IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 11 February 2023, accepted 13 March 2023, date of publication 17 March 2023, date of current version 24 March 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3258681

RESEARCH ARTICLE

PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration

RAHUL RAZDAN[1], (Senior Member, IEEE), MUSTAFA İLHAN AKBAŞ[2], (Member, IEEE), RAIVO SELL[3], (Member, IEEE), MAURO BELLONE[4], (Member, IEEE), MAHESH MENASE[5], AND MOHSEN MALAYJERDI[3]
[1]Florida Polytechnic University, Lakeland, FL 33805, USA
[2]Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA
[3]Department of Mechanical and Industrial Engineering, Tallinn University of Technology, 19086 Tallinn, Estonia
[4]FinEst Smart City Centre of Excellence, Tallinn University of Technology, 19086 Tallinn, Estonia
[5]Acclivis Technologies Pvt. Ltd., Pune, Maharashtra 411038, India

Corresponding authors: Rahul Razdan (rrazdan@floridapoly.edu) and Mustafa İlhan Akbaş (akbasm@erau.edu)

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

ABSTRACT Validation and Verification (V&V) of Artificial Intelligence (AI) based cyber physical systems such as Autonomous Vehicles (AVs) is currently a vexing and unsolved problem. AVs integrate subsystems in areas such as detection, sensor fusion, localization, perception, and path planning. Each of these subsystems contains significant AI content integrated with traditional hardware and software components. The complexity for validating even a subsystem is daunting and the task of validating the whole system is nearly impossible. Fundamental research in advancing the state-of-the-art for AV V&V is required. However, for V&V researchers, it is exceedingly difficult to make progress because of the massive infrastructure requirements to demonstrate the viability of any solution. This paper presents PolyVerif, the world's first open-source solution focused on V&V researchers with the objective of accelerating the state-of-the-art for AV V&V research. PolyVerif provides an AI design and verification framework consisting of a digital twin creation process, an open-source AV engine, access to several open-source physics based simulators, and open-source symbolic test generation engines. PolyVerif's objective is to arm V&V researchers with a framework which extends the state-of-the-art on any one of the many major axes of interest and use the remainder of the infrastructure to quickly demonstrate the viability of their solution. Given its open-source nature, researchers can also contribute their innovations to the project. Using this critical property of open-source environments, the innovation rate of the whole research community to solve these vexing issues can be greatly accelerated. Finally, the paper also presents results from several projects which have used PolyVerif.

INDEX TERMS Autonomous vehicles, validation and verification, modeling and simulation, artificial intelligence.

I. THE AI V&V RESEARCH CHALLENGE

Autonomous Vehicles (AVs) offer significant benefits in terms of safety, access, and utilization for transportation services. The Artificial Intelligence (AI) paradigm around Machine Learning (ML) has been the key enabling technology for AVs. Thus, by definition, validation and verification of AVs involves the ability to validate AI/ML components.

The ML technique is the third major phase in the long history of AI [1]. In its first phase, the center of activity was on symbolic AI which offered unique solutions to problems such as chess, but broad applicability was limited by the

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

TABLE 1. Contrast of conventional and machine learning algorithms from V&V perspective.

Conventional software	AI/ML software	Comment
Logical Theory	No Theory	In conventional algorithms, one needs a theory of operation to implement the solution. ML algorithms can often "work" without a clear understanding of exactly why they work.
Analyzable	Not Analyzable	Conventional algorithms are encoded in a way one can see and analyze the software code. Most validation and verification methodologies rely on this ability to find errors. ML algorithms offer no such ability, and this leaves a large gap in validation.
Causal	Correlation	Conventional algorithms have built in causality and ML algorithms discover correlations. The difference is important if one wants to reason at a higher level.
Deterministic	Non-Deterministic	Conventional algorithms are deterministic in nature. In contrast, the combination of training and inference generates a situation where ML algorithms are fundamentally probabilistic in nature.
Known Computational Complexity	Unknown Computational Complexity	Given the analyzable nature of conventional algorithms, one can build a model for computational complexity. That is, how long will it take the algorithm to work. For ML techniques, no generic method exists to evaluate computational complexity.

exponential computational nature of the underlying algorithms [2]. The next phase moved to the domain of ''expert systems'' with general rule solvers which solved the exponential explosion problem, but left the burden of building large libraries of domain specific rules [3] to armies of experts. Today, AI has been redefined again with the techniques of ML and data-driven algorithms.

Recognizing the inherent limitation of a process which involves human ''programming,'' ML has built a ''learning'' paradigm (see Fig. 1). In this paradigm, there is a period of training where the AI machine ''learns'' from data to build its own rules. The process of learning is defined by employing traditional optimization algorithms which minimize some notion of error.

AI/ML methods are the next leap in building function with an impact as dramatic as the move from hardware to software. For hardware, implementing a function requires a long design and manufacturing process, and functional updates require updates to the physical device. The software paradigm allowed for building unique system functionality without the need for changing physical assets. Further, breakthrough concepts such as the concept of a computer architecture allowed for the update of the underlying hardware without the need to change the software. However, while building conventional software development minimizes the physical constraints of hardware, there is a requirement to have armies of software developers to design and implement the functionality. AI/ML paradigm offers a jump in productivity because humans, armed only with data sources and metrics, can employ machines to automatically build functionality through the inference software.

The process of simply giving data to an engine and automatically developing an algorithm has enormous potential to solve interesting problems. This technique does seem to ''work'' in areas such as vision or natural language processing where conventional algorithmic solutions have been difficult. However, this ''AI Inference Software'' is not quite the same as conventional human generated software. Table 1 contrasts conventional and AI generated software.

This contrast between AI and conventional software components has deep implications in many areas, but especially in the area of validation and verification (V&V). V&V is a critical feature because the usefulness of any solution reduces dramatically if one cannot validate its correctness. The V&V constraint is especially important in the context of safety critical situations.

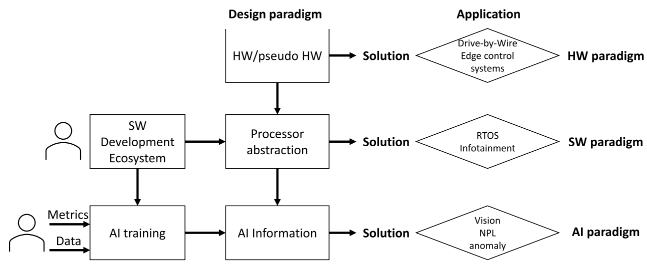


FIGURE 1. AI/ML as a fundamentally new programming paradigm.

For safety critical systems, there has been a historical adoption of V&V methods to design methods, and many of the innovations have been led by the airborne domain. In the hardware dominated time-frame, safety focused airborne systems were built upon a trust framework to verify, validate, and certify airborne systems is a series of laws, orders, and best-practice guidelines used to demonstrate conformance with airworthiness standards [4]. Critical aspects of this framework are:

- **System Design Process**: A process-oriented structured development assurance for these complex systems with safety certification as part of the integrated development process;
- **Formalization**: Formal definitions of system operating conditions, functionality, expected behaviors, risks, and hazards which must be mitigated; and
- **Lifecycle**: Lifecycle management of components, systems, and development systems.

The driving methodology was to define the system design, formalize the expected behavior as well as likely issues, and make sure one understands the impact over the life of the product very carefully and formally.

When the power of the conventional software paradigm was introduced, safety critical V&V was adapted by maintaining the original system design paradigm, but now system components could also be software. Each software component maintained the same overall structure of fault analysis, lifecycle management, and system design hazard analysis. However, the underlying details had to be extended. As an example, an associated airborne standard (DO-178C) defining the "Software Considerations in Airborne Systems and Equipment Certification" was created [5]. The standard updated the notion of hazard from physical failure mechanisms to functional defects because software does not suffer from physical process-driven reliability degradation. Also updated were the concepts of lifecycle management, which was changed to reflect the conventional software development process. Finally and most importantly, Design Assurance Levels (DALs) were used to manage the inherent risk from software components. Through this process, software components were integrated into the system design, functional allocation, performance specification, and V&V process.

AI/ML design requires the next jump in V&V technology. Similar to the introduction of software, AI/ML components can be defined as simply software components which are programmed with data. The above simple statement is very powerful because it allows AI/ML components to inherit the time tested infrastructure built over the last 30 years for integrating software components into safety critical systems. However, now, one must manage the issue of how to handle the fact that we have a data generated "code" vs conventional programming code. In the world of validation, this difference impacts validation in three significant aspects: coverage analysis, code reviews, and version control. Table 2 offers the contrast and V&V challenges.

For well defined systems where the system level abstractions for correctness are available, AI/ML components significantly increase the difficulty of intelligent test generation, coverage analysis, and V&V closure. With a great deal of difficulty, it is still possible to follow a structured process to make significant progress.

However, one of the most compelling uses of AI/ML is to employ it in situations where the specification of the system is not well defined or not viable using conventional programming. In these Specification-Less /ML (SLML) situations, not only is building interesting tests difficult but evaluating the correctness of the results creates further difficulty. Unfortunately, most of the major systems in AVs fall into this category of system function and AI usage. Overall, AI/ML components offer the promise to add a fundamentally new tool in system design, but also introduce unique and interesting V&V challenges. Solving these challenges offers the motivation for the PolyVerif work proposed in this paper.

II. RELATED WORK

There have been various efforts to attack the AV validation and verification problem in the literature. Huang et al. [6] performed one of the earlier reviews of the space of AV testing methods. Huang included the applicability of V&V tools such as test driving, vehicle-in-the-loop (VEHIL) simulation, model-based testing, rapid control prototyping, hardware-in-the-loop simulation and software-in-the-loop simulation. The review showed the importance of different validation types and the creation of an integrated testing methodology. However, the review did not address the unique challenges of validating AI V&V components.

Riedmaier et al. [7] developed a novel taxonomy for the scenario-based approaches in their survey of scenario-based safety assessment methods for AVs. Ma et al. [8] presented a review of validation and verification methods developed specifically for decision making and planning systems in CAV. This review compared and evaluated scenario-based testing, fault injection testing, and formal verification using multiple criteria. Then the results are used to recommend validation and verification approaches for different portions of the decision making systems in CAV. These approaches advanced the state-of-art, but unfortunately leveraging this work at a broader level is very difficult without an open-source paradigm. In contrast, PolyVerif provides a framework to implement and test each of these specific approaches which can be leveraged more broadly.

Both industry and research groups have been developing tools for testing and verifying their AVs. For instance, Waymo tests each of their primary subsystems, the vehicle, their hardware, and their software, individually as well as together using simulation, closed-course testing, and real world driving [9]. The scenarios they focus on are based on training the vehicle behavioral competencies. PreScan is a popular commercial product which allows the user to define scenarios and execute them in its runtime environment with lots of different 3rd party integration [10]. SynCity is a simulator with an emphasis on realistic scenarios for sensor testing and ML training, working at the communication protocol level for realistic sensor feedback [11]. Carcraft combines virtual maps with sensor data from the real world to recreate scenarios that their shadow driving fleet have encountered, fuzzing them for variation [12]. All of the above approaches are proprietary in nature and extremely difficult for V&V researchers to use based on cost and access issues.

Open-source AV stacks are also available, such as the one from Baidu [13], but with a focus on design components. PEGASUS, a joint effort among multiple groups from science and industry in Germany, aims to develop a full toolchain for AV verification, looking at traditional vehicle verification and new innovations in the field for inspiration [14]. Similarly, CARMA [15] provides an open source environment for traffic management in the context of Connected and Automated Vehicles (CAV). Vires developed several open-source standards to define scenarios, roads, and vehicle dynamics, as well as Virtual Test Drive, a tool-chain for driving simulation that uses those standards [16]. These platforms offer reasonable open-source environments for design, but none focus on the fundamental issues of individual AV Validation. As Kang et al. [17] point out, there are also a large number of relatively disconnected and unaligned data-sets available. A unified V&V platform such as PolyVerif is a robust and reliable platform which can connect these data-sets

TABLE 2. Comparison and challenges in V&V for AI-based software and conventional software.

V&V Technique	Software	AI/ML
Coverage Analysis	Code Structure provides basis of coverage	No structure
Code Reviews	Crowd source expert knowledge	No Code to Review
Version Control	Careful construction/release based on source code which is version controlled	Data updates change the inference function, so data sets must be version controlled. This is challenging in real-life environments, but especially so for streaming applications.

into active live simulation environments for use by AV V&V researchers.

Traffic simulators are used in CAV validation and verification to implement the logic of and simulate traffic at the micro or macro level. MATSIM is a well-known open-source microscopic traffic simulator, which is built for large simulations as it utilizes traffic simulation, large-scale computation, and agent based modeling [18]. PolyVerif uses SUMO, which is a well-known open-source traffic simulator with a low-overhead for running city-size road networks at a microscopic level [19]. It has been used by both academia and industry in various projects [20], [21], [22].

Seshia et al. [23] identified environment modeling, formal specification, modeling of learning systems, scalable formal engines and correct-by-construction design as the main challenges of applying formal methods to AI validation and verification. PolyVerif provides the capability to test approaches in practice for solving each of these challenges for CAVs.

III. OPEN RESEARCH PROBLEMS FOR AV V&V

AI components form the critical enabling technology which has demonstrated the potential viability of AVs. Fig. 2 shows the major pieces of an AV Application architecture which include components such as detection, perception, localization, and path planning. Modern AV systems integrate many AI/ML models in all these functions, and, in most situations, the form of these applications is in the SLML category.

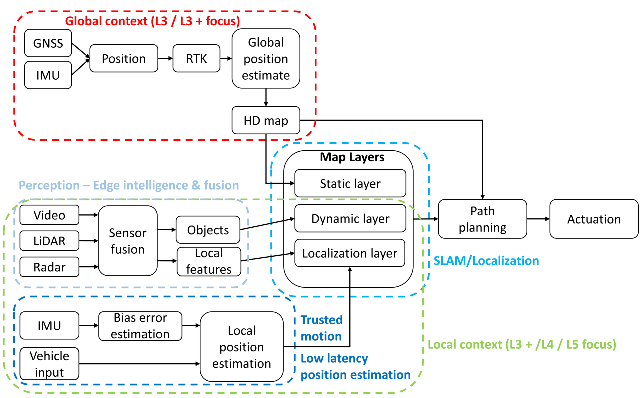


FIGURE 2. Generic AV software components highlighting a typical AV application architecture and challenges.

This combination of elements leads to very significant V&V challenges. As outlined in [24], the nature of these challenges can be outlined at three levels of size: AI components, AI subsystems, and AI systems.

For AI components, the research challenges are as follows:

1) **Correctness**: If the AI model does not have a clear specification, how does one determine correctness?
2) **Completeness**: Without any view of the internal structure of the algorithm, how can one determine the completeness of the testing task?
3) **Training Data Analysis**: Since data is the seed which generates the algorithm, verification of the training data set for completeness is necessary. What is the correct methodology to do this for large, and, often streaming, data sets?
4) **Safety Level Analysis**: If the nature of the AI component correctness is fundamentally probabilistic, what is the methodology to protect the result from feeding highly hazardous actuation functions?

Above the component level at the AI subsystem level, abstraction is a very powerful tool which has been used in a wide variety of fields to manage complexity. As an example, it is a fundamental part of building and validating complex semiconductors chips with billion transistors. For semiconductors, well defined abstraction layers (layout, schematic, Register Transfer Level, etc.) transition from the deep physics of a sub-micron transistor to the abstraction provided by a computer architecture. All the stages of the design pipeline have an accompanying V&V methodology which builds an inductive argument for the correctness of the entire chip. Since it is impossible to simulate a modern semiconductor at the device physical level of abstraction, this methodology is critical for validating large complex chips with first-pass level of correctness.

For AI systems, the critical pipeline for AVs consists of detection, perception, location services, path planning, and actuation [25]. At a higher level, the AV is building an internal view of the external world which is reasoning at different levels of abstraction. However, clear specifications do not exist for the intermediate levels of abstraction, and there is currently no V&V infrastructure which accompanies each abstraction level of design. As an example, it is not uncommon to feed "photo-realistic" scenario libraries [26] in an attempt to validate the whole AV stack. This approach is problematic on many fronts:

1) **Computational Performance**: Without decomposition and testing at higher levels of abstraction, the cost of simulation performance over the full spectrum of the Operational Design Domain (ODD) becomes the limiting factor.

2) **Diagnosability**: Diagnosing errors without the use of internal data is very difficult.
3) **Completeness**: Each of the abstraction levels work in a subset of the problem, and, in fact, generate constraints which can aid with the V&V task. As an example, the detection phase works strictly in the world of physics which is a property very useful for validation.

A sampling of the open research questions in AV V&V are:

1) What are the right intermediate abstraction levels? How can they be specified mathematically? What is the accompanying V&V infrastructure that can break down the overall safety problem?
2) At the system level, how can one build an inductive argument for correctness based on the intermediate validation stages at the component and subsystem level.
3) For sensor modalities, camera output combined with photogrammetry techniques are reasonably well developed for vision. However, similar techniques for building accurate digital twins with LiDAR, RADAR, and other sensor modalities is not well developed. How can one build the V&V mechanism to test the appropriate "corner" cases connected to sensor modalities?

Given the breadth and depth of the AV stack, it is exceedingly difficult for any AV V&V researchers to build reasonable experiments or validate correctness. This is the primary motivation for developing the open-source PolyVerif environment.

IV. THE PolyVerif FRAMEWORK

The open-source development paradigm is a powerful mechanism to crowd-source innovation. One of the critical techniques which allows for progress is that an individual or a small group can attack a specific part of the problem while relying on the remainder of the open-source infrastructure for the remaining functionality. If the functionality is useful, it can be integrated into the open-source stack and the whole stack progresses. To accelerate the rate of progress of AV V&V, PolyVerif offers a full open-source AV design and validation simulation based infrastructure upon which researchers can build innovative capability. The critical component pieces of the PolyVerif Open-Source stack are:

1) **AV Engine**: Included within Polyverif is the Autoware [27] open-source AV engine. Because it is open-source, internal abstraction points from the engine are easily available to researchers. In addition, an API that connects the AV Engine to the rest of the environment has been constructed by PolyVerif such that other AV engines can be easily integrated.
2) **Simulation**: Included with Polyverif are native integrations to the open-source SUMO [28], LGsim [29], Tier IV [30] and CARLA [31]. Each of these simulators provides different values for validation. In addition, an API that connects the simulator to the rest of the environment is specified such that other simulators can easily integrate.
3) **Test Generation**: Included with PolyVerif is a native integration with Scenic [32], an open-source abstract test generation system. In addition, an API that connects the test generation engines is available for integration with other engines such as OpenScenario 2.0 [33].

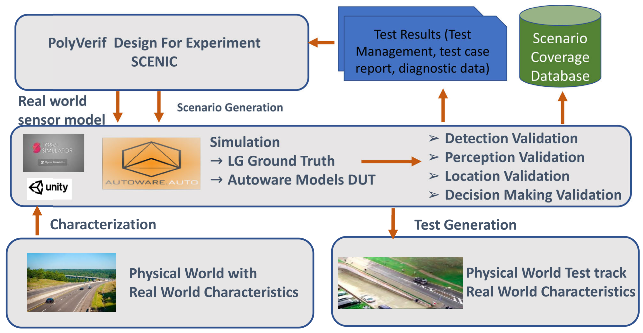


FIGURE 3. PolyVerif layers.

On top of these standalone components, PolyVerif provides:

1) **Digital Twin Flow**: A tested digital twin flow which can build a digital twin in the simulators based on google map images and further refined with direct drone sensor data is provided. The intermediate parts of the flow are available in the environment for easy manipulation by researchers.
2) **Path to Physical Testing**: Since the TalTech team has built an implementation with the iseAuto project [34]. It is possible to collaborate with the TalTech research team to test updates to the open-source software stack on a physical implementation.

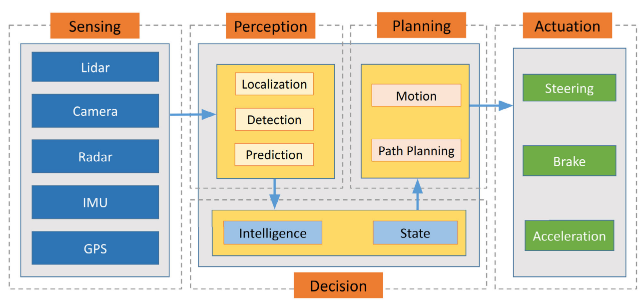


FIGURE 4. Autonomous vehicle software stack.

In terms of modeling abstraction, the Autoware AV stack (or any AV stack) is operating in a conventional Newtonian physics universe. To be useful, any simulation environment must model key concepts such as momentum, light processing, sound dynamics, and more. These concepts can be modeled at various levels of fidelity with a tradeoff between accuracy and simulation performance (Fig. 3). At a component level, the internal useful abstractions of the major pieces of the Autoware AV stack are shown in Fig. 4 and can be listed as:

1) **Localization**: Localization takes sensory input (GPS, IMU) to generate an abstract positioning of the unit under test on a global map. Models of noise can be introduced to test the robustness of the localization engines.
2) **Detection**: This stage accepts sensor inputs, and the outputs are abstract objects in 3-D space. Thus, it is possible to test Detection functionality independently of the rest of the stack in simulation under a variety of conditions.
3) **Prediction**: This unit takes the result of Detection and annotates abstract objects with predicted future behavior. Thus, it is possible to generate synthetic Detection output to test the Prediction engine individually.
4) **Path Planning**: Path Planning consumes abstract objects from Perception to build an actuation function and a predicted future path. Again, synthetic data can be used to test the path planning function individually.

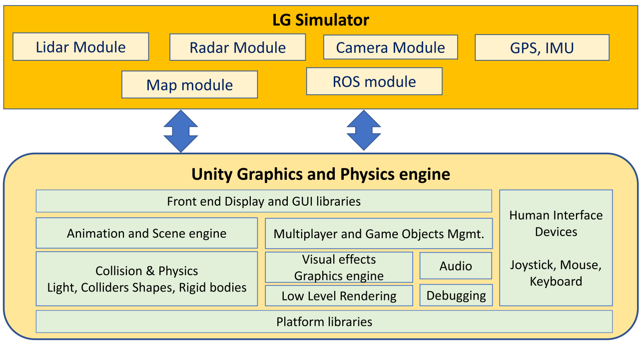


FIGURE 5. Unity Graphics engine and LG simulator communication block diagram.

Using the machinery described above, one can construct many interesting designs for experiment flows. As an example, one can conceive of the following design for experiment around detection.

1) A simulator which contains a model of the world and simulates this world in the Newtonian realm. The simulator contains all the Ground Truth data.
2) The AV Stack receives sensor data from the simulator (through characterized sensor models) and builds an internal representation of the external world.
3) At the interface between the simulator and the sensors, a functional noise generation function can be inserted to model interference (rain, snow, etc).
4) The abstract objects at the output of the Detection stage can be monitored for divergence between ground truth and AV Stack version of the ground truth.
5) Functional checkers can be defined to determine the range of severity required to consider the test a pass/fail at various phases (detection abstract object phase, perception abstract object phase,path planning phase).
6) The device under test (DUT) can be placed in various scenarios in the environment and this process can be managed by sophisticated test generation software.

Fig. 4 shows a block diagram view of the software critical pieces for the above flow. In summary, once a simulation, AV stack, and design for experiment machinery is available, it is possible to quickly build and evaluate a variety of V&V research solutions. Fig. 3 summarizes the various PolyVerif Layers and flows.

V. PolyVerif SOFTWARE ENVIRONMENT

To be useful, all the AV components within PolyVerif must be organized in a well structured software environment. As a base layer, PolyVerif builds a simulation paradigm around a Unity framework (see Fig. 5). Unity has modeling of rigid bodies, modeling of visual and acoustic waves, a rich software environment for programmable construction of worlds, and of course world-class rendering. Thousands of man hours have been invested in Unity for large end markets in computer gaming and movie production.

While Unity is an excellent starting point, AV Simulation also requires modeling of the other sensor types and of location services. In the PolyVerif environment, these can be added as required while benefiting from the rest of the infrastructure with convenient software APIs. In addition, working at the detailed low-level 4-D world of Unity for AV V&V is very cumbersome, so there are requirements for automation for critical tasks such as world building, scenario specification, and test synthesis.

A. WORLD BUILDING

Constructing comprehensive "simulation" worlds is an exceedingly difficult process. In the realm of computer games, designers can easily spend millions of dollars to build interesting worlds. Fortunately, AV V&V can use the real world as a great aid in doing this task. Fig. 6 shows the fundamental flow.

In this flow Google Map, Earth, and Street Views can be used to build an initial 3-D rendition. While very powerful, and an excellent starting point, additional characterization information is required to build a model with higher fidelity [35]. In the visual spectrum, this involves high fidelity cameras which capture the environment on drones or ground vehicles. PolyVerif has enabled a near automated flow with Agisoft Metashape [36] to enable the construction of higher fidelity models. The end result of all of this activity is the development of a simulation model which accurately processes the physics of the various sensor modalities. This area of digital twin generation is itself an active research area. As mentioned, researchers are exploring the methods for the inclusion of other sensor types (acoustic, LiDAR, RADAR, and more) and the required fidelity for the needs of the validation paradigm. PolyVerif is an excellent framework to explore these issues.

B. SCENARIO BUILDING

Once the world is built, one must construct test cases which give the unit under test a goal, and introduce other dynamic components into the test evaluation. The range of goals is

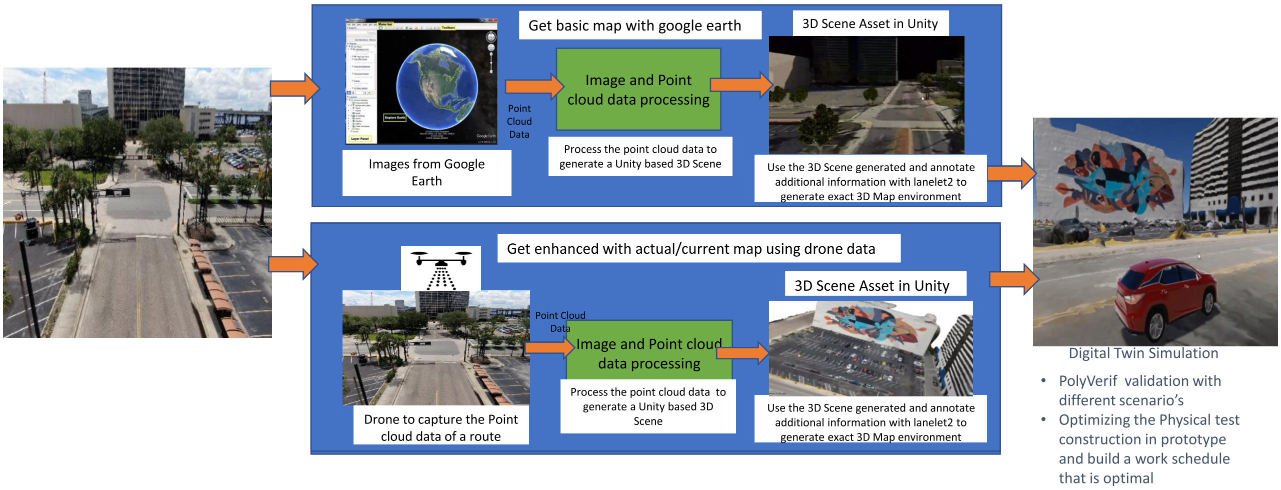


FIGURE 6. From physical track to digital twin creation block diagram.

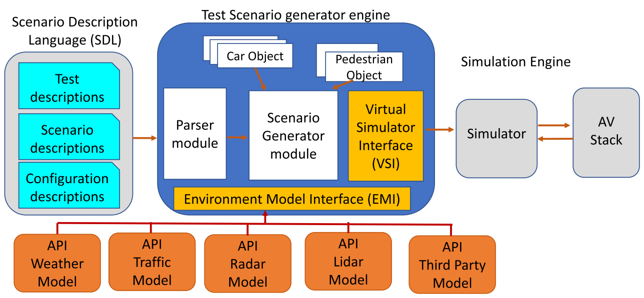


FIGURE 7. Test generation and validation software architecture for PolyVerif.

highly varied and the range of the behavior of dynamic components (cars, pedestrians, environmental weather, etc) is even larger. To efficiently manage this situation, there is a need for a symbolic test generation language which can specify critical pieces of a test, and even more importantly allow the automatic generation of "don't care" cases. As an example, one might specify a test around collision avoidance where the unit under test starts at a random starting velocity within a reasonable range, and must engage active braking independent of all the don't care conditions. This powerful concept is used extensively in the validation of digital systems and has been brought over to the cyber-physical world by the UC Berkeley Team with the Scenic capability [32], which is integrated into the PolyVerif platform [37], [38]. Within the industry, standardization efforts have gained traction in this area with OpenSCENARIO 2.0 [33].

C. TEST SYNTHESIS

Once simulation finds a potential issue, it is useful to be able to recreate the test in a physical environment. However, this is not an easy task because the simulation specifies a very particular environment which must be mimicked in real life. The methodology of recreating test cases, especially with particular sensor inputs and dynamic movements, is also a research area.

Finally, for a serious V&V task, one must build a design for experiment infrastructure which is programmatic in nature because of the automation required for a large number of tests. Key elements of the DoE flow mimic the process for any sophisticated large software project with elements such as:

1) Consistent Master Open-Source Repository: The Source code repository is maintained and available.[1] The code repository is consistently maintained with incremental releases with new feature additions and bug fixes.
2) Consistency in open-source licensing terms across the framework. The PolyVerif framework has Apache Version-2 license enabling users to use, modify and deliver as necessary. The Apache License explicitly allows you to copy and redistribute the covered product, without any license fees or royalties.
3) Consistency in installation, build, and execution methodology PolyVerif framework consists of multiple open source frameworks and codebases, so installation can become complex. Moreover due to different versions of software it can be confusing and difficult to deal with. To avoid this we have provided build scripts to ease the installation across Linux platforms. Apart from that our development team can assist you in enabling the framework and getting started.
4) Consistency in documentation for use of the framework with special emphasis on well-defined interfaces. The framework can be used using PythonAPI's and Scenic (Scenario description language) scenario generation library across different simulators. The documentation helps users to Kickstart the usage with:

[1] https://github.com/MaheshM99/PolyVerif

a) Installation guide.
b) Sample Test scenarios and scripts have also been provided using Scenic.
c) How to write Test cases and scenarios.

As might be expected, establishing this baseline involved significant engineering time as it is not trivial to establish this baseline across multiple active open-source projects. Given a well engineered, environment sophisticated Designs of Experiments (DoE) flows can now be constructed. Fig. 7 shows the software architecture for integrating all the pieces.

VI. PolyVerif PROJECTS

Using the PolyVerif Framework partner teams addressed their specific V&V research challenges. Current ongoing active projects with the PolyVerif include:

1) Digital Twin Model of AV route in Tallinn, Estonia with an Autoware engine [34], [39], [40].
2) Digital Twin Model of AV route in Jacksonville, Florida with a non Autoware engine [41].
3) Design for Experiment flows for detection, prediction, localization, path planning, and control by a design and validation services company for the automotive industry, Acclivis [42]. Table 3 shows the flows, the V&V focus point for the flow, and the resulting reports.
4) Abstracted Tests of all the currently tracked AV test cases from the state of California database of AV incidents by Embry-Riddle Aeronautical University [21], [43].

The results of all of these projects have now been added to the PolyVerif open-source environment. The power of this approach is that a third-party researcher can choose to use these environments to validate updates in design or validation technology for their domain. The detailed results of these efforts will be published by companion teams in subsequent research publications. In this paper, a deeper examination of the TalTech results is presented.

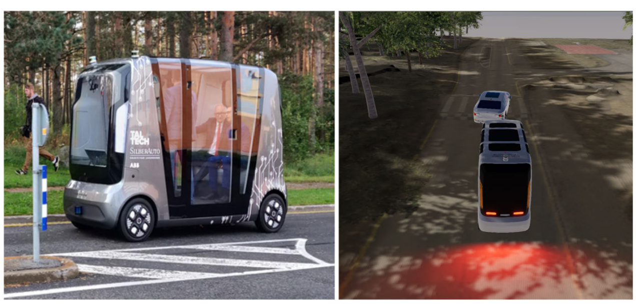


FIGURE 8. Vehicle digital twin representation. The real vehicle is on the left side, while its digital twin is on the right.

TalTech, a polytechnic university in Estonia, has active research in building an autonomous shuttle - iseAuto. The underlying software for the shuttle is based on the Autoware open-source framework. Similar to most of the AV industry, the team tested the vehicle with physical experimentation on an on-campus test track. Physical validation is a slow process which is expensive and incomplete. Thus, a deeper move to simulation as well as a more robust V&V environment was desirable and drove an engagement with PolyVerif.

In the first step, the TalTech team built a digital twin of the TalTech campus test track. A virtual copy of the vehicle was created and defined with the same sensor configuration as the actual vehicle [39] (see Fig. 8).

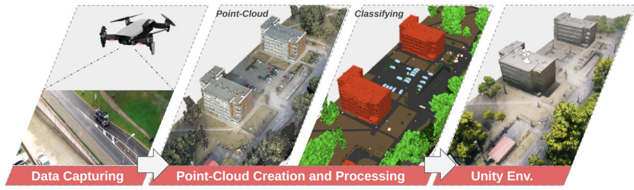


FIGURE 9. Digital environment creation steps from data capturing to final mesh reconstruction.

The shuttle kinematics and dynamics were 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 limited vision 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 are not in the other LiDARs' field of view. Processes such as calibration, filtration, and concatenation are performed on the LiDARs' point cloud for an optimized perception.

For simulations, real events and conditions were mimicked in order to generate, test, and evaluate reliable data sets. The validation group tested and validated in the PolyVerif framework using the custom open-source full-scale AV shuttle software stack built by TalTech.

The validation was carried out in two steps.

Step 1: An example scenario and the virtual environment model was created on a selected area in the city of Tallinn. The validation area was in an unstructured environment featuring cars and pedestrians in a semi-restricted area (the university campus).

Step 2: The Real-world validation was connected to the self-driving AV shuttle iseAuto. Developed scenarios and PolyVerif framework were validated with real equipment in real world.

The work was conducted in three stages:

Stage1: Digital Twin Route in Tallinn - Estonia
Stage2: Validation scenarios in PolyVerif
Stage3: Scenario Validation with the physical device

In stage one, a digital twin with model abstraction and characterization considerations was generated. This activity was carried out by doing the following activities: Aerial image capture, LiDAR mapping, post-processing of raw data, classification, and virtual environment creation (see Fig. 9).

In stage two, several selected scenarios were validated by applying the PolyVerif framework by building a simulatable model. The design of the experiment stage exercised the

TABLE 3. Design of experiment flow examples.

DOE Flow	Description	Reports generated
Detection	Do the sensors actually "see" the objects of interest?	· Object Detection: Success Rate · Object Detection ranges in AV stack
Perception	Are Objects Understood sufficiently for future behavior prediction?	· Correct object identification success rate · True positive/negative rate, and false positive/negative rate
Localization	Global and Relative Location	· Per Frame ego vehicle deviation · Max/Min/Mean deviation
Path Planning	Goal success with constraints	· Mission completion statistics · Obstacle avoidance Report · Vehicle Controls (brake, acceleration) testing reports
Control	Does the actual trajectory followed by the vehicle match the path to goal?	· Time-to-Collision (TTC) calculations · Real responses time in the simulator after object is detected · Response time available in AV stack after object is detected by perception module · Delay in responses time due to perception algorithm · Control – Brake properly applied/not applied (threshold based)

model in various configurations with an eye towards exposing "worst-case" conditions and showing the completeness of testing (coverage). The objective was to find a reliable way to discover edge case scenarios, and cover most of the high-risk scenarios in simulation.

Finally, in stage 3, the chosen scenario was run both in simulation, and the real world with the intention of finding inconsistencies between the two. Performance indicators were recorded and compared. For more in-depth diagnostics and comparison the autonomous driving, data were also recorded. The vehicle software was running on the ROS and the autonomous driving stack Autoware.ai which is a customized software architecture.

1) DESCRIPTION OF THE EQUIPMENT

Vehicle: iseAuto ver 1.2 no.3 (Level 4 AV, 6 seat minibus for urban mobility) powered by open-source ROS & Autoware. Technical parameters: Passenger capacity: 6; Speed: avg 10 km/h, max 50 km/h; Turning radius: 9 m; Main motor power: 47 kW; Battery: 16 kWh; Mass: 1250 Kg; Height: 2,4 m; Length: 3,6 m; Width: 1,50 mm;

Sensors: Main LiDARs Velodyne x 2; Side LiDARs x 2; Front safety LiDAR; Front cameras x 4; Back camera; RTK-GNSS + IMU;

2) SCENARIO

The validation scenario was TalTech campus test track. In this route, the most challenging situation was selected involving the intersection, and other cars. Such a scenario was chosen to include multiple aspects that are perceived to be challenging for AVs. The scenario includes different yielding situations, and a pedestrian crossing the street out of the crosswalk. The defined scenario is visualized in Fig. 10.

3) SIMULATION

SVL Simulator 2021.3 was used for the digital twin validation. The environment was created, and ported into the simulator, using a real 3D scans done using a commercial drone. Free software was used to derive 3D data from the drone captures. Based on the scenario functional level different NPC (Non-Player Character) cars are defined to play in the scene. Each NPC needs at least a starting point and a trajectory to follow. This trajectory includes points, directions, and speed

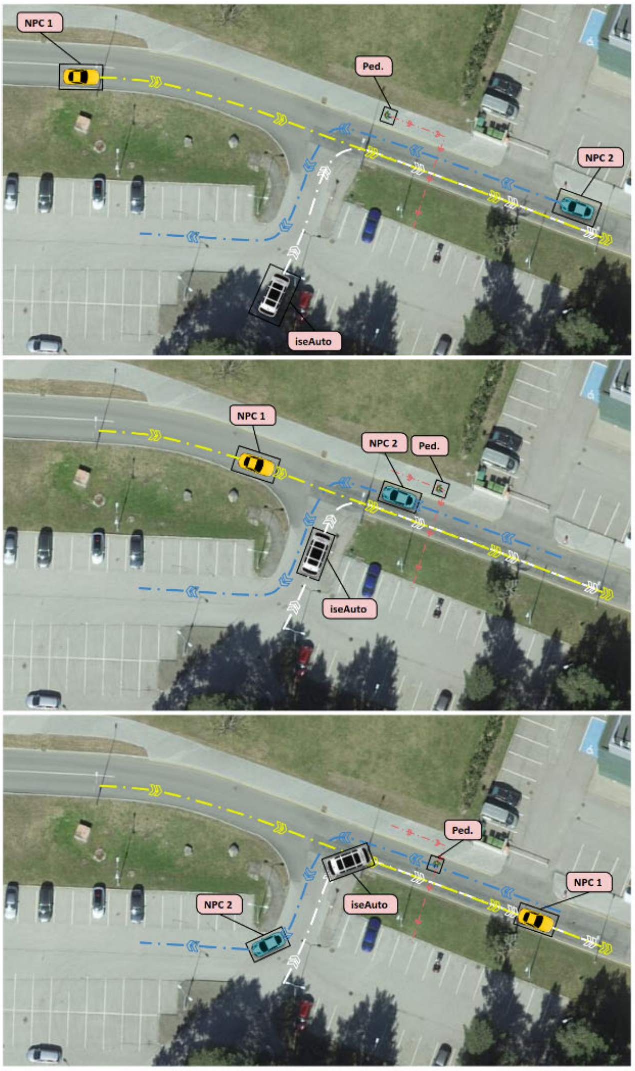
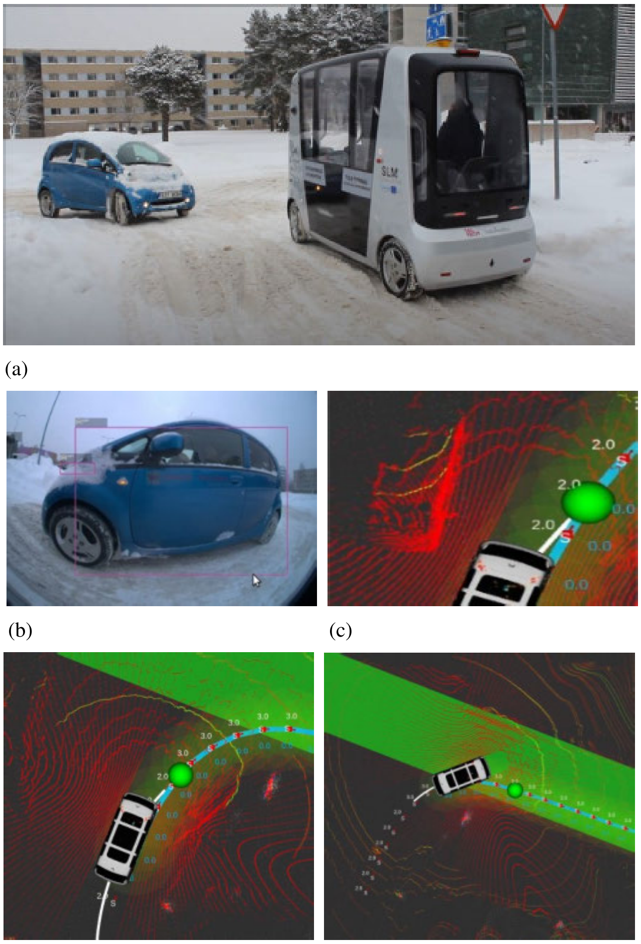


FIGURE 10. Three different frames of the scenario including the unit under the test and non-player characters, including vehicles and pedestrians.



(a) (b) (c) (d) (e)

FIGURE 11. **Validation scenario representation at the TalTech campus, in (a), and the result of each validation task in: (b) detection, (c) prediction, (d) localization, and (e) control.**

followed by the NPC. Furthermore, the ego (iseAuto), which is controlled by Autoware, should be spawned in a position in the environment, therefore, it requires an initial pose. Fig. 10 shows a few frames of a scenario in which the ego and NPCs are operating.

VII. PHYSICAL VALIDATION RESULTS

The physical validation was carried out in January 2022. The initial scenario was simplified due to the harsh weather conditions, which are a well-known challenge for AVs [44]. Roads were narrowed by snow and huge snow piles were blocking the visual view as well as LiDAR field of view. However, additional valuable data was gathered on how AV handles harsh weather and snowy roads.

From a qualitative point of view, the validation procedure was successful. Fig. 11 demonstrates components under the validation. In summary, the validation procedure results are the following:

Localization: The ego was able to localize itself in the predefined map.

Detection: The detection system successfully perceived and classified the obstacles.

Prediction: All the sensors perceived the obstacles with sufficient quality to perform classification and use the data in decision making with respect to the object behavior. The ego stopped before continuing the mission when the NPC was approaching.

Control: The ego was able to follow the predefined path with sufficient precision.

Decision: The ego was able to make the right decision and yield based on the classified data.

The vehicle successfully detected the approaching car and stopped to give it the right of way.[2] A rosbag was recorded during the physical validation. The rosbag contains all sensor data and internal messages used in autonomous driving like decision making and path planning. Overall, this process validated the use of PolyVerif such that more V&V methods can be developed in simulation.

VIII. CONCLUSION

This paper presents PolyVerif, the world's first open-source infrastructure focused on V&V researchers with the objective of accelerating the state-of-the-art for AV V&V research. PolyVerif provides an AI design and verification framework consisting of a digital twin creation process, an open-source AV engine, access to several open-source physics based simulators, and open-source symbolic test generation engines. PolyVerif's objective is to arm V&V researchers with a framework which extends the state-of-the-art on any one of the many major axes of interest and use the remainder of the infrastructure to quickly demonstrate the viability of their solution. Given its open-source nature, researchers can also contribute their innovations to the project. Using this critical property of open-source environments, the innovative potential of the whole research community to solve these vexing issues can be greatly accelerated. Finally, qualitative results from projects which have used PolyVerif are provided.

REFERENCES

[1] R. Razdan. (Mar. 2021). *Reflections on a Decade of AI*. Accessed: Oct. 1, 2023. [Online]. Available: https://www.forbes.com/sites/rahulrazdan/2022/03/02/reflections-on-a-decade-of-ai/

[2] M. Haenlein and A. Kaplan, "A brief history of artificial intelligence: On the past, present, and future of artificial intelligence," *California Manage. Rev.*, vol. 61, no. 4, pp. 5–14, Aug. 2019.

[3] J. Liu, X. Kong, F. Xia, X. Bai, L. Wang, Q. Qing, and A. Lee, "Artificial intelligence in the 21st century," *IEEE Access*, vol. 6, pp. 34403–34421, 2018.

[4] K. L. Fox, K. R. Niewoehner, and R. Razdan, "Novel framework to advance verification, validation, and certification of non-deterministic, AI-based algorithms for safety critical applications," *J. Air Traffic Control*, vol. 63, no. 3, pp. 14–29, 2021.

[5] *Software Considerations in Airborne Systems and Equipment Certification*, Standard DO-178C, 2011.

[6] W. Huang, K. Wang, Y. Lv, and F. Zhu, "Autonomous vehicles testing methods review," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 163–168.

[2] https://youtu.be/TJZQ9N-2ObQ

[7] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87456–87477, 2020.
[8] Y. Ma, C. Sun, J. Chen, D. Cao, and L. Xiong, "Verification and validation methods for decision-making and planning of automated vehicles: A review," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 480–498, Sep. 2022.
[9] *Waymo Safety Report*, Waymo, Mountain View, CA, USA, Feb. 2021.
[10] M. Tideman and M. van Noort, "A simulation tool suite for developing connected vehicle systems," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 713–718.
[11] J. Keirstead, N. Samsatli, and N. Shah, "SynCity: An integrated tool kit for urban energy systems modelling," in *Energy Efficient Cities: Assessment Tools and Benchmarking Practices*, 2010.
[12] M. Kehrer, J. Pitz, T. Rothermel, and H.-C. Reuss, "Framework for interactive testing and development of highly automated driving functions," in *Proc. Internationales Stuttgarter Symp.* Cham, Switzerland: Springer, 2018, pp. 659–669.
[13] H. Yu, S. Yang, W. Gu, and S. Zhang, "Baidu driving dataset and end-to-end reactive control model," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 341–346.
[14] R. Galbas, J. Reich, H. Schittenhelm, and N. Wagener, "Safeguarding methods for complex traffic scenarios for approval of automated driving functions," *ATZ Worldwide*, vol. 124, no. 9, pp. 56–61, Aug. 2022.
[15] Y. Lou, A. Ghiasi, M. Jannat, S. Racha, D. Hale, W. Goforth, and P. Bujanovic, "Cooperative automation research: CARMA proof-of-concept TSMO use case testing: CARMA cooperative perception concept of operations," Tech. Rep., 2022.
[16] Vires. (2022). *Vires Virtual Test Drive (VTD)*. [Online]. Available: https://vires.mscsoftware.com/
[17] Y. Kang, H. Yin, and C. Berger, "Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 2, pp. 171–185, Jun. 2019.
[18] K. W. Axhausen, A. Horni, and K. Nagel, *The Multi-Agent Transport Simulation MATSim*. London, U.K.: Ubiquity Press, 2016.
[19] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo—Simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation*, 2011, pp. 1–6.
[20] J. Wishart, S. Como, U. Forgione, J. Weast, L. Weston, A. Smart, and G. Nicols, "Literature review of verification and validation activities of automated driving systems," *SAE Int. J. Connected Automated Vehicles*, vol. 3, no. 4, pp. 267–323, Feb. 2021.
[21] Q. Goss, Y. AlRashidi, and M. I. Akbas, "Generation of modular and measurable validation scenarios for autonomous vehicles using accident data," in *Proc. IEEE Intell. Vehicles Symp.*, Jul. 2021, pp. 251–257.
[22] Q. Goss and M. I. Akbas, "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.
[23] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Toward verified artificial intelligence," *Commun. ACM*, vol. 65, no. 7, pp. 46–55, Jul. 2022.
[24] R. Razdan, "Unsettled technology areas in autonomous vehicle test and validation," SAE Tech. Paper, 2019.
[25] J. Ainsalu, V. Arffman, M. Bellone, M. Ellner, T. Haapamäki, N. Haavisto, E. Josefson, A. Ismailogullari, B. Lee, and O. Madland, "State of the art of automated buses," *Sustainability*, vol. 10, no. 9, p. 3118, 2018.
[26] T. Dueser and C. Gutenkuns, "A comprehensive approach for the validation of virtual testing toolchains," IAMTS, Chennai, India, Tech. Rep., IAMTS0001202104, Apr. 2021.
[27] 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.
[28] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, no. 3, pp. 1–48, 2012.
[29] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Mozeiko, 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.
[30] Tier IV Inc. (2022). [Online]. Available: https://tier4.jp/
[31] 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.
[32] D. J. Fremont, E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: A language for scenario specification and data generation," *Mach. Learn.*, vol. 2022, pp. 1–45, Feb. 2022.
[33] ASAM. (2022). *ASAM Openscenario V2.0*. [Online]. Available: https://www.asam.net/project-detail/asam-openscenario-v20-1/
[34] M. Malayjerdi, V. Kuts, R. Sell, T. Otto, and A. C. Baykara, "Virtual simulations environment development for autonomous vehicles interaction," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, vol. 84492, 2020, Art. no. V02BT02A009.
[35] H.-J. Przybilla, M. Lindstaedt, and T. Kersten, "Investigations into the quality of image-based point clouds from UAV imagery," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 13, pp. 539–545, Jun. 2019.
[36] Agisoft LLC. (2020). *Agisoft Metashape 1.6.5—Professional Edition: Agisoft LLC Software*. [Online]. Available: https://www.agisoft.com/downloads/installer/
[37] C. L. G. Quentin and M. I. Akbas, "SUMO traffic simulator interface for scenic," in *ERAU Discovery Day*, Mar. 2022.
[38] C. Medrano-Berumen, M. Malayjerdi, M. I. Akbas, R. Sell, and R. Razdan, "Development of a validation regime for an autonomous campus shuttle," in *Proc. SoutheastCon*, Mar. 2020, pp. 1–8.
[39] R. Sell, E. Malayjerdi, M. Malayjerdi, and B. C. Baykara, "Safety toolkit for automated vehicle shuttle -Practical implementation of digital twin," in *Proc. Int. Conf. Connected Vehicle Expo (ICCVE)*, Mar. 2022, pp. 1–6.
[40] 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.
[41] R. Razdan. (Jan. 2020). *Jacksonville's Journey in Engaging With Autonomous Vehicle Technology*. Accessed: Oct. 1, 2023. [Online]. Available: https://www.forbes.com/sites/rahulrazdan/2020/01/25/jacksonvilles-journey-in-engaging-with-autonomous-vehicle-technology/
[42] Acclivis. (2022). *Acclivis Technologies*. [Online]. Available: https://acclivistech.com/
[43] C. Stark, C. Medrano-Berumen, and M. I. Akbas, "Generation of autonomous vehicle validation scenarios using crash data," in *Proc. SoutheastCon*, Mar. 2020, pp. 1–6.
[44] M. Bellone, A. Ismailogullari, J. Müür, O. Nissin, R. Sell, and R.-M. Soe, "Autonomous driving in the real-world: The weather challenge in the Sohjoa Baltic project," in *Towards Connected and Autonomous Vehicle Highways*. Cham, Switzerland: Springer, 2021, pp. 229–255.

RAHUL RAZDAN (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Carnegie Mellon University, in 1984 and 1985, respectively, and the Ph.D. degree in electrical engineering from Harvard University, in 1994. He maintains academic appointments with the University of Florida, Florida Polytechnic University, and the Institute for Human and Machine Cognition. He is currently the CEO of Raztech Research Institute, a technology-based research institute, which incubates ideas in the intersection of technology and society, such as autonomous vehicles, STEM education, and long-lifecycle products. In corporate roles, he was the Senior Vice President of Strategy with Flextronics (FLEX), a General Manager of the Systems and Functional Verification Business with Cadence Design Systems, and an Alpha CPU Architect with Digital Equipment Corporation. He has successfully built startups in wireless power, machine learning, and low-power electronics design.

MUSTAFA İLHAN AKBAŞ (Member, IEEE) received the Ph.D. degree in computer engineering from the University of Central Florida. He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University. He has industry experience in projects with multinational defense industry partners and large enterprises. His research interests include 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 is a member of ACM, AAAI, and SAE. He serves on the editorial boards and program committees for ACM and IEEE journals and conferences.

RAIVO SELL (Member, IEEE) received the Ph.D. degree in product development from the Tallinn University of Technology (TalTech), in 2007. He has been a Visiting Researcher with ETH Zürich, Aalto University, and most recently with Florida Polytechnic University, USA. He is currently a Professor of robotics with TalTech. He is running 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. His research interests include mobile robotics and self-driving vehicles, smart city, and early design issues of mechatronic system design. He was awarded as a Chartered 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. From 2015 to 2020, he worked 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. His research interests include 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.

MAHESH MENASE received the master's degree in computer science from Pune University. He has diverse and rich work experience of over 15 years in various industries across multiple disciplines. He has experience in managing multiple teams focusing on building embedded systems and applications for semiconductors, telecom, the IoT, and networking products. He has held senior leadership roles in both technology and management and has served global markets, including the USA, Europe, Asia–Pacific, and India. He is currently building a PolyVerif AV validation framework for the validation of the autonomous vehicle software stack. His research interests include robotics, simulation, and autonomous vehicles.

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 include the intersection of autonomous vehicle developments and safety verifications. His current research is a simulation approach for the safety verification and development of a real AV shuttle in a virtual environment.