnhoff, and H. Winner, "Sequential lidar sensor system simulation: a modular approach for simulation-based safety validation of automated driving," *Automotive and Engine Technology*, vol. 5, no. 3, pp. 187–197, 2020.

[29] ASAM, "ASAM OpenDRIVE, asam.net/standards/detail/opendrive/," 2022.

[30] ——, "ASAM OpenSCENARIO, asam.net/standards/detail/openscenario/," 2022.

[31] ——, "ASAM OpenCRG, asam.net/standards/detail/opencrg/," 2022.

[32] ——, "ASAM OpenODD, asam.net/standards/detail/openodd/," 2022.

[33] ——, "ASAM OpenLABEL, asam.net/standards/detail/openlabel/," 2022.

[34] ——, "ASAM OSI, asam.net/standards/detail/osi/," 2022.

[35] Google, "Google Protocol Buffers, developers.google.com/protocol-buffers," 2022.

[36] A. Piazzoni, J. Cherian, M. Slavik, and J. Dauwels, "Modeling perception errors towards robust decision making in autonomous vehicles," in *Proc. of the 29th International Joint Conference on Artificial Intelligence, IJCAI-20*, 7 2020, pp. 3494–3500, main track.

[37] M. Hoss, M. Scholtes, and L. Eckstein, "A review of testing object-based environment perception for safe automated driving," *arXiv:2102.08460*, 2021.

[38] E. L. Zec, N. Mohammadiha, and A. Schliep, "Statistical sensor modelling for autonomous driving using autoregressive input-output hmms," in *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1331–1336.

[39] H. Arnelid, E. L. Zec, and N. Mohammadiha, "Recurrent conditional generative adversarial networks for autonomous driving sensor modelling," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1613–1618.

[40] P. Mitra, A. Choudhury, V. R. Aparow, G. Kulandaivelu, and J. Dauwels, "Towards modeling of perception errors in autonomous vehicles," in *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3024–3029.

[41] J. Sadeghi, B. Rogers, J. Gunn, T. Saunders, S. Samangooei, P. K. Dokania, and J. Redford, "A step towards efficient evaluation of complex perception tasks in simulation," *arXiv preprint arXiv:2110.02739*, 2021.