Scenario-Based Validation for SAE Level 2+ Features Using Simulation-in-the-Loop (SiL) Systems

Gaurav Pokharkar

Independent Research
gpokhark@asu.edu

Abstract:
As SAE Level 2+ Advanced Driver Assistance Sys- tems (ADAS) become increasingly integrated into production vehicles, the need for structured and repeatable validation strategies becomes paramount to ensure functional safety under real-world conditions. Traditional road and vehicle-level testing methods are not only resource-intensive and time-consuming but also fall short in covering edge cases and maintaining scalability across development cycles. This paper presents a scenario-based validation framework utilizing Simulation-in-the- Loop (SiL) systems, which offers a scalable, cost-effective, and high-fidelity virtual testing environment. Central to this ap- proach is the use of the NHTSA-defined Operational Design Domain (ODD) taxonomy, which guides systematic scenario generation across physical infrastructure, operational constraints, objects, connectivity, environmental conditions, and zones. This structured ODD mapping enables traceability, risk-based test prioritization, and efficient scenario classification into Normal, Stress, Edge, and Out-of-ODD categories. This paper details a modular SiL architecture, automation pipelines, and evaluation metrics aligned with ISO 26262 [4] safety goals. It focuses on validating key SAE Level 2+ features such as Highway Assist, and includes pseudo-code, tool-chain integrations, and visual artifacts to demonstrate practical deployment and regression test management.

Index Terms: Simulation-in-the-Loop (SiL), SAE Level 2+, ADAS Validation, Scenario-based Testing, Safety Verification

I. INTRODUCTION

The automotive industry is rapidly evolving toward higher levels of vehicle automation, with SAE Level 2+ Advanced Driver Assistance Systems (ADAS) becoming increasingly prevalent in modern passenger vehicles. These systems inte- grate functionalities such as adaptive cruise control, lane cen- tering assist, and automatic emergency braking to deliver both longitudinal and lateral vehicle control under defined operating conditions. While these features enhance driving comfort and safety, their real-world deployment introduces challenges due to environmental variability, behavioral unpredictability, and system complexity—making them inherently safety-critical.

Validating SAE Level 2+ systems presents a multifaceted challenge. Traditional road and vehicle-level testing methods, while still valuable, are cost-prohibitive, time-intensive, and often insufficient for identifying rare but critical edge-case behaviors. The limited reproducibility of physical test environ- ments, combined with logistical constraints such as track avail- ability and safety restrictions, further hinder comprehensive coverage. As a result, traditional testing alone cannot fulfill the functional safety and performance validation requirements mandated by standards such as ISO 26262 [4].

To address these limitations, industry practitioners are in- creasingly leveraging Simulation-in-the-Loop (SiL) environ- ments, which allow ADAS software to interact with high- fidelity virtual models of the vehicle, surrounding traffic, infrastructure, and environmental conditions. A core enabler of scenario-based validation in SiL is the adoption of the NHTSA-defined Operational Design Domain (ODD) taxon- omy, which provides a structured classification of the system’s operating boundaries—including roadway types, environmen- tal conditions, objects, connectivity, zones, and operational constraints.

By mapping test scenarios to these six ODD dimensions, SiL enables scalable, repeatable, and traceable validation of ADAS behavior across Normal, Stress, Edge, and Out-of-ODD conditions. This paper presents a comprehensive SiL validation strategy for SAE Level 2+ systems, detailing scenario model- ing, combinatorial coverage techniques, performance metrics, and CI/CD-ready automation frameworks—all aligned with ISO 26262 safety life-cycle requirements.

II. Simulation-in-the-Loop Architecture

A. Overview of SiL Systems

Simulation-in-the-Loop (SiL) serves as a critical layer in the V-Model of system development and validation as explained in [2], bridging the gap between model-in-the-loop (MiL) design and hardware-in-the-loop (HiL) or road-level valida- tion. SiL enables the integration of actual production-intent software into a simulated environment that emulates the full stack—from sensors and vehicle dynamics to external traffic actors. This facilitates repeatable and automated validation of ADAS features under a wide range of conditions.

In the context of the V-model (Figure 1), SiL plays a role in both early verification (left side) and late validation (right side), supporting iterative testing before deploying code on embedded hardware. This is especially valuable for SAE Level 2+ features, where the software behavior must be validated against specific scenarios defined by the Operating Design Domain (ODD)—a formal specification that outlines the environmental and operational boundaries where a feature is designed to function safely. For example, an adaptive cruise control system may be valid only on highways under clear weather and daylight conditions.

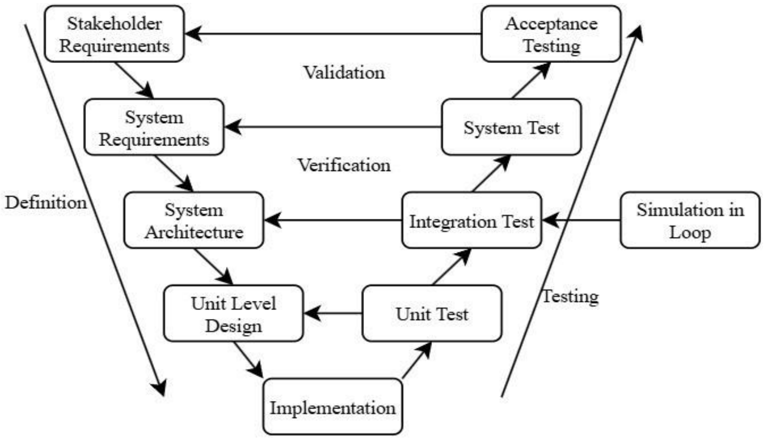

Fig. 1. Automotive SW V-Model with Simulation in Loop

B. Simulation in Loop (SIL) Architecture

The Simulation-in-the-Loop (SiL) architecture for ADAS validation is a modular, closed-loop system that replicates real- world vehicle behavior to test and verify software functionality under diverse driving scenarios. At its core is the Scenario Database, which provides parameterized test cases classified into four categories: Normal Case, Stress Case, Edge Case (as described in [8]), and Out-of-ODD (Operational Design Domain) Case. This classification enables comprehensive eval- uation of ADAS software behavior, both within and beyond the defined operational boundaries.

These scenarios feed into the Environment Simulation, which renders road geometries, dynamic actors (e.g., vehicles, pedestrians), and ambient conditions such as weather, time of day, and visibility. The Vehicle Dynamics Model simulates the physical response of the vehicle to control inputs, while the Vehicle Bus Simulation emulates communication interfaces (e.g., CAN, FlexRay, or Ethernet), ensuring realistic data flow among subsystems.

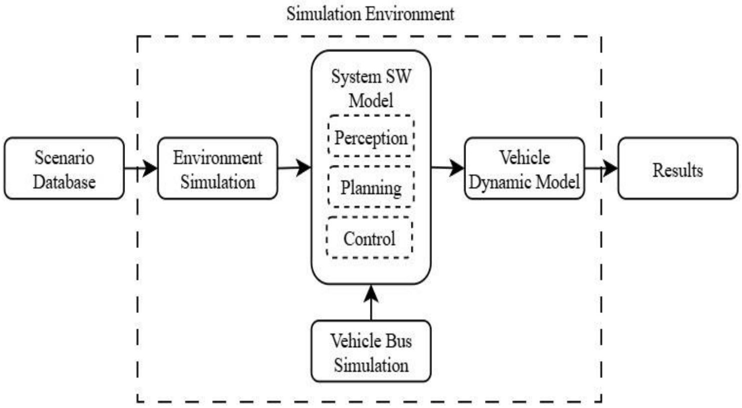

Fig. 2. SiL Architecture for ADAS Validation

At the heart of the simulation environment is the System Software (SW) Model (Figure 2), which is the ADAS software under test. It includes three critical modules: Perception, which interprets virtual sensor data; Planning, which computes driving strategies; and Control, which generates actuator-level commands. The system is validated against the boundaries defined by its ODD—for example, highways during daylight and clear weather—ensuring correct operation in valid con- ditions and graceful degradation or fallback behavior when encountering an Out-of-ODD scenarios. A Results and Met- rics Module continuously logs performance indicators such as time-to-collision, braking response time, and lane-keeping accuracy for detailed analysis.

III. Scenario Modeling and Automation

A. *Scenario Taxonomy and Templates*

In alignment with the NHTSA’s framework for testable cases and scenarios, scenario modeling in Simulation-in-the- Loop (SiL) systems should begin with a structured breakdown of the Operational Design Domain (ODD). The ODD defines the specific environmental, roadway, traffic, and temporal conditions under which an SAE Level 2+ feature is expected to operate safely. According to NHTSA report [1], ODD attributes can be grouped into six main categories:

1. Physical infrastructure: Roadway types, roadway sur- faces, roadway edges, roadway geometry
2. Operational Constraints: Speed limit, traffic conditions
3. Objects: Signage, roadway users (pedestrian, bicycle, motorcycle, car etc.), non-roadway user obstacles/ob- jects
4. Connectivity: Vehicles, traffic density info, remote fleet management system, infrastructure sensors and commu- nication
5. Environmental Conditions: Weather, weather-induced roadway conditions, particulate matter, illumination
6. Zones: Geo-fencing, traffic management zone, school/- construction zones, regions/states, interference zones

To build meaningful and testable scenarios, each scenario should be decomposed along the ODD dimensions. For exam- ple, a highway assist feature might specify: multilane divided highway (roadway), daylight and clear weather (environment), and speeds ranging from 60–120 km/h (operational constraint). These parameters serve as scenario boundary conditions and help filter relevant test cases from the scenario library.

Scenario automation is achieved by encoding ODD elements into machine-readable scenario templates using standards such as OpenSCENARIO [5] or frameworks like CommonRoad. Simulation platforms such as CARLA [6] also support pro- grammatic scenario definition through APIs, enabling fine- grained control over road geometry, dynamic actor behavior, environmental conditions, and sensor configurations. A com- binatorial expansion engine can then systematically generate scenario variations across the ODD matrix—for example, by modifying road curvature, visibility conditions, traffic density, or object types. Each generated scenario can be automatically tagged based on its compliance with the feature’s declared ODD—classified as Normal, Stress, Edge, or Out-of-ODD—to support coverage tracking, risk prioritization, and regression planning within the Simulation-in-the-Loop (SiL) validation workflow.

This ODD-aware approach supports traceability, risk-based scenario prioritization, and alignment with Object and Event Detection and Response (OEDR) tasks. It ensures SiL tests are not only comprehensive but also consistent with regulatory expectations from NHTSA report [1]. Ultimately, ODD-based scenario modeling enables scalable, automated validation that mirrors the operational reality of SAE Level 2+ systems.

B. Scenario Combinations

One of the key challenges in scenario-based SiL validation for SAE Level 2+ systems lies in managing the combinatorial explosion of test conditions. The full Cartesian product of ODD parameters (e.g., Road Type × Weather × Traffic) often results in hundreds or thousands of possible test scenarios. To address this, we apply a combination of combinatorial testing (pairwise or n-wise) and equivalence partitioning.

Pairwise (2-wise) testing ensures every possible pair of input parameters is covered at least once, capturing interaction faults between two variables. n-wise testing extends this to three or more variables. To reduce redundant testing, we apply equivalence partitioning, which groups values with similar expected outcomes into classes. For instance, fog and rain may be grouped into a “reduced visibility” class if system behavior is functionally similar. Figure 3 shows a sample scenario generation pipeline that can be used for generating the scenarios.

To further illustrate the practical application of scenario generation for virtual validation, the following Python imple- mentation provides a clear example:

```
from itertools import combinations, product
# Define equivalence-partitioned values for each NHTSA ODD category
ODD = {
"PhysicalInfrastructure": [
"UrbanRoad", "DividedHighway", "Roundabout" # Roadway types
],
"OperationalConstraints": [
"LowSpeed", "HighSpeed", "HeavyTraffic"  # Speed limit, traffic conditions
],
"Objects": [
"Pedestrian", "Car", "Debris"  # Roadway user + non-roadway user obstacle
],
"Connectivity": [
"V2IAvailable", "NoConnectivity"  # Infrastructure sensor access
],
"EnvironmentalConditions": [ "ClearDay", "RainyLowFriction",
"LowVisibility"  # Partition: weather, road condition, lighting
],
"Zones": [
"SchoolZone", "ConstructionZone", "NoZone" # Geo-fenced, managed, or neutral areas
]
}
# Generate pairwise scenarios from combinations of ODD parameter pairs
def generate_pairwise_scenarios(odd_params):
keys = list(odd_params.keys()) scenario_list = []
for param_pair in combinations(keys, 2):  # All pairwise combinations of 6 categories
param1, param2 = param_pair for value_combo in
product(odd_params[param1], odd_params[param2]): scenario = {
param1: value_combo[0], param2: value_combo[1]
}
scenario_list.append(scenario) return scenario_list
# Generate scenarios
scenarios = generate_pairwise_scenarios(ODD)
```

C. *Example of Scenario Modeling for a Highway Assist Fea- ture*

Highway Assist (HA) is a SAE Level 2+ feature that combines Adaptive Cruise Control (ACC) and Lane Centering Assist (LCA) to offer semi-automated longitudinal and lateral control on limited-access highways. The HA system is de- signed to operate within a defined Operational Design Domain (ODD)—typically including highway environments with well- marked lanes, moderate curvature, and clear weather.

In a SiL setup, HA validation involves injecting controlled scenarios via the simulation environment to test how well the feature performs in maintaining lane discipline, handling merges, negotiating curves, and adapting to slow or stopped traffic. Simulation allows us to isolate specific stress conditions and corner cases without endangering real vehicles.

Consider a Highway assist feature is designed to operate only under following conditions

TABLE I- ODD EXAMPLE FOR HA FEATURE

ODD Category	**Allowed Conditions for Normal HA Op- eration**
Physical Infrastructure	Divided Highway
Operational Constraints	High Speed (≥ 60 km/h), Moderate Traffic
Objects	Cars, Motorcycles (no sudden entries like pedestrians or stationary objects)
Connectivity	Optional, but must handle both V2I Avail- able and No Connectivity
Environmental Conditions	Clear Day, Dry Surface, Daylight (no fog/rain/snow)
Zones	No Zone (no construction or school zones)

Based on the classification rules in Table I, the following scenario classification labels can be used to generate the scenarios for the highway assist feature.

Here is a sample output Table III and a visual pairwise coverage matrix generated using the NHTSA ODD-based pairwise combinatorics logic. This illustrates how various pa- rameter interactions are covered while avoiding full Cartesian expansion.

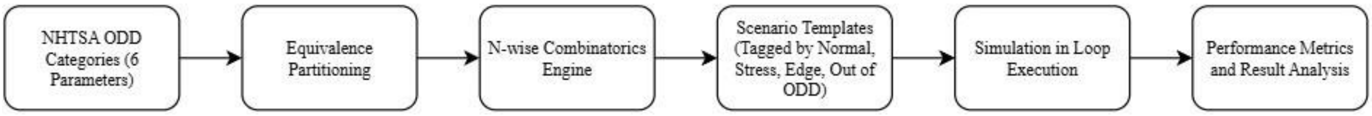
NHTSA ODD Categories (6 Parameters)
Equivalence Partitioning
N-wise Combinatorics Engine
Scenario Templates (Tagged by Normal, Stress, Edge, Out of ODD)
Simulation in Loop Execution
Performance Metrics and Result Analysis

Fig. 3. Scenario Generation Pipeline

TABLE II- SCENARIO CLASSIFICATION

Label	Condition
Normal	All ODD parameters fall strictly within the defined boundaries above.
Stress	One parameter is at the boundary limit (e.g., moderate traffic approaching dense, slight curve).
Edge	At least one parameter is outside the ODD but recoverable (e.g., Low Visibility, unexpected pedestrian).
Out-of-ODD	Any critical parameter completely violates the feature’s ODD (e.g., Roundabout, Construction Zone).

TABLE III- SCENARIO GENERATED FOR SIL

Scenario ID	Scenario	Classification
HA001	Divided Highway, High Speed, Car, No Connectivity, Clear Day, No Zone	Normal
HA002	Divided Highway, High Speed, Pedestrian, No Connectivity, ClearDay, No Zone	Edge
HA003	Divided Highway, High Speed, Car, No Connectivity, Low Visibility, School Zone	Out-of-ODD
HA004	Divided Highway, Heavy Traffic, Car, V2I Available, Clear Day, No Zone	Stress
HA005	Roundabout, High Speed, Car, V2I Available, Clear Day, No Zone	Out-of-ODD

IV. ODD COVERAGE, TEST PROGRESSION & REGRESSION IN SIL FOR AN EXAMPLE HIGHWAY ASSIST FEATURES

As SAE Level 2+ features like Highway Assist mature through iterative software development and calibration cycles, maintaining robust and systematic validation coverage within the defined Operational Design Domain (ODD) becomes es- sential. Simulation-in-the-Loop (SiL) testing provides a vir- tualized, scalable environment to evaluate system behavior by aligning scenario test cases with structured ODD dimen- sions—such as road geometry, speed constraints, visibility, traffic density, object types, and functional zones (e.g., school or construction areas).

Each SiL iteration (Figure 4) evaluates Highway Assist against a diverse scenario pool categorized into

Normal, Stress, Edge, and Out-of-ODD conditions. These scenarios are as- sessed using critical performance and safety KPIs, including Time-to-Collision (TTC), headway distance, steering torque, driver monitoring states, system up-time, ODD violation detec- tion, fallback activation time, and warning latency. This struc- tured, metrics-driven approach enables precise quantification of ODD coverage and system performance trends.

In agile development environments—especially those de- ploying Over-the-Air (OTA) updates—SiL enables fast- turnaround validation of perception, planning, or control changes. Regression testing is automated and version- controlled across software builds, ensuring that key safety functionalities are preserved, and no regressions are intro- duced. Test prioritization is guided by risk profiles, emphasiz- ing edge-case scenarios and interactions with vulnerable road users or degraded conditions.

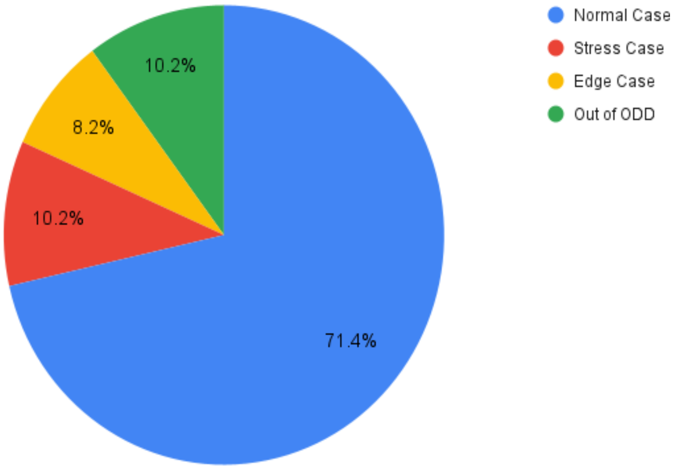

Fig. 4. HA Iteration 1 ODD Coverage

A centralized SiL dashboard aggregates results from each validation cycle (Figure 5), visualizing scenario coverage progression, pass/fail trends, and ODD boundary adherence. This dashboard also supports traceability to ISO 26262 safety goals by mapping each test case to specific ODD categories and corresponding functional safety requirements.

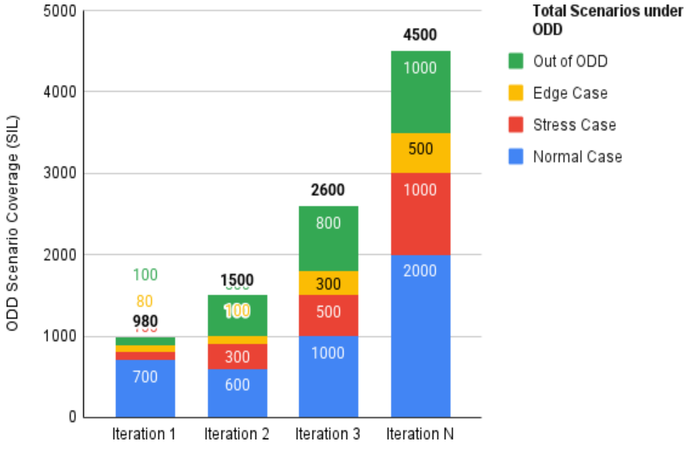

Fig. 5. HA ODD Coverage over various iterations

Expanded KPI sets—including steering oscillations, braking response curves, and system deactivation frequency—can be layered into the dashboard for fine-grained regression tracking. This enables cross-functional decision-making, allowing developers, safety engineers, and release managers to:

- Monitor feature evolution across builds
- Identify risk exposure through scenario-based metrics
- Validate OTA readiness with objective evidence

- Ensure alignment with declared ODD boundaries

Ultimately, this SiL-based framework empowers organi- zations to accelerate feature delivery without compromising safety, supporting scalable, high-confidence deployment of Highway Assist and other Level 2+ systems for virtual vali- dation [3] [7].

V. CONCLUSION

As the deployment of SAE Level 2+ driver assistance features accelerates across the automotive industry, the need for rigorous, scalable, and safety-oriented validation becomes increasingly critical. This paper has presented a comprehensive framework for Scenario-Based Validation using Simulation-in- the-Loop (SiL) systems as a central pillar for verifying func- tional behavior, safety performance, and ODD conformance of features such as Highway Assist, Adaptive Cruise Control, and Collision Mitigation.

Through detailed discussion of scenario modeling and au- tomation strategies, this paper demonstrated how leveraging the NHTSA-defined ODD taxonomy enables structured sce- nario generation across six critical domains—physical infras- tructure, operational constraints, objects, connectivity, environ- mental conditions, and zones. The integration of equivalence partitioning and pairwise/n-wise combinatorial allows for ef- ficient test space exploration, ensuring broad ODD coverage while minimizing redundancy.

Furthermore, this paper highlighted how SiL environments support iterative validation, enabling rapid assessment of software across multiple development stages. The use of performance KPIs such as lane deviation, steering torque, ODD violation detection, and fallback activation time ensures quantifiable safety verification. With each new iteration or OTA update, regression testing confirms system stability, while scenario dashboards and traceability matrices provide trans- parency into test coverage and risk-based prioritization.

Ultimately, Simulation-in-the-Loop serves not only as a validation mechanism but also as a powerful engineering decision-making tool, enabling early bug detection, accelerated development cycles, and safer, traceable software releases. As ADAS features grow in complexity and regulatory ex- pectations increase, SiL-based scenario validation will be indispensable in bridging the gap between simulation realism, safety assurance, and production readiness.

REFERENCES:

1. L. Staplin, T. Mastromatto, K. H. Lococo, K. W. Gish, and J. O. Brooks, *The effects of medical conditions on driving performance*, Report No. DOT HS 812 623, Washington, DC: National Highway Traffic Safety Administration, Sep. 2018.
2. P. Pathrose, *ADAS and Automated Driving: A Practical Approach to Verification and Validation*. SAE International, 2022.
3. X. Meng, H. Gan, and Y. Yan, “Simulation Test of Driving Assistance System Based on Virtual Scene,” *Journal of Physics: Conference Series*, vol. 1419, p. 012047, 2019, doi: 10.1088/1742-6596/1419/1/012047.
4. *ISO 26262: Road Vehicles – Functional Safety*, International Organiza- tion for Standardization, 2018.
5. *OpenSCENARIO Specification*, ASAM e.V. [Online]. Available: https:
6. //www.asam.net/standards/detail/openscenario/. Accessed: Jun. 2025.
7. A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,” in *Proc. Conf. Robot Learning (CoRL)*, 2017.
8. M. Nobis, A. Telle, J. Reiter, and H. Winner, “Simulation-based Vali- dation of Advanced Driver Assistance Systems,” SAE Technical Paper 2019-01-0147, 2019.
9. A. G. Dawood, A. Schneider, and J. Rauh, “Edge Case Generation for Testing Automated Driving Systems,” in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, 2020.