



AUTONOMOUS DRIVING WORKING GROUP

**SIMULATION, TESTING,
VERIFICATION, AND VALIDATION
(STV²) OF AUTONOMOUS
DRIVING**

Authored by

Autonomous Driving Working Group

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

Copyright © 2024 by The Institute of Electrical and Electronics Engineers, Inc.

All rights reserved. 9 October 2024. Printed in the United States of America.

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

IEEE is a registered trademark in the U. S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated. All other trademarks are the property of the respective trademark owners.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system, or otherwise, without the prior written permission of the publisher.

Find IEEE standards and standards-related product listings at: <http://standards.ieee.org>.

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....	6
ABSTRACT.....	6
1. INTRODUCTION.....	7
1.1. GENERAL	7
1.2. CHALLENGES OF AUTONOMOUS DRIVING.....	7
1.2.1. GENERAL.....	7
1.2.2. SAFETY AND SECURITY	7
1.2.3. REGULATIONS.....	8
1.2.4. BENEFITS.....	9
1.3. THE ROLE OF STV ²	9
1.3.1. RESEARCH AND DEVELOPMENT.....	9
1.3.2. VERIFICATION AND VALIDATION	10
1.3.3. OPERATIONS AND ASSURANCE.....	14
1.4. MARKET LANDSCAPE.....	15
2. OVERVIEW OF STV ²	16
2.1. THE LIFECYCLE OF STV ²	16
2.2. KEY ELEMENTS AND GAP ANALYSES	17
2.2.1. SCENARIOS.....	17
2.2.2. SIMULATION MODELS.....	18
2.2.3. BENCHMARKS AND CHALLENGES	19
2.2.4. FURTHER DEVELOPMENT AND BEYOND	19
2.2.5. EVALUATIONS.....	20
2.2.6. SIMULATION LEVELS	21
2.2.7. OPERATIONS.....	22
3. STV ² IMPLEMENTATION.....	23
3.1. SYSTEM ARCHITECTURE	23
3.2. MODELS.....	25
3.2.1. SENSOR MODELS	25
3.2.2. VEHICLE DYNAMICS	26
3.2.3. MULTI-AGENT INTERACTIONS	27
3.3. EVALUATIONS.....	28
4. TECHNOLOGIES AND TREND.....	30
4.1. VIRTUALIZATIONS.....	30
4.1.1. GENERAL	30
4.1.2. NAVIGATING URBAN COMPLEXITY: THE ROLE OF CARLA	30
4.1.3. CLOUD COMPUTING: AWS ROBOMAKER'S SCALABLE SOLUTIONS	30
4.1.4. PIONEERING VIRTUAL VALIDATION: BMW IFACTORY	31
4.1.5. SCENARIO GENERATION THROUGH AI: NVIDIA'S DRIVE SIM.....	31
4.1.6. GCP CLOUD ROBOTICS CORE	31
4.2. SAFETY AND SECURITY IN THE ERA OF AGI	32
4.2.1. ALIGNMENT.....	32
4.2.2. SIMULATIONS AND GENERATIVE AI.....	33
4.2.2.1 GENERAL	33

4.2.2.2	LARGE LANGUAGE MODELS (LLMs)/WORLD MODELS	33
4.2.2.3	AUTONOMOUS AGENTS	34
4.2.2.4	VISUAL LANGUAGE MODEL (VLM)	34
5.	REFERENCES.....	35



SIMULATION, TESTING, VERIFICATION, AND VALIDATION (STV²) OF AUTONOMOUS DRIVING

ABSTRACT

STV² 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 STV², as well as how the full lifecycle of autonomous driving systems is covered, is presented in this white paper. STV² is part of the infrastructure and tooling layer of autonomous driving architecture.

STV²'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 (STV²) are critical to addressing these challenges. This white paper aims to provide an in-depth exploration of STV², 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 STV²

1.3.1. RESEARCH AND DEVELOPMENT

In the R&D of autonomous vehicles, STV² 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, STV² leads the progression from initial design to a fully functional AV prototype. Through these comprehensive steps, STV² 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 Decouples hardware and software development	Brute force for high coverage Requires large computation power Tests only for known scenarios 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 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 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 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 Formalizes non-functional requirements Strengthens system security	Difficult to set up systematically No coverage schemes 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) Enables designers to foresee system interface failures	Depends heavily on human knowledge 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 Covers wide range of electrical systems	Depends heavily on human knowledge Labor intensive 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 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 Validates all systems at once Comprehensive view and coverage Standardizes scenario storage format and tagging	High effort for coverage Unclear coverage Most of the test cases are redundant 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 Open data sets, such as nuScenes		Reuse generated test cases from dependency database	Improved transparency Automatically considers dependencies to external environment and internal functions Automates major part of test procedure Standardizes scenario storage format and tagging Sharing test scenarios across V-model abstraction levels	High effort to set up AI-based test environment Needs large computation power 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.