IEEE SA
STANDARDS ASSOCIATION

IEEE SA WHITE PAPER

AUTONOMOUS DRIVING WORKING GROUP

# SIMULATION, TESTING, VERIFICATION, AND VALIDATION (STV$^2$) OF AUTONOMOUS DRIVING

*Authored by*

Autonomous Driving Working Group

IEEE

## ACKNOWLEDGMENTS

Special thanks are given to the members of the IEEE SA Autonomous Driving Working Group (ADWG) for their discussions, comments, and reviews.

Boon Chong Ang, *Intel*
Wen Cui, *Synkrotron*
Kunkun Hao, *Synkrotron*
Zhihao Jiang, *Shanghai Tech University*
Simran Khokha, *Infineon Technologies Germany*
Astha Kukreja, *IEEE Senior Member*
Subhadip Kumar, *IEEE Senior Member*
Hongyang Li, *Shanghai AI Lab*
Xiao Liang, *China General Certification Center*
Linda Lim, *The University of California, Berkeley*
Jiaqi Ma, *The University of California, Los Angeles*
Zeyu Ma, *Shanghai Development Center of Computer Software Technology*
Zvikomborero Murahwi, *Gratia ICT Projects Advisory*
Yi Lu Murphey, *The University of Michigan*
Yuxi Pan, *Synkrotron*
Gaurav Pandey, *Texas A&M University*
Scott Schnelle, *Waymo*
Jin Shang, *IEEE SA ADWG Vice Chair*
Dong Sun, *IEEE SA ADWG Chair*
Lei Sun, *TuSimple*
Huijie Wang, *Shanghai AI Lab*
Zijiang Yang, *Synkrotron & IEEE SA ADWG Vice Chair*
Mohammad Yasin, *Tech Mahindra*

*The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA*



*PDF: STDVA27368 979-8-8557-1305-3*

## NOTICE AND DISCLAIMER OF LIABILITY CONCERNING THE USE OF IEEE SA DOCUMENTS

This IEEE Standards Association ("IEEE SA") publication ("Work") is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the activity that produced this Work. IEEE and the Autonomous Driving Working Group (ADWG) expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the ADWG disclaim any and all conditions relating to: results; and workmanlike effort. This document is supplied "AS IS" and "WITH ALL FAULTS."

Although the ADWG members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE SA OR ADWG MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patents claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder.

This Work is published with the understanding that IEEE and the ADWG members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

# TABLE OF CONTENTS

# SIMULATION, TESTING, VERIFICATION, AND VALIDATION (STV$^2$) OF AUTONOMOUS DRIVING


## ABSTRACT

STV$^2$ is a set of processes that support the development, validation, and operation of autonomous driving systems from the perspectives of *safety* and *cost*. The scope, architecture, and critical components of STV$^2$, as well as how the full lifecycle of autonomous driving systems is covered, is presented in this white paper. STV$^2$ is part of the infrastructure and tooling layer of autonomous driving architecture.

STV$^2$'s business objective and process are fulfilled by a simulation system supported by data infrastructure as described in this white paper.

# 1. INTRODUCTION

## 1.1. GENERAL

As the world advances into a new era of mobility, the evolution of autonomous vehicles (AVs) has been remarkable, from the early experiments with remotely controlled vehicles in 1921 to the innovative steps taken more recently by emerging companies [1],[2].[1] By 2030, there may be progress toward the commercialization of Level 4 autonomous vehicles, as defined by the Society of Automotive Engineers, which are designed to perform driving tasks independently under specific conditions. However, it is important to note that achieving full Level 4 capabilities might remain challenging and uncertain [3]. The development of AVs presents increasingly intricate challenges at each stage. Simulation, testing, verification, and validation ($STV^2$) are critical to addressing these challenges. This white paper aims to provide an in-depth exploration of $STV^2$, emphasizing how simulation and robust data infrastructure can significantly enhance the development process while upholding stringent safety and performance standards.

## 1.2. CHALLENGES OF AUTONOMOUS DRIVING

### 1.2.1. GENERAL

The key challenges of autonomous driving are integral to its supporting infrastructure, operational environment, and the management of its operations. While the management of operations is primarily from within the autonomous vehicle at any point in time, infrastructure and environmental issues can be internal or external to the AV. Therefore, a deep understanding of the challenges would require individual identification and analyses of issues or items of concern in infrastructure, operational environment, and operations management, followed by an integrated analysis of the same.

### 1.2.2. SAFETY AND SECURITY

Safety and security in autonomous driving can be viewed from two scenarios. The first scenario is where the autonomous vehicle is a security safety threat to its environment and surroundings, and the second scenario is where there are safety and security threats to the autonomous vehicle. The following is a detailed analysis and discussion of these two scenarios:

[1] Numbers in brackets correspond to the references in Section 5

a) *The autonomous vehicle causing safety and security challenges to its environment*—An autonomous vehicle faces significant security challenges when its internal or local system is compromised, leading to unauthorized changes in its settings and configurations. These alterations can cause malfunctions, such as loss of direction, which may suggest attempts at hijacking. Such vulnerabilities are critical concerns for the safety of the vehicle and the content within, including passengers. To effectively address these challenges, robust cybersecurity measures are crucial. Compliance with standards like ISO/SAE 21434:2021, Road vehicles—Cybersecurity engineering plays a key role in protecting the vehicle's systems. However, despite these precautions, unauthorized adjustments can still occur. These adjustments can severely degrade the vehicle's ability to recognize external elements such as other vehicles and road signs, increasing the risk of accidents, physical harm, and even loss of life.

b) *The operational environment causing safety and security challenges to the autonomous vehicle*—The autonomous vehicle's operating environment and structures that support the operations of the autonomous vehicle can also be compromised and subjected to security attacks and/or distortions/corruption in control data and transmission signals, which can result in errors and even loss of communication of the environment with the autonomous vehicle. Such loss of communication and corruption of routing data, for example, through interceptions, can also cause malfunctioning of the autonomous vehicle resulting in loss of direction.

### 1.2.3. REGULATIONS

Regulations here encompass three aspects, i.e., compliance, certification, and supervision, which are critical for ensuring that AVs meet established safety and operational standards. Specifically, *compliance* involves adhering to legal and regulatory requirements set by governmental and industry bodies [4],[5],[6],[7]. *Certification* is the process by which AVs are officially recognized as meeting these standards [8],[9],[10]. *Supervision* refers to the ongoing oversight of AV operations to ensure continuous adherence to these regulations [11]. Together, these components form the backbone of a regulatory framework essential for safely and reliably integrating AVs into public roadways.

The interplay between developers and regulators is pivotal in the rapidly evolving realm of AV. On the one hand, it is crucial to recognize the significant responsibility placed on AV developers. As AVs increasingly take on roles typically filled by human drivers, their design complexity escalates to address both anticipated and unforeseen safety concerns. Conversely, regulators respond to these challenges by demanding greater transparency in AV operations, a common hurdle in AI systems. While these regulations add complexity to AV development, their importance cannot be overstated. They strike a necessary balance, preventing premature development of underdeveloped technology and avoiding excessively restrictive rules that might hinder the advancement of this innovative field. Through this balanced approach, developers and regulators collaboratively promote responsible development but also help maintain public trust and safety in an industry that is experiencing swift and significant changes.

### 1.2.4. BENEFITS

Full adoption of AVs is projected to yield significant cost benefits, estimated at around $4,000 per vehicle annually [12], or approximately $0.44 per trip-mile for shared AVs [13]. For individual users, accessing these technologies could be economically feasible. For instance, one developer’s self-driving package is available for a monthly subscription of around $199 [14]. The benefits of AVs encompass various aspects of daily life, including enhanced safety, reduced traffic congestion, and savings on parking space. Over time, these advantages are expected to offset the substantial costs associated with research and development (R&D), testing and certification, data management, and infrastructure updates.

Despite the billions already invested in this sector and progress being gradual yet promising [15],[16], the long-term economic outlook remains favorable. It is worth noting that the overall sustainability of the AV business model hinges crucially on how companies operate these vehicles [13]. This requires continuous and thorough research to adapt to the dynamic nature of this field.

## 1.3. THE ROLE OF STV$^2$

### 1.3.1. RESEARCH AND DEVELOPMENT

In the R&D of autonomous vehicles, STV$^2$ is pivotal for transforming theoretical concepts into practical and reliable AV systems. During the R&D phase, the essential first steps are simulation [17],[18],[19] and testing [20],[21]. They enable developers to model and refine AV systems within both virtual environments and controlled real-world scenarios, identifying potential issues, and optimizing system performance early in the development cycle. Verification and validation, explored in detail in subsequent sections, are the later stages of the R&D process [22]. Verification ensures that each aspect of the AV system meets specific design and technical criteria, a necessary step for confirming the system’s adherence to initial specifications. Validation, in contrast, ensures whether the system fulfills its intended operational purpose and meets end-user requirements, crucial for ensuring the vehicle’s practical usability and safety in real-world conditions.

Thus, $STV^2$ leads the progression from initial design to a fully functional AV prototype. Through these comprehensive steps, $STV^2$ not only enhances the safety and reliability of the final product but also contributes to the efficient allocation of resources and cost-effectiveness during the development, making the AV systems both technologically sound and aligned with market and user expectations.

### 1.3.2. VERIFICATION AND VALIDATION

In the development of autonomous systems, both verification and validation are crucial for ensuring system integrity and operational efficacy, each addressing distinct aspects. Verification focuses on confirming that the system adheres to specific predefined requirements and that all elements are implemented correctly according to design specifications. This includes the process of requirement traceability, which tracks each requirement throughout the development lifecycle to ensure it is implemented and met by the final product. Verification also involves static analysis, where the code is analyzed without execution to identify potential vulnerabilities or deviations from the specified standards. Code reviews and inspections are thorough examinations of code and design documents to verify that all specifications are met and that the design can effectively handle both expected and unexpected scenarios. Integration testing is another facet of verification, examining the interfaces between components to ensure they work together as intended.

Validation contrasts by assessing whether the system meets the needs and expectations of the end-users and operates effectively in the intended environment. Although verification establishes the foundational accuracy of system design against specifications, our following discussion will emphasize validation to demonstrate how systems perform under real-world conditions, ensuring they meet both functional and user-centric requirements. Validation can be broadly categorized into white box and black box testing, each with distinct methodologies and transparency levels [22]. White box testing is characterized by its openness and visibility into the system's internal workings. It typically involves the use of sophisticated simulation environments that incorporate Model-in-the-Loop (MIL), Hardware-in-the-Loop (HIL), and Software-in-the-Loop (SIL). These simulations are critical for testing the integration and interaction of software with physical hardware components in a controlled environment. White box testing also extends to function tests, where specific functions are validated against expected outcomes and fault injection techniques, which deliberately introduce faults to test system responses. Furthermore, it includes the application of failure analysis methods such as Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA). These analyses are valuable in identifying potential failure points and evaluating safety implications, especially by simulating negative requirements and scenarios that involve misuse, abuse, or confusion by users.

Black box testing, on the other hand, does not reveal the internal mechanisms of the system to the tester, focusing instead on the outputs generated in response to certain inputs and conditions. This method employs simulation environments where models or entire systems are looped through various test cases to observe behavior without regard to the internal processing paths. Black box testing also involves empirical strategies where real-world conditions are mimicked as closely as possible, often utilizing a brute force approach to run through numerous realistic scenarios to test robustness and reliability. This approach is complemented by specific quality requirements testing, such as penetration testing, which assesses security vulnerabilities, and usability testing, which evaluates the user interface and user experience aspects.

Both white box and black box testing are increasingly utilizing automated techniques to enhance the efficiency and effectiveness of the testing process. Automated validation employs artificial intelligence and cognitive testing methods to simulate human interaction with the system, aiming to uncover issues that might not be apparent through conventional testing methods. This automation is crucial in handling the complex, dynamic scenarios typical of autonomous system environments, enabling more frequent and thorough testing cycles that are essential for ensuring the safety and functionality of these systems. The choice between white box and black box testing methods, along with the degree of automation implemented, will depend on specific requirements, the nature of the system under test, and the criticality of the functions it performs. Each testing approach offers unique advantages and is often used in conjunction to provide a comprehensive validation framework for autonomous systems.

TABLE 1 summarizes the various methods used to verify the integrity of autonomous systems, both statically and dynamically. All the system's functional requirements should be met, and positive testing techniques are designed to ensure this. In contrast, non-functional requirements can be checked off thanks to the techniques used in negative testing. System requirement specifications (SRS) often do not list negative requirements, which include things like safety and cybersecurity.

## TABLE 1 AUTONOMOUS SYSTEM STATIC AND DYNAMIC VALIDATION TECHNOLOGIES [22]

| Method | Characteristics | Tool support, technologies | Coverage | Regression strategy | Strength | Weakness | Effectiveness | Efficiency |
|---|---|---|---|---|---|---|---|---|
| Modeling and simulation environments with SIL, HIL, MIL | Static and dynamic | Model checker, e.g., Matlab, dSPACE, Vector VT System, NovaCarts, Vires, PreScan | 0 | Repeat impacted scenarios (low efficiency) | Reduces validation cost<br>Decouples hardware and software development | Brute force for high coverage<br>Requires large computation power<br>Tests only for known scenarios<br>Scenario banks are not comprehensive to validate autonomous systems | 0 | 0 |
| Function test | Dynamic, all functions | Modeling tool for functional abstraction with unit test tools (e.g., JUnit, PHPUnit), dedicated test environments for stub generation | 0 | Repeat functional test cases for impacted functions | Tests all AI aspects: sensing, decision-making, and action taken<br>Validates all functional requirements | Insufficient to validate complete system | + | + |
| Integration test | Dynamic | Test suites, test management, combinatorial tools such as AETG, Citrus, etc. | 0 | Regenerate test cases | Tests integration of components | Large number of interfaces; easy to miss some links<br>Fault localization is difficult | + | + |
| Fault injection | Static for residual defect estimation | Test environment and defect modeling, e.g., beSTORM, Security Innovation | – | Introduce few selected defects | Provides estimate on residual defects and coverage<br>Exposes weaknesses, enabling designers to strengthen them | Need concrete understanding of underlying system architecture and behavior | + | – |
| Negative requirements with misuse, abuse, confused cases | Static, specifically for safety, security, usability | Directly modeled and traced with requirement tools, e.g., DOORS, Visure, PTC, PREvision, Enterprise Architect, HP ALM | 0 | Reuse situational negative cases | Good for scenarios to be avoided<br>Formalizes non-functional requirements<br>Strengthens system security | Difficult to set up systematically<br>No coverage schemes<br>The test cases do not necessarily cover all possible negative cases | + | + |
| FMEA, FTA | Static, specifically for safety-critical systems | FMEA worksheets, component abstractions, reuse library | 0 | Retest for the changed components | Well established for safety and security (attack tree)<br>Enables designers to foresee system interface failures | Depends heavily on human knowledge<br>Labor intensive | + | + |
| Experiments, empirical test strategies | Empirical test generation for load test, performance, thermal, etc. | Experiment-specific test tools, such as Parasoft DTP, EggPlant, Thermal imager, etc. | + | Repeat the test strategies for changed functions | Relatively easier to frame the test cases<br>Covers wide range of electrical systems | Depends heavily on human knowledge<br>Labor intensive<br>Very little or no test automation | + | 0 |
| Specific quality requirements test, e.g., pen testing, fuzzing | Dynamic, specifically for quality requirements | Dedicated test tools, e.g., automatic fuzzing extensions, e.g., CANoe, OWASP ZAP, Vega, etc. | – | Retest for impacted components | Well established for security<br>Effective in ensuring that the system meets known quality requirements | Often insufficient to validation complete system security and safety | – | + |
| Brute force usage in real world while running realistic scenarios | Dynamic for ensuring situational coverage | Recording and replay with actual scenario libraries with data loggers from various sensor systems, e.g., Tecnomatix, CarMaker, EB Assist, CANape | 0 | Repetition (low efficiency) | Closest to real world and thus highly effective<br>Validates all systems at once<br>Comprehensive view and coverage<br>Standardizes scenario storage format and tagging | High effort for coverage<br>Unclear coverage<br>Most of the test cases are redundant<br>Untransparent situational coverage | + | – |
| Intelligent validation, e.g., cognitive testing | Dynamic test generation and selection depending on situation and environment | Machine-learning frameworks, such as Tensorflow, Apache Spark, and so on<br>Open data sets, such as nuScenes | | Reuse generated test cases from dependency database | Improved transparency<br>Automatically considers dependencies to external environment and internal functions<br>Automates major part of test procedure<br>Standardizes scenario storage format and tagging<br>Sharing test scenarios across V-model abstraction levels | High effort to set up AI-based test environment<br>Needs large computation power<br>Growing discipline, i.e., not many methods and tools available | + | + |

The World Forum for Harmonization of Vehicle Regulations, more generally known as Working Party 29 (WP.29), bears the responsibility of providing regulations for road vehicles. They give special consideration to facilitate the introduction of innovations and technologies from a legal standpoint into the vehicles. WP.29 operates as a working group of the sustainable mobility division of the United Nations Economic Commission for Europe (UNECE). WP.29 now consists of seven working groups that focus on formulating new technical regulations within their subject expertise.

In June 2018, the working party on automated/autonomous and connected vehicles (GRVA) was formed to evaluate the significance of WP.29's actions related to automated/autonomous and connected vehicles. The GRVA team is also concerned with the security of automated vehicles. The Functional Requirements for Automated and Autonomous Vehicles (FRAV) working group is a loosely organized organization that has been working on the ideas of operational design domain (ODD), object and event detection and response (OEDR), and human-machine interaction (HMI) to better understand the functional requirements of AVs from a safety standpoint. This committee oversees outlining the criteria for AV approval, including safety standards. Validation method for automated driving (VMAD) is one of the GRVA's informal working groups, along with FRAV. The mission of this team is to create innovative approaches to testing and evaluating AVs that can be employed in the certification procedure. The combination of FRAV and VMAD would lead to comprehensive regulatory frameworks for automated driving systems (ADS) [23].

Perception algorithms for ADAS primarily rely on deep learning techniques. Two main approaches to validating ADAS systems are scenario-based and algorithm-based.

*Scenario-based methods* involve creating and collecting a wide range of complex scenarios to simulate potential situations that may arise during real-world driving. The objective is to provoke system failures in the advanced driver assistance systems (ADAS), thereby identifying their weaknesses during the validation phase. Typically implemented on simulation platforms, these scenarios often draw from real-world vehicular accidents and artificially constructed complex situations. ADAS can be validated through software in the loop (SIL) or hardware in the loop (HIL). A cover article in *Nature* on March 23, 2023, titled "Dense Reinforcement Learning for Safety Validation of Autonomous Vehicles," [24] introduced a method of adaptively adjusting scenarios through dense reinforcement learning. This approach can construct adversarial scenarios, expediting the discovery of conditions under which ADAS are prone to failure.

*Scenario-based testing,* generally applied to the entire ADAS, has the advantage of closely mirroring real-world conditions and providing easily interpretable failure modes for improvement. This approach is widely adopted by leading manufacturers and testing agencies. However, its limitation lies in the granularity of the test elements, which are often components of the scenario description language. For example, in a snowy or rainy scenario, this method cannot pinpoint which specific snowflake or raindrop causes the system to fail, meaning that it only covers a small part of the real parameter space.

Alternatively, *algorithm-based testing* focuses on specific modules of the autonomous driving system, such as lane line detection or traffic sign recognition. Given the natural susceptibility of deep learning algorithms to adversarial attacks, these methods identify weaknesses in perception algorithms using such attacks. Various approaches exist, but the essence is to identify pixel combinations within a scenario that are most likely to cause algorithm failure. Typically, these attacks are conducted in simulated environments, but there are also instances of physical world attacks. A notable example is the Best Technical Poster Award at NDSS 2020, where researchers used a carpet designed to mimic road dirt to fool the open-source autonomous driving system, OpenPilot.

The strength of algorithm-based methods lies in their ability to uncover the hidden capacity limits of ADAS and identify deeper vulnerabilities through algorithmic analysis, theoretically exposing a broader range of potential risks. However, these methods also have drawbacks. The attack techniques vary for different perception algorithms, and designing them is challenging, especially since algorithms in real scenarios often operate as black boxes. Moreover, the attack patterns generated by these methods can differ significantly from real-world scenarios, raising the issue of how to attribute successful attacks to interpretable patterns that can enhance the robustness of algorithms. This remains an unresolved challenge.

### 1.3.3. OPERATIONS AND ASSURANCE

STV$^2$'s role in operations and assurance is vital for securing the long-term success and trustworthiness of autonomous driving technologies. Beyond initial testing, this involves a continuous focus on post-deployment processes, which is crucial for maintaining the operational safety and effectiveness of AVs. Adapting to the dynamic, ever-changing conditions of the real world is key. This adaptation includes but is not limited to the following: 1) continuous anomaly detection and performance monitoring using AV sensory data [26],[27]; 2) regular testing of AVs against a wide range of scenarios, particularly rare and unexpected ones [28],[29]; 3) keeping systems updated with the latest safety features and technological advancements [30]; 4) continuously tracking and adhering to evolving regulatory standards [31],[32]; 5) incorporating diverse feedback to enhance user experience and safety [33],[34]. These examples, among other strategies, are pivotal in upholding rigorous safety and reliability standards in AV technology.

## 1.4. MARKET LANDSCAPE

The autonomous driving market is currently witnessing an optimistic growth phase, as projected by market research institutions. Valued at approximately USD 100 billion in 2023, the sector is expected to significantly advance with a compound annual growth rate (CAGR) of 20% to 40% over the next decade. By 2035, it is estimated that the revenue generated specifically by the autonomous driving sector could be between USD 300 to 400 billion. This is a substantial portion of the broader autonomous vehicle market, which itself is anticipated to achieve a total market size of around USD 4 trillion by 2035 [35],[36],[37],[38]. In terms of regional market share, North America, particularly driven by Silicon Valley's technological innovations and robust regulatory policies, accounts for about 40% of the global market. The Asia-Pacific region, especially China, Japan, and South Korea, is forecasted to witness the fastest growth [35],[36], powered by their commitment to traffic and energy efficiency goals—key factors aligning with autonomous vehicle development.

The market landscape is characterized by stiff competition among startups and established OEMs vying for dominance. Specifically, Tesla, a leading company in autonomous driving, aims to sell 20 million vehicles annually by 2030 [39], targeting around 20% of the global vehicle market.[2] Moreover, a McKinsey report suggests that by 2035, in an accelerated adoption scenario, approximately 57% of all vehicles could be equipped with advanced autonomous driving technologies [38], underscoring the rapid technological integration within the automotive sector.

The flourishing market for autonomous driving is significantly impacting the landscape of STV[2]. As autonomous vehicles become increasingly advanced and widespread, there is a corresponding escalation in demand for rigorous and comprehensive STV[2] processes. It is estimated that design validation and related activities will constitute a major, if not the largest, cost component in the development of autonomous driving systems [40],[41]. Concurrently, the market for simulations is expected to grow at a CAGR of 13.4% over the next decade, reaching approximately USD 3 billion [42]. The transition from traditional hardware-defined vehicles to software-defined ones is recognized as a critical step toward fully adopting autonomous driving. This shift, however, introduces significant challenges for manufacturers and their suppliers in terms of STV[2], necessitating

[2] This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

a revolutionary approach to industry frameworks. Nevertheless, this complexity also presents a unique advantage for technological leaders who can outpace competitors in this domain [43]. These trends underscore the vital need for robust and comprehensive STV² implementation to ensure that autonomous vehicles are not only constructed efficiently but also operate with utmost safety and reliability.

# 2. OVERVIEW OF STV²

## 2.1. THE LIFECYCLE OF STV²

Traditionally, STV² processes in autonomous driving have been rooted in model-based methods with manual parameter tuning, focusing on modular procedures. The emergence of AI, however, is revolutionizing this paradigm. Challenging tasks are increasingly addressed through data-driven, end-to-end processes, leveraging advanced neural networks and the utilization of high-quality, diverse datasets. As a result, the entire STV² lifecycle is becoming more data-centric, intertwining requirements with various model enhancements, underpinned by methods like simulations, on-road testing, and feedback from operations. The lifecycle is illustrated in FIGURE 1.

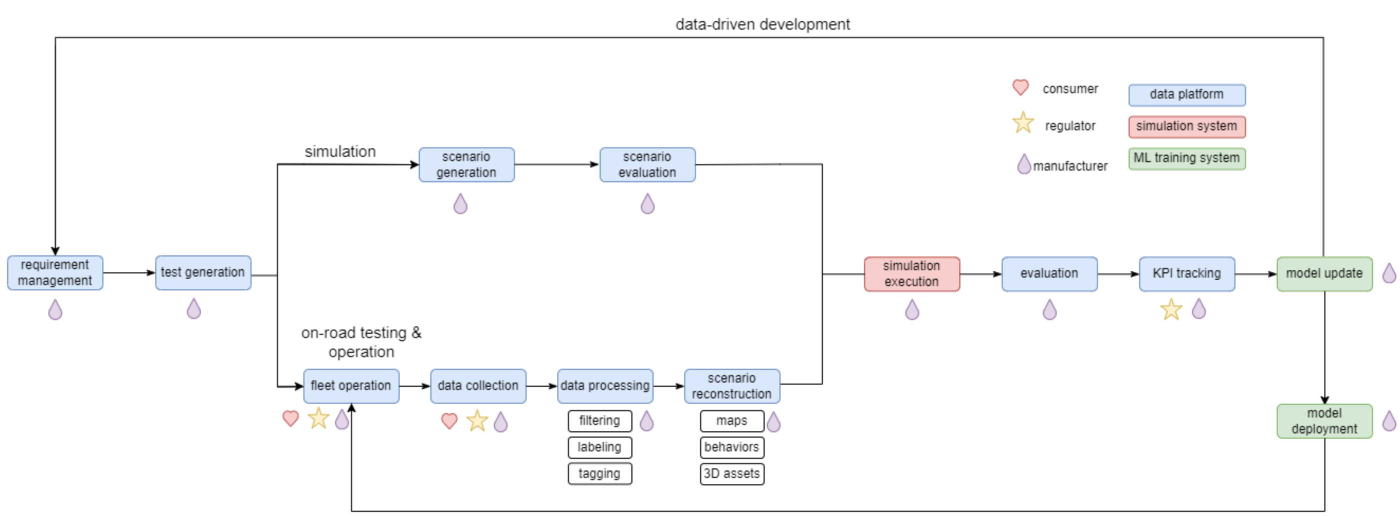


**FIGURE 1 LIFECYCLE OF STV² AND STAKEHOLDERS (KPI: KEY PERFORMANCE INDICATOR)**

## 2.2. KEY ELEMENTS AND GAP ANALYSES

### 2.2.1. SCENARIOS

A *scenario* includes detailed elements within a specific driving situation or environment. These elements, ranging from road configurations to weather conditions and the behavior of other road users, are critical for the tests or simulations for AVs. Scenarios can be broadly categorized into real or virtual formats, each serving unique purposes in the development and testing of AVs.

Real scenarios are as follows:

- **Direct observation:** Derived from actual driving conditions, captured through vehicle-mounted sensors, documenting real traffic, pedestrian behavior, and environmental conditions.
- **Proving ground:** Custom-built physical settings that replicate real-world conditions, allowing for safe and controlled testing of scenarios that are hard to encounter in the real world.

Virtual scenarios are as follows:

- **Pure computer-generated simulations:** These scenarios are purely digital and are constructed to maintain a no-risk environment to test algorithms across diverse conditions.
- **Logsim:** Utilizes recorded real-world driving data and replays in a simulated environment to analyze interesting past events.
- **Open-loop data replay:** Involves testing with predetermined inputs to observe the AV's response without its actions influencing the scenario.
- **Worldsim:** A dynamic simulation where the virtual environment reacts to the AV's actions, offering a more realistic and interactive testing experience.
- **Log2world:** Merges real-world data with interactive features, allowing for "what-if" scenarios and counterfactual analysis.

The development of scenarios for autonomous vehicle testing and simulation is a complex process that faces multiple challenges. Some major challenges, listed next, need to be effectively addressed to ensure the scenarios are realistic, comprehensive, and ready for advancing AV technology.

a) Realism gap: Advanced simulations may still fall short of fully encapsulating the unpredictability and complexity of real-world driving.

b) Data requirements: The creation of realistic scenarios demands extensive and varied data, posing challenges in acquisition and management.
c) Validation and standardization: The absence of universal standards for scenario validation complicates the consistent assessment of AV system readiness and safety.
d) Scalability issues: Developing, managing, and testing an exhaustive range of scenarios to cover all driving conditions is a formidable task.
e) AI integration: Seamlessly integrating scenario testing with the ongoing development and refinement of AI algorithms is an ongoing challenge.

### 2.2.2. SIMULATION MODELS

The current development and application of simulation models reveal several key findings. These include advancements in sensor models, vehicle dynamics, and agent behavior models, as well as performance benchmarks like realism and latency, and associated bottlenecks.

- **Sensor models:** Sensor simulation is critical for validating the perception systems of autonomous vehicles. Recent developments have focused on creating more realistic and varied simulation environments to test sensor performance under different conditions.

  **Examples:** Tools like NVIDIA's DRIVE Sim and Unity's High-Definition Render Pipeline (HDRP) simulate light detection and ranging (Lidar) sensors, allowing for realistic rendering of Lidar data in various weather and lighting conditions. CARLA, an open-source simulator for autonomous driving, offers camera simulation capabilities that include different types of cameras (RGB, depth, semantic segmentation) to mimic real-world scenarios. An enhanced version of this simulator, Synkrotron Oasis, offers more accurate camera and Lidar sensor models, along with user-friendly tools to configure all the parameters depending on various sensor types.

- **Vehicle dynamics:** Advanced models simulate vehicle behavior and interactions with various road conditions. The goal is to create realistic representations of how vehicles move and respond to control inputs.

  **Examples:** Simulators like IPG CarMaker and VI-CarRealTime provide high-fidelity vehicle dynamics models suitable for real-time applications, including various types of vehicles and driving conditions. rFpro offers detailed models for vehicle handling and control, allowing engineers to test vehicle responses to steering, braking, and acceleration under different road conditions.

- **Agent behavior models:** Developing sophisticated models for pedestrian, cyclist, and other vehicle behavior is essential. This includes modeling unpredictable human behavior to ensure the autonomous system can handle real-world scenarios.

  **Examples:** SUMO (Simulation of Urban MObility) is an example of a tool for simulating traffic flow, including individual vehicle behavior and interactions at a large scale. The PedSim library simulates pedestrian dynamics and behavior, which is crucial for testing autonomous vehicles in urban environments.

- **Map models:** High-definition maps provide accurate information on factors such as road geometry, traffic sign placement, and traffic topology. Their primary task is to ensure that map information can be transferred to and be recognized by AVs so that these vehicles can plan routes like humans.

  **Examples:** Many high-definition map formats are available, with the main ones being Autoware Vector Map, OpenDRIVE, Lanelet2, and Navigation Data Standard (NDS). OpenDRIVE is mainly used for simulation-based applications and is available in an open-source format.

### 2.2.3. BENCHMARKS AND CHALLENGES

The following are some of the challenges facing STV[2]:

- **Realism:** The fidelity of simulations is a crucial benchmark. High realism in simulating real-world conditions is essential for effective training and validation of autonomous systems.
- **Latency:** Low latency is critical for real-time decision-making in autonomous driving simulations. However, complex models, especially those incorporating large language models (LLMs), often suffer from high latency, impacting the timeliness of decision-making in simulations.
- **Computational efficiency:** The computational demands of high-fidelity simulations are significant. This includes the need for advanced hardware and optimization techniques to run simulations efficiently.

### 2.2.4. FURTHER DEVELOPMENT AND BEYOND

The integration of various simulation components into a cohesive and realistic system remains a significant challenge. Specifically, the scalability of simulations to encompass a broad spectrum of real-world scenarios and their ability to generalize to new, unseen situations presents ongoing difficulties. Another bottleneck is the availability and quality of data; accessing diverse, high-quality datasets is essential for creating realistic simulation scenarios but remains limited. Accurately modeling human behavior in simulations is particularly challenging due to the unpredictability and diversity of human actions in driving scenarios. Looking to the future, efforts are needed to enhance the realism of simulations, especially in complex scenarios. Finally, there is a push to develop more efficient simulation algorithms and hardware optimizations to reduce latency and computational demands.

### 2.2.5. EVALUATIONS

Verification and validation (V&V) processes typically focus on ensuring that the system correctly implements specified functions (verification) and that it meets the initial design requirements and regulations (validation). Beyond V&V, evaluating AD systems involves assessing how these systems perform in real-world or realistic virtual environments from the perspectives of users, regulators, and other stakeholders, by considering a broader range of factors, such as safety, usability, efficiency, and regulatory compliance. The evaluation metrics and scenarios used can vary widely based on specific goals, but generally, they aim to ensure that AD systems can operate safely and effectively in the environments and situations they will encounter.

From different perspectives, users prioritize usability, comfort, and trust, using metrics like satisfaction and safety. Regulators focus on compliance with safety standards, assessing performance through indicators like disengagement rates. Stakeholders such as insurers and manufacturers consider system reliability, its effect on traffic, and environmental impacts. Specifically, the available metrics are as follows:

- **Safety metrics:** Number of disengagements, incident/accident rates per miles driven, and reaction time to unforeseen events
- **Performance metrics:** System efficiency (e.g., fuel consumption, traffic throughput), accuracy in following routes, and handling complex driving scenarios
- **Perception metrics:** Object detection recall and precision, tracking accuracy, localization accuracy
- **User experience metrics:** Comfort measures (such as smoothness of acceleration and braking), system responsiveness, and ease of control or interaction
- **Reliability metrics:** Mean time between failures (MTBF), software stability, and sensor accuracy under various conditions

Different evaluation scenarios are used to test AD systems under controlled but realistic conditions, either virtually in simulations or in real-world environments designed to mimic typical or extreme driving situations.

- **Urban driving:** Involves complex interactions with pedestrians, cyclists, and other vehicles. Focuses on navigating urban streets, intersections, and compliance with traffic laws.
- **Highway driving:** Tests the system's ability to merge, change lanes, maintain safe distances, and react to high-speed scenarios.
- **Adverse conditions:** Includes scenarios with poor weather (rain, snow, fog), challenging lighting (glare, nighttime), and degraded road conditions to evaluate sensor performance and system robustness.
- **Emergency situations:** Simulates unexpected events such as sudden stops, evasive maneuvers to avoid obstacles, and system responses to equipment failures.

In summary, the evaluation of AD systems from any perspective necessitates a mix of simulation-based testing and real-world trials to guarantee their safety, reliability, and readiness for broad adoption.

### 2.2.6. SIMULATION LEVELS

Using different levels of virtual validation across various platforms and vehicles is a crucial aspect of the development lifecycle. These virtual validation techniques allow developers to thoroughly test and refine their systems at different stages of development, minimizing costs and maximizing efficiency and safety. Here's how they are typically used across different development phases:

**Model-in-the-loop (MiL):** During the early concept and design phase, MiL simulations are utilized to validate the system's conceptual integrity and functional behavior. These simulations are typically run on high-performance computing platforms, where the emphasis is on algorithm validation and system modeling rather than hardware interaction.

**Software-in-the-loop (SiL):** As the project moves into the software development phase, SiL simulations become more prevalent. These are executed on general-purpose computing environments such as X86 platforms, allowing for the validation of software components in a virtualized environment. This phase focuses on ensuring that the software behaves as expected in a simulated context that abstracts away from the hardware.

**Hardware-in-the-loop (HiL):** In the integration and testing phase, HiL simulations are critical. This is where the developed software is tested against the actual hardware, albeit in a controlled environment. Electronic Control Units (ECUs) and other hardware components are integrated into a test setup that simulates the vehicle's inputs and outputs. HiL tests are essential for uncovering any software-hardware interaction issues that were not evident during SiL testing.

**Vehicle-in-the-loop (ViL) and proving-groud-vehicle-in-the-loop (PG-ViL)**: ViL testing integrates the software and hardware within the context of the actual vehicle, allowing for real-world testing scenarios with certain simulated components. PG-ViL, often focusing on powertrain or specific processor-level simulations, is used for detailed analysis and optimization of specific system components or functions. These stages are crucial for final validations, ensuring that the system performs safely and efficiently in real-world conditions.

The transition from one validation technique to another is not strictly linear; instead, it involves back-and-forth iterations as insights gained from later stages lead to refinements in earlier stages. Moreover, these virtual validation stages are increasingly integrated with continuous integration/continuous deployment (CI/CD) pipelines, automating the validation process. This integration helps in continuously testing and validating software against a suite of predefined test cases across all development phases, ensuring that new code commits do not break existing functionalities. Despite their extensive use, virtual validation techniques face challenges such as the high-fidelity simulation of physical environments, integration complexities among different validation stages, and the scalability of tests to cover an exhaustive range of scenarios and conditions. Incorporating advancements in simulation technology, such as AI, would bring a more efficient and robust product development lifecycle.

### 2.2.7. OPERATIONS

Incorporating STV$^2$ processes into the autonomous driving system development cycle is crucial for iterative improvement and product quality optimization from a business standpoint. This approach enhances the feedback loop for ongoing system refinement and allows for early detection of potential issues, significantly cutting development costs and accelerating market readiness, thereby boosting competitive edge and customer satisfaction. Yet, deploying effective STV$^2$ strategies faces significant challenges: crafting exhaustive testing scenarios that mimic real-world conditions, setting industry-standard protocols for uniform results, and scaling testing methods to keep up with fast-evolving autonomous technology. Addressing these challenges is vital for staying at the forefront of technological progress and maintaining a competitive market stance.

# 3. STV$^2$ IMPLEMENTATION

## 3.1. SYSTEM ARCHITECTURE

An exemplary simulation platform tailored for STV$^2$ is presented. The design integrates inputs from system users and leverages a continuous integration and delivery (CI/CD) framework to ensure seamless operation between the backend and frontend systems.

The architecture is structured around a backend system that processes and manages data. At its core, the backend houses various assets critical to the simulation process. These include scenarios for testing, which are managed through a scenario editor that facilitates the creation and generalization of dynamic and traffic scenarios. Each scenario is evaluated to ensure it meets set criteria, supported by comprehensive model management that includes sensor models, dynamics models, and traffic models. Additionally, the simulation registry maintains all necessary simulation images, such as simulator images, dynamics simulation images, and sensor simulation images, ensuring resources are efficiently organized and readily accessible.

The campaign executor manages the orchestration and execution of simulation campaigns. It encompasses the campaign control, which directs the overall execution flow, and the simulation task executor, which undertakes specific simulation tasks. Within this framework, the simulation task executor is equipped with a sim task monitor and a sim result evaluator to track and assess simulation progress and outcomes. The simulator component within this executor runs the actual simulations, employing tools such as the scenario runner, virtual environment sim, vehicle dynamics sim, sensor sim, and traffic sim to replicate and analyze real-world driving conditions.

The unit under test (UUT) is an essential part of the simulation setup, featuring a virtual ECU (vECU), middleware, and a data adapter. These elements work together to mimic the vehicle's control systems under test conditions, providing a realistic platform for V&V activities.

On the frontend, the system offers an interactive user interface that allows users to configure, initiate, and monitor simulations. This interface provides access to various simulation settings and real-time control capabilities, making it a pivotal tool for researchers and engineers.

Evaluation of the simulation outcomes is handled by the simulation evaluator, which uses a series of metrics and a dashboard to provide insightful and actionable feedback on the performance of the autonomous systems being tested.

To enhance the system's realism and accuracy, the HiL reprocessing manager interfaces with an external HiL bench that includes a HiL reprocessing service. This setup allows for the reprocessing and integration of real-world data into the simulation environment, further enhancing the quality of the testing procedures. Meanwhile, the data connector facilitates the exchange of data with an external data platform, ensuring that data flows seamlessly for comprehensive analysis and enhancement of the simulation processes. An example is shown in FIGURE 2.

The integration of the backend and frontend through the CI/CD pipeline is a critical aspect of the architecture, ensuring that the system remains up to date with the latest technological advancements and operational best practices. This setup not only facilitates the continuous deployment of new features but also ensures that the platform can adapt to evolving testing requirements without disruptions.

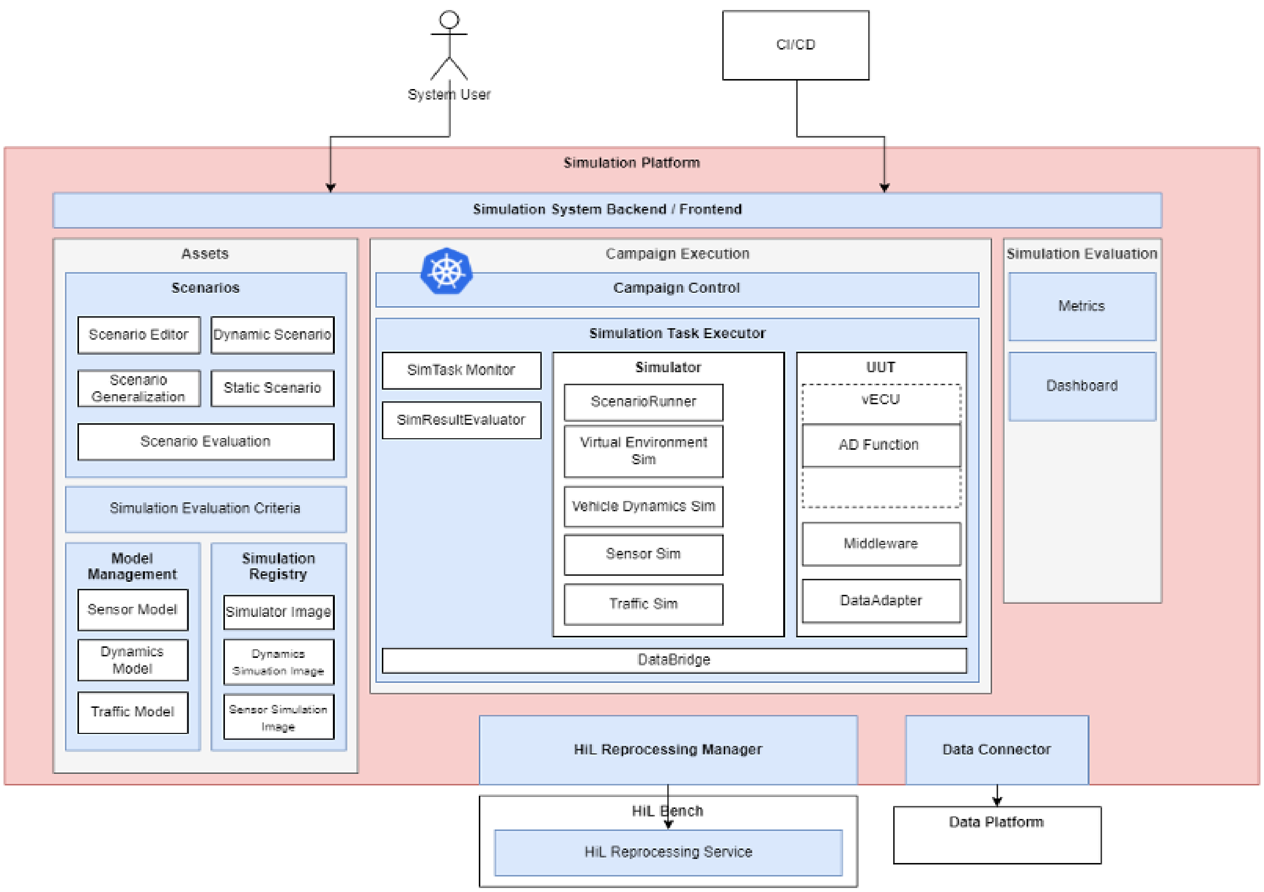


**FIGURE 2 AN EXEMPLARY SIMULATION PLATFORM TAILORED FOR THE STV²**

## 3.2. MODELS

### 3.2.1. SENSOR MODELS

Sensor models are crucial components in the simulation, testing, verification and validation (V&V) of autonomous driving systems. These models simulate the input from real-world sensors—such as cameras, lidar, radar, and ultrasonic sensors—that an autonomous vehicle relies on to perceive its environment [44]. By accurately replicating how sensors interact with diverse environmental conditions and scenarios, sensor models enable developers to rigorously assess the vehicle's decision-making algorithms without the need for extensive physical testing.

The development and validation of sensor models can broadly be categorized into two approaches: model-based and data-driven solutions.

**Model-based solutions:** These rely on predefined physics and mathematical models to simulate sensor behavior. Model-based sensor models use geometric and physical principles to predict how sensors will respond to different stimuli, considering factors like object distances, light reflection properties, and sensor noise. This approach is particularly effective for scenarios where the underlying physics are well-understood and can be accurately modeled, such as the propagation of radar waves or the reflection of light for lidar sensors.

**Data-driven solutions:** These solutions leverage learning techniques to build sensor models based on vast datasets collected from real-world driving situations. By training models on this data, data-driven approaches can capture complex and subtle patterns in sensor responses that might be overlooked or too intricate to be effectively modeled in a physics-based framework. These models are especially useful for handling unpredictable environmental variables such as varying lighting conditions for cameras or complex object shapes and materials for lidar detection. Data-driven models adapt and evolve as they are fed more data, continually improving their accuracy and reliability.

Both approaches have their merits and are often used in conjunction to capitalize on their strengths. Model-based methods provide a high level of control and interpretability, which is crucial for debugging and refining sensor algorithms. Conversely, data-driven methods offer robustness and adaptability, essential for coping with the diverse and dynamic nature of real-world environments. In practice, a hybrid approach often yields the best results, combining the predictive power of physical models with the adaptive capabilities of data-driven models to cover a broader spectrum of test conditions and improve the overall performance of autonomous driving systems.

Challenges ahead, particularly in achieving high fidelity in simulation that mirrors complex real-world conditions, is the accurate simulation of sensor degradation and failures under various environmental conditions such as fog, heavy rain, or direct sunlight, which can drastically affect sensor performance [45]. Additionally, modeling the interaction of sensors with a wide range of materials and surfaces—such as different types of road surfaces or the varied reflectivity of vehicles and other objects—is complex and computationally demanding [46]. Several advanced computational techniques are enhancing the realism and accuracy of sensor models. Key among these are generative adversarial networks (GANs) [47], neural radiance fields (NeRF) [48], and 3D Gaussian splatting [49]. GANs are instrumental in creating detailed synthetic datasets that closely replicate the variability in environmental conditions encountered by real-world sensors, thereby aiding in the robust training and validation of perception algorithms. NeRF contributes significantly by generating high-fidelity 3D reconstructions from 2D images, providing precise visual and spatial context for optical sensors such as cameras and lidar, which is crucial for simulating realistic driving scenarios. Additionally, 3D Gaussian splatting effectively processes point cloud data from lidar sensors, offering smoother and more continuous environmental representations that enhance the accuracy of sensor inputs. These improvements are pivotal for advancing the safety and efficacy of autonomous vehicles in diverse operating conditions.

### 3.2.2. VEHICLE DYNAMICS

The simulation of vehicle dynamics is integral for predicting the behavior of a self-driving vehicle under various operational scenarios [50]. An accurate vehicle dynamics model captures complex interactions across multiple subsystems including the chassis, suspension, tires, brakes, and powertrain. This model needs to consider the physical properties such as mass distribution, center of gravity, and inertia, as these parameters critically influence the vehicle's handling and stability.

Tire dynamics are crucial because tires are the primary contact points with the road, affecting traction, slip, and wear. Models like the Magic Formula or Fiala models are employed to articulate the intricate dynamics between the tire and varying road surfaces under different weather conditions [51]. The suspension system's accurate portrayal ensures the vehicle's response to road irregularities and dynamics during maneuvers such as cornering, accelerating, or braking is realistic, influencing ride quality and handling dynamics [52]. Aerodynamic forces become significant at higher speeds, impacting fuel efficiency, stability, and cabin noise levels, necessitating detailed aerodynamic modeling [53]. The drivetrain and powertrain dynamics [54], encompassing the engine, transmission, differential, and driveline, are also critical, especially in hybrid and electric vehicles where interactions between electric motors, batteries, and traditional engines need meticulous integration. Moreover, the simulation incorporates advanced control systems such as traction control, electronic stability control, and adaptive cruise control, which require precise calibration to function correctly under diverse conditions. Techniques such as software-in-the-loop (SIL) and hardware-in-the-loop (HIL) simulations [55] are crucial for testing the vehicle's electronic systems and software algorithms in a virtual environment, thus minimizing the reliance on physical testing, and enhancing the overall safety and reliability of the autonomous systems.

### 3.2.3. MULTI-AGENT INTERACTIONS

Multi-agent interaction systems work better in autonomous driving because they enable effective coordination and collaboration between agents, which is crucial in complex traffic environments [56]. Multi-agent reinforcement learning (MARL) is a powerful method used in multi-agent systems as it considers the interaction between agents and allows for decentralized training, making it highly scalable [57]. Recently, MARL has been greatly advanced and successfully applied to a variety of complex multi-agent systems such as games [58], traffic light control [59], and fleet management [60]. The MARL algorithms have also been applied to autonomous driving [61], with the objective of accomplishing autonomous driving tasks cooperatively.

In multi-agent interaction systems involving multiple agents (e.g., AVs,), the agent's actions affect not only their own rewards but also the rewards of other agents [62].

In contrast to single-agent reinforcement learning (RL), multi-agent RL assumes the existence of more than one agent interacting within a shared environment. MARL problems are formalized using Markov games [64], an extension of MDP (Markov Decision Process) to multiple agents. In general, a Markov game (MG) is a tuple:

$$(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{\mathbb{R}^i\}_{i\in\mathcal{N}}, \gamma)$$

where

- $\mathcal{N}$ is the number of agents
- $\mathcal{S}$ is the state space consisting of the states observed by all the agents
- $\{\mathcal{A}^i\}$ is the action space of the ith agent such that $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ is the joint action space of all the agents

$\mathcal{P}$ represents a transition probability function for each state, which can be either known or unknown for a given problem

$\mathbb{R}^i$ is the reward function, which is a scalar value obtained by the ith agent on making a state transition

$\gamma$ is the discount factor such that $\gamma \in [0,1)$

The goal is to maximize the discounted sum of rewards over a given episode.

A MARL problem can be defined as belonging to one of the following three scenarios depending on the nature of interaction among the agents: cooperative, competitive, and mixed. In a fully cooperative scenario, the agents are expected to collaborate with each other to solve a common task and aim to maximize a common return jointly for all the agents. In a fully competitive scenario, the agents compete in a zero-sum game, where only one agent may emerge victorious; therefore, the agents are expected to maximize their individual rewards while trying to minimize other agents' rewards. Some examples of competitive scenarios are car racing, blackjack, chess, etc. In a mixed scenario, the aim is to maintain a balance between collaborating and competing. Some examples are soccer, basketball, and other team games.

It is essential to note that CAVs in real-world settings inherently assume a cooperative scenario where each vehicle is expected to cooperate while maximizing its rewards. Multi-agent interaction simulation can simulate autonomous vehicles in fully interactive scenarios with a variable number of intelligent agents, which can accelerate the training and testing of autonomous driving algorithms [63].

## 3.3. EVALUATIONS

Evaluating the performance of a self-driving vehicle like the $STV^2$ requires a comprehensive and multifaceted approach to ensure it meets the necessary safety standards and operational efficacy within its operational design domain (ODD). Here are some key aspects and methodologies that could be integral to the evaluation process:

- **ODD-specific evaluation criteria:** It is essential to define the specific conditions under which the $STV^2$ is expected to operate, including geographic locations, types of roads, weather conditions, and times of day. Evaluations should test the vehicle's performance against these conditions to ensure it handles its designated ODD effectively. This involves creating scenarios that simulate these conditions, such as navigating urban environments with high pedestrian traffic or driving in adverse weather conditions like fog or heavy rain.

- **Custom evaluation templates:** Developing custom templates or protocols that guide the testing process can help standardize evaluations. These templates would outline specific tests for different aspects of the vehicle's operation, such as response to emergency situations, interaction with various road users, compliance with speed limits, and maneuvering through complex traffic situations. Each template would be tailored to test functionalities and ensure they meet predefined benchmarks.

- **Traffic rules and regulations compliance:** The vehicle must adhere to all traffic laws and regulations within its operating region. This involves testing the vehicle's ability to recognize and appropriately respond to road signs, traffic signals, road markings, and give-way rules. Simulations and closed-circuit road tests can be used to assess compliance and the vehicle's decision-making in scenarios where traffic rules must be interpreted under different conditions.

- **Scenario-based testing:** Creating specific driving scenarios to test the vehicle's responses in both common and rare situations. This could involve scenarios like sudden stops, unexpected obstacles on the road, or navigation on poorly marked roads. The aim is to ensure the vehicle can handle unexpected situations safely and efficiently.

- **Software and cybersecurity assessments:** Ensuring the software that controls the vehicle is robust against failures and cyber threats. This involves regular updates and patches, as well as rigorous testing of the software's response to simulated attacks and system failures.

- **Sensor and component testing:** Regular testing of the vehicle's sensors and other critical components to ensure they function correctly under various conditions. This includes lidar, radar, cameras, and ultrasonic sensors, which must accurately detect and interpret the vehicle's surroundings and any potential hazards.

- **User experience feedback:** Incorporating feedback from operators or passengers within the $STV^2$ can provide insights into the real-world experience of interacting with the vehicle's systems. This includes ease of use, comfort, trust in the vehicle's driving decisions, and overall satisfaction.

- **Compliance with international standards:** Finally, ensuring that the $STV^2$ meets international safety and performance standards, such as those set by the IEEE, ISO, SAE, and local transportation authorities, can help gain regulatory approval and public trust.

By addressing the above aspects in the evaluation process, developers can ensure that the $STV^2$ is not only compliant with all necessary regulations but also safe and reliable in its intended operational environments.

# 4. TECHNOLOGIES AND TREND

## 4.1. VIRTUALIZATIONS

### 4.1.1. GENERAL

Virtualization technologies have increasingly become more critical in developing autonomous driving systems. Recent advancements have propelled virtualization from a supportive role to a central strategy in $STV^2$ processes. Highlighted below are key examples that illustrate the critical role.

### 4.1.2. NAVIGATING URBAN COMPLEXITY: THE ROLE OF CARLA

At the forefront of this technological evolution is CARLA, an open-source simulation tool that offers a sophisticated platform for testing autonomous vehicles in complex, dynamic urban environments. By simulating detailed scenarios—from sudden weather changes to erratic pedestrian movements—CARLA provides a risk-free setting for developers to refine and validate vehicle responses to diverse real-world challenges. The enhanced version, Synkrotron OASIS, builds on this foundation with a more comprehensive suite of tools, including advanced scenario editing, improved sensor models, and capabilities for connecting to various hardware for real-time simulations. Additionally, it facilitates large-scale simulations in the cloud, providing tools that support expansive and complex testing environments.

### 4.1.3. CLOUD COMPUTING: AWS ROBOMAKER'S SCALABLE SOLUTIONS

AWS RoboMaker exemplifies the transformative impact of cloud computing in autonomous vehicle development. By facilitating the simulation of vast fleets of virtual vehicles across an array of environments, AWS RoboMaker enables parallel testing that drastically compresses development schedules. This scalable platform supports the iterative refinement of autonomous systems, ensuring comprehensive testing is both feasible and efficient.

### 4.1.4. PIONEERING VIRTUAL VALIDATION: BMW IFACTORY

BMW's iFACTORY initiative represents the pioneering use of digital twin technology in the automotive industry. By creating virtual replicas of their vehicles, BMW can extensively test and optimize vehicle performance before physical prototypes are ever built. The iFACTORY initiative demonstrates how digital twins have become an integral part of developing safer, more reliable autonomous vehicles by enabling thorough pre-emptive testing and validation.

### 4.1.5. SCENARIO GENERATION THROUGH AI: NVIDIA'S DRIVE SIM

NVIDIA's DRIVE Sim stands out for its use of AI to automatically generate a wide range of driving scenarios, including rare or dangerous situations that would be difficult or impossible to replicate in real-world testing. This capability ensures that autonomous vehicles are exposed to and can learn from an exhaustive set of challenges, preparing them for anything they might encounter on the road.

These examples underscore the transformative role of virtualization technologies in developing autonomous driving systems. With these advanced applications, virtualization is setting new standards for how autonomous vehicles are brought from conception to reality.

### 4.1.6. GCP CLOUD ROBOTICS CORE

GCP Cloud Robotics Core represents a groundbreaking framework that leverages standard Kubernetes management tools alongside various open-source packages. At its core, it operates across four distinct layers. The foundational Layer 0 facilitates Kubernetes deployment on robots, ensuring efficient resource utilization and seamless operation. Layer 1 focuses on establishing robust connectivity and security protocols for managing fleets of robots, fostering reliability, and safeguarding sensitive data. Moving up to Layer 2, the framework enables streamlined application management, empowering developers to deploy and monitor software with ease. Finally, Layer 3 offers managed repositories, facilitating the storage and versioning of crucial robotics applications and configurations. Through these layers, GCP Cloud Robotics Core provides a comprehensive and scalable solution for orchestrating complex robotic systems in cloud environments.

## 4.2. SAFETY AND SECURITY IN THE ERA OF AGI

### 4.2.1. ALIGNMENT

Artificial general intelligence (AGI) systems' decisions and actions should be congruent with human values, ethical standards, and societal norms. This concept is foundational to the development and deployment of AGI systems that are not only technologically advanced but also socially responsible and beneficial. Addressing the dual challenges of safety and security in AGI involves a commitment to ethical guidelines and incorporating findings from advanced AI safety research. Ethical guidelines provide a framework for the responsible development and deployment of AGI systems, emphasizing the importance of transparency, accountability, and alignment with human values. Meanwhile, AI safety research is an ongoing endeavor that seeks to identify and mitigate the potential risks associated with AGI. This field of research is crucial for developing effective strategies to ensure that AGI systems operate safely and predictably, even as they evolve and adapt over time.

Specifically, ensuring safety in the context of AGI is about guaranteeing that AGI decision-making processes are in harmony with human ethical standards and values. This challenge is multifaceted, requiring a deep understanding of both the technical capabilities of AGI systems and the ethical considerations they provoke. For instance, an AGI-powered autonomous vehicle must navigate not just the physical roads but also complex moral decisions, such as those presented by theoretical dilemmas like the "trolley problem." To address these concerns, the development of AGI systems necessitates the integration of ethical decision-making frameworks. These frameworks guide the AGI's decisions, ensuring they result in outcomes beneficial to humans and the environment. This might include programming AGI systems with a hierarchy of ethical considerations or developing algorithms that can interpret and apply ethical principles in diverse scenarios. Moreover, safety research in AI must evolve to encompass the broader implications of AGI, exploring new methodologies for risk assessment, mitigation, and the development of robust fail-safe mechanisms. These mechanisms are designed to automatically limit the AGI's capabilities or shut it down entirely in case it deviates from its intended operational parameters, thereby preventing harm.

On the flip side, the security of AGI systems focuses on fortifying them against external threats, notably cyber-attacks that could hijack these systems for malicious purposes. The sophistication of AGI systems makes them potent tools in the wrong hands, capable of causing unprecedented damage if compromised. To safeguard against such scenarios, $STV^2$ processes must be augmented with advanced cybersecurity measures tailored to the unique vulnerabilities of AGI systems. This involves not only traditional cybersecurity protocols such as robust encryption and intrusion detection systems, but also novel approaches designed to protect against attacks that exploit the specific characteristics of AGI, such as adversarial attacks on machine learning models. The development of secure coding practices for AGI development is another crucial step. These practices ensure that AGI software is designed from the ground up to be resistant to exploitation, incorporating secure design principles that address the unique challenges posed by AGI technology.

## 4.2.2. SIMULATIONS AND GENERATIVE AI

### 4.2.2.1 General

Generative AI significantly enhances STV$^2$ by automating test case generation, creating realistic simulation environments, and improving the efficiency and effectiveness of testing processes. It facilitates data-driven decision-making and continuous testing, helping to identify and mitigate biases, and ensuring systems are thoroughly evaluated and reliable before deployment.

### 4.2.2.2 Large language models (LLMs)/world models

LLMs like GPT are advancing STV$^2$ by automating the generation of complex and nuanced test cases. For example, in software development, an LLM can automatically generate test scripts based on the specifications of the software, creating a wide range of scenarios that might include edge cases not immediately obvious to human testers. This capability is crucial for comprehensive testing coverage. In analysis, LLMs can process and interpret vast amounts of test logs, extracting meaningful insights and summarizing results efficiently, which is particularly beneficial for identifying subtle bugs or issues in large software projects. Moreover, LLMs facilitate realistic user interaction simulations, as seen in the testing of AI-driven customer service chatbots, where they generate conversational scenarios to test the chatbot’s responses under various conditions, thereby enhancing the system’s robustness and user experience.

World models, used in the development of autonomous systems, provide dynamic simulations of the real world that allow for extensive testing of systems like self-driving cars. For instance, Tesla’s Autopilot development leverages such models to simulate driving scenarios, testing the vehicle’s responses to different road conditions, obstacles, and unpredictable events without the need for real-world exposure. This method significantly reduces the risks and costs associated with physical testing. In predictive modeling, world models can forecast the outcomes of systems in different environments, such as predicting the behavior of a vehicle in varying weather conditions, enabling preemptive adjustments to the system. Additionally, these models support data-driven decision-making by simulating countless scenarios, thereby aiding in the validation of complex systems where different driving conditions and emergencies can be simulated and tested thoroughly. The continuous learning and adaptation feature of world models ensures that the simulated environments evolve with the system, providing an up-to-date and relevant testing framework.

The integration of LLMs and world models into $STV^2$ processes represents a significant advancement, delivering more automated, precise, and efficient testing and validation capabilities. This is essential for developing reliable and high-performing systems, marking a milestone in the evolution of $STV^2$ methodologies.

#### 4.2.2.3 Autonomous agents

In autonomous driving, autonomous agents are used to create realistic and challenging scenarios for testing and validation. For instance, in a simulated urban environment, these agents can represent different types of vehicles, such as buses, trucks, and cars, each with its own unique driving patterns and behaviors. They can also simulate erratic pedestrian movements, like suddenly crossing the road, to test the autonomous vehicle's sensor accuracy and decision-making capabilities. A concrete example is the use of autonomous agents to simulate emergency situations, such as a child chasing a ball onto the street or a vehicle suddenly braking to avoid an obstacle. These scenarios test the autonomous vehicle's ability to detect and respond to unexpected events promptly and safely. Another example is the simulation of complex traffic conditions, such as congested urban centers or multi-lane highways with merging traffic. Autonomous agents can mimic aggressive drivers who change lanes unpredictably or tailgate, providing a comprehensive test environment to ensure the autonomous vehicle can handle real-world driving challenges effectively and safely. In these simulations, the autonomous agents not only behave according to predefined patterns but also adapt their behaviors based on the autonomous vehicle's actions, creating a dynamic and interactive testing environment.

#### 4.2.2.4 Visual language model (VLM)

The utilization of large language models (LLMs) is significantly enhancing the $STV^2$ framework for autonomous driving.

By integrating LLMs, the system's comprehension of context in various driving scenarios is improved, utilizing a set of predefined question-answer pairs. Given that camera sensors are among the most prevalent in autonomous vehicles, the progression toward language integration has been facilitated by visual language models (VLMs). VLMs combine visual and textual data, allowing for spatial reasoning augmented by the capabilities of pre-trained LLMs. The system is designed to articulate its visual perceptions in natural language and to determine suitable actions, such as stopping at a red light.

Furthermore, LLMs contribute to the improvement of driving-related question-and-answer performance. For instance, Chen and colleagues have compiled a dataset comprising 160,000 questions. This extensive dataset is instrumental in developing a proprietary LLM tailored for autonomous driving applications [65].

# 5. REFERENCES


[1] Maurer, Markus, J. Christian Gerdes, Barbara Lenz, and Hermann Winner. *Autonomous Driving: Technical, Legal and Social Aspects.* Springer Nature, 2016.

[2] Chen, Long, Yuchen Li, Chao Huang, Bai Li, Yang Xing, Daxin Tian, Li Li et al. “Milestones in autonomous driving and intelligent vehicles: Survey of surveys.” *IEEE Transactions on Intelligent Vehicles* 8, no. 2 (2022): 1046–1056.

[3] McKinsey Reports. “Autonomous Driving’s Future: Convenient and Connected,” January 6, 2023. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected

[4] Omeiza, Daniel, Helena Webb, Marina Jirotka, and Lars Kunze. “Explanations in autonomous driving: A survey.” *IEEE Transactions on Intelligent Transportation Systems* 23, no. 8 (2021): 10142–10162.

[5] Doshi-Velez, Finale, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O’Brien, Kate Scott et al. “Accountability of AI under the law: The role of explanation.” *Development in Practice* 663 (2007): 663–665.

[6] Mehdipour, Noushin, Matthias Althoff, Radboud Duintjer Tebbens, and Calin Belta. “Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges.” *Automatica* 152 (2023): 110692.

[7] Tabani, Hamid, Leonidas Kosmidis, Jaume Abella, Francisco J. Cazorla, and Guillem Bernat. “Assessing the adherence of an industrial autonomous driving framework to ISO 26262 software guidelines.” In *Proceedings of the 56th Annual Design Automation Conference 2019,* pp. 1–6. June 2019.

[8] Zhao, Tong, Ekim Yurtsever, Joel A. Paulson, and Giorgio Rizzoni. “Formal certification methods for automated vehicle safety assessment.” *IEEE Transactions on Intelligent Vehicles* 8, no. 1 (2022): 232–249.

[9] Dia, Hussein, Richard Tay, Ryszard Kowalczyk, Saeed Bagloee, Eleni Vlahogianni, and Andy Song. "Artificial intelligence tests for certification of autonomous vehicles." *Academia Letters* 3 (2021).

[10] Ponn, Thomas, Dirk Fratzke, Christian Gnandt, and Markus Lienkamp. "Towards Certification of autonomous driving: Systematic test case generation for a comprehensive but economically-feasible assessment of lane keeping assist algorithms." In *VEHITS 2019—Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems,* pp. 333–342. May 2019.

[11] Widen, William H., and Philip Koopman. "Autonomous vehicle regulation & trust: The impact of failures to comply with standards." *UCLA Journal of Law & Technology* 27, no. 3 (2022): 169.

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

[13] Bösch, Patrick M., Felix Becker, Henrik Becker, and Kay W. Axhausen. "Cost-based analysis of autonomous mobility services." *Transport Policy* 64 (2018): 76–91.

[14] Tesla, Inc. "Full Self-Driving Capability Subscriptions," last accessed January 11, 2024, [Online]. Available: https://www.tesla.com/support/full-self-driving-subscriptions

[15] Lee, Dave. "Slow-and-Steady Waymo Is Winning the Self-Driving Race," January 9, 2024, [Online]. Available: https://www.bloomberg.com/opinion/articles/2024-01-09/slow-and-steady-waymo-is-winning-the-self-driving-race-for-alphabet

[16] Grant, Andrew. "This $220 Billion Market Opens Up a Path for Driverless Cars," November 29, 2022, [Online]. Available: https://www.bloomberg.com/news/newsletters/2022-11-29/this-220-billion-market-opens-up-a-path-for-driverless-cars

[17] Chao, Qianwen, Huikun Bi, Weizi Li, Tianlu Mao, Zhaoqi Wang, Ming C. Lin, and Zhigang Deng. "A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving." *Computer Graphics Forum* 39, no. 1, (2020): 287–308.

[18] Alghodhaifi, Hesham, and Sridhar Lakshmanan. "Autonomous vehicle evaluation: A comprehensive survey on modeling and simulation approaches." *IEEE Access* 9 (2021): 151531–151566.

[19] Hu, Xuemin, Shen Li, Tingyu Huang, Bo Tang, Rouxing Huai, and Long Chen. "How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence." *IEEE Transactions on Intelligent Vehicles* 9, no. 1 (2023): 593–612.

[20] Hauer, Florian, Tabea Schmidt, Bernd Holzmüller, and Alexander Pretschner. “Did we test all scenarios for automated and autonomous driving systems?” In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2950–2955. October 2019.

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

[22] Wishart, Jeffrey, Steve Como, Uilani Forgione, Jack Weast, Lewis Weston, Andrew Smart, and George Nicols. “Literature review of verification and validation activities of automated driving systems.” *SAE International Journal of Connected and Automated Vehicles* 3 (2020): 267–323.

[23] Nordenström, Martin. “Future certification of autonomous vehicles and the use of virtual testing methods.” (2020).

[24] Feng, Shuo, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X. Liu. “Dense reinforcement learning for safety validation of autonomous vehicles.” *Nature* 615, no. 7953 (2023): 620–627.

[25] Sato, Takami, Junjie Shen, Ningfei Wang, Yunhan Jack Jia, Xue Lin, and Qi Alfred Chen. “Security of deep learning based lane keeping system under physical-world adversarial attack.” arXiv preprint arXiv:2003.01782 (2020).

[26] Stocco, Andrea, and Paolo Tonella. “Towards anomaly detectors that learn continuously.” In *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW),* pp. 201–208. October 2020.

[27] Ryan, Cian, Finbarr Murphy, and Martin Mullins. “End-to-end autonomous driving risk analysis: A behavioural anomaly detection approach.” *IEEE Transactions on Intelligent Transportation Systems* 22, no. 3 (2020): 1650–1662.

[28] Hao, Kunkun, Wen Cui, Yonggang Luo, Lecheng Xie, Yuqiao Bai, Jucheng Yang, Songyang Yan, Yuxi Pan, and Zijiang Yang. “Adversarial safety-critical scenario generation using naturalistic human driving priors.” *IEEE Transactions on Intelligent Vehicles* (2023).

[29] Hao, Kunkun, Lu Liu, Wen Cui, Jianxing Zhang, Songyang Yan, Yuxi Pan, and Zijiang Yang. "Bridging data-driven and knowledge-driven approaches for safety-critical scenario generation in automated vehicle validation." arXiv preprint arXiv:2311.10937 (2023).

[30] Herberth, Roland, Sidney Körper, Tim Stiesch, Frank Gauterin, and Oliver Bringmann. "Automated scheduling for optimal parallelization to reduce the duration of vehicle software updates." *IEEE Transactions on Vehicular Technology* 68, no. 3 (2019): 2921–2933.

[31] Riehl, Damien A. "Car minus driver: autonomous vehicles driving regulation, liability, and policy." *Computer & Internet Lawyer* 35, no. 5 (2018): 1–18.

[32] Widen, William H., and Philip Koopman. "Autonomous vehicle regulation & trust: The impact of failures to comply with standards." *UCLA Journal of Law & Technology* 27, no. 3 (2022): 169–261.

[33] Schneider, Tobias, Joana Hois, Alischa Rosenstein, Sabiha Ghellal, Dimitra Theofanou-Fülbier, and Ansgar R. S. Gerlicher. "Explain yourself! transparency for positive UX in autonomous driving." In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems,* pp. 1–12. May 2021.

[34] Häuslschmid, Renate, Max von Buelow, Bastian Pfleging, and Andreas Butz. "SupportingTrust in autonomous driving." In *Proceedings of the 22nd International Conference on Intelligent User Interfaces,* pp. 319–329. March 2017.

[35] The Brainy Insights Report. "Autonomous Vehicle Market," 2022. [Online]. Available: https://www.thebrainyinsights.com/report/autonomous-vehicle-market-13887

[36] Gran View Research Report. "Autonomous Vehicle Market Size and Share Report, 2030," 2021. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/autonomous-vehicles-market

[37] Next Move Strategy Consulting Report. "Autonomous Vehicle Market," January 2023. [Online]. Available: https://www.nextmsc.com/report/autonomous-vehicle-market

[38] McKinsey Reports. "Autonomous Driving's Future: Convenient and Connected," January 6, 2023. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected

[39] Lienert, Paul and Joseph White. "Musk's Bold Goal of Selling 20 Million EVs Could Cost Tesla Billions," August 30, 2022. [Online]. Available: https://www.reuters.com/technology/musks-bold-goal-selling-20-mln-evs-could-cost-tesla-billions-2022-08-30/

[40] Roland Berger Reports. “Autonomous Driving Disruptive Innovation That Promises to Change the Automotive Industry as We Know It—It’s Time for Every Player to Think: Act!” November 2014, [Online]. Available: https://www.rolandberger.com/publications/publication_pdf/roland_berger_tab_autonomous_driving.pdf

[41] Siemens Reports. “Create the Trust Your Customers Need Autonomous Vehicle Development.” 2021. [Online]. Available: https://resources.sw.siemens.com/en-US/infographic-verification-and-validation-in-autonomous-vehicle-development

[42] BIS Research Reports. “Autonomous Vehicle Simulation Solutions Market—A Global Market and Regional Analysis,” 2022. [Online]. Available: https://bisresearch.com/industry-report/autonomous-vehicle-simulation-solution-market.html

[43] Fletcher, Ryan, Abhijit Mahindroo, Nick Santhanam, and Andreas Tschiesner. “The Case for an End-to-End Automotive-Software Platform,” January 16, 2020. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-case-for-an-end-to-end-automotive-software-platform

[44] Schlager, Birgit, Stefan Muckenhuber, Simon Schmidt, Hannes Holzer, Relindis Rott, Franz Michael Maier, Kmeid Saad et al. “State-of-the-art sensor models for virtual testing of advanced driver assistance systems/autonomous driving functions.” *SAE International Journal of Connected and Automated Vehicles* 3, no. 12-03-03-0018 (2020): 233–261.

[45] Vargas, Jorge, Suleiman Alsweiss, Onur Toker, Rahul Razdan, and Joshua Santos. “An overview of autonomous vehicles sensors and their vulnerability to weather conditions.” *Sensors* 21, no. 16 (2021): 5397.

[46] Yeong, De Jong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. “Sensor and sensor fusion technology in autonomous vehicles: A review.” *Sensors* 21, no. 6 (2021): 2140.

[47] Yang, Zhenpei, Yuning Chai, Dragomir Anguelov, Yin Zhou, Pei Sun, Dumitru Erhan, Sean Rafferty, and Henrik Kretzschmar. “SurfelGAN: Synthesizing realistic sensor data for autonomous driving.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 11118–11127. June 2020.

 

[48] Zhong, Shaohong, Alessandro Albini, Oiwi Parker Jones, Perla Maiolino, and Ingmar Posner. “Touching a NeRF: Leveraging neural radiance fields for tactile sensory data generation.” In *Conference on Robot Learning,* pp. 1618–1628. November 2023.

[49] Zhou, Xiaoyu, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. “Driving Gaussian: Composite Gaussian splatting for surrounding dynamic autonomous driving scenes.” arXiv preprint arXiv:2312.07920 (2023).

[50] Kong, Jason, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. “Kinematic and dynamic vehicle models for autonomous driving control design.” In *2015 IEEE Intelligent Vehicles Symposium (IV),* pp. 1094–1099. June 2015.

[51] Fors, Victor. *Autonomous Vehicle Maneuvering at the Limit of Friction.* Vol. 2102. Linköping University Electronic Press, 2020.

[52] Kang, Juyong, Wongun Kim, Jongseok Lee, and Kyongsu Yi. “Design, implementation, and test of skid steering-based autonomous driving controller for a robotic vehicle with articulated suspension.” *Journal of Mechanical Science and Technology* 24 (2010): 793–800.

[53] Jessing, Christoph, Daniel Stoll, Timo Kuthada, and Jochen Wiedemann. “New horizons of vehicle aerodynamics.” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 231, no. 9 (2017): 1190–1202.

[54] Shao, Yunli. “Optimization and Evaluation of Vehicle Dynamics and Powertrain Operation for Connected and Autonomous Vehicles.” PhD diss., University of Minnesota, 2019.

[55] Cime, Karina Meneses, Mustafa Ridvan Cantas, Garrett Dowd, Levent Guvenc, Bilin Aksun Guvenc, Archak Mittal, Adit Joshi, and James Fishelson. Hardware-in-the-Loop, Traffic-in-the-Loop and Software-in-the-Loop Autonomous Vehicle Simulation for Mobility Studies. No. 2020-01-0704. SAE Technical Paper, 2020.

[56] Candela, Eduardo, et al. “Transferring multi-agent reinforcement learning policies for autonomous driving using sim-to-real.” In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8814–8820. October 2022.

[57] Duan, Weiping, Zhongyi Tang, Wei Liu, and Hongbiao Zhou. “Autonomous driving planning and decision making based on game theory and reinforcement learning.” *Expert Systems* 40 (2023): e13191.

[58] Vinyals, Oriol, et al. “Grandmaster level in StarCraft II using multi-agent reinforcement learning.” *Nature* 575 (2019): 350–354.

[59] Chu, Tianshu, Jie Wang, Lara Codecà, and Zhaojian Li. "Multi-agent deep reinforcement learning for large-scale traffic signal control." *IEEE Transactions on Intelligent Transportation Systems* 21, no. 3 (2019): 1086–1095.

[60] Lin, Kaixiang, Renyu Zhao, Zhe Xu, and Jiayu Zhou. "Efficient large-scale fleet management via multi-agent deep reinforcement learning." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* (2018): 1774–1783.

[61] Palanisamy, Praveen. "Multi-agent connected autonomous driving using deep reinforcement learning." In *2020 International Joint Conference on Neural Networks (IJCNN),* pp. 1–7. July 2020.

[62] Hua, Min, Dong Chen, Xinda Qi, Kun Jiang, Zemin Eitan Liu, Quan Zhou, and Hongming Xu. "Multi-agent reinforcement learning for connected and automated vehicles control: Recent advancements and future prospects." arXiv preprint arXiv:2312.11084 (2023).

[63] Chen, Dong, Longsheng Jiang, Yue Wang, and Zhaojian Li. "Autonomous driving using safe reinforcement learning by incorporating a regret-based human lane-changing decision model." In *2020 American Control Conference (ACC),* pp. 4355–4361. July 2020.

[64] Littman, Michael L. "Markov games as a framework for multi-agent reinforcement learning." In *Machine Learning Proceedings 1994,* pp. 157–163*.* July 1994.

[65] Chen, Long, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. "Driving with LLMs: Fusing object-level vector modality for explainable autonomous driving." arXiv preprint arXiv:2310.01957 (2023).

# RAISING THE WORLD'S STANDARDS

3 Park Avenue, New York, NY 10016-5997 USA http://standards.ieee.org

Tel.+1732-981-0060 Fax+1732-562-1571