International Alliance for Mobility Testing and Standardization™
Best Practice

IAMTS0001202104

Issued 2021-04-01

A Comprehensive Approach for the Validation of Virtual Testing Toolchains

Rationale

"Autonomous vehicles need to be driven more than 11 billion miles to be 20% better than humans. With a fleet of 100 vehicles, 24 hours a day, 365 days a year, at 25 miles per hour, this would take 518 years—about a half a millennium." [43]

This quote shows the challenges very well. Due to the ever-shorter development cycles, new methods are required. Otherwise, the large number of scenarios/miles can no longer be completed in the real world, making it impossible to release ADAS/AD systems during this short development cycle. In this article, a comprehensive approach for validating simulation is explained. Making simulation accessible, for the validation/homologation of ADAS/AD systems, this large number of driving tasks can be handled.

This approach is based on four steps. In the first step, the models get validated on subsystem level. After that, the whole passive vehicle system gets validated. In the third step, the sensor system is under investigation; in the last step, the whole integrated system with all models/subsystems gets validated. This process is roughly shown on some examples in the appendix.

Preface

IAMTS is a global, membership-based association of organizations that are stakeholders in the testing, standardization, and certification of advanced mobility systems and services. IAMTS brings together testing consumers and providers at a global scale to help develop a commonly accepted framework of test scenarios, validation and certification methods, and terminology.

Our mission is to develop and grow an international portfolio of advanced mobility testbeds that meet the highest quality implementation and operational standards.

Our vision is to create a global community of advanced mobility testing service providers with companies, organizations, and agencies in need of such services; to learn, develop, and share best practices to ensure consistent, replicable, and reliable testing; to maintain a global directory of physical, virtual, and cyber-physical testbeds and support and promote their audited capabilities; and to promote the rapid evolution of standards and certifications to ensure the safe deployment of advanced mobility systems and services.

Introduction

A major challenge to market introduction of automated driving systems is validation. This stems from the overall system increasing in complexity as interaction between vehicle, humans (drivers, passengers, and those external to the vehicle), and environment becomes more intertwined. What is more, responsibility for safe operation once assumed by the driver, including obeying traffic rules and managing complex and critical situations, shifts to the manufacturer. Understanding and accounting for regionally specific differences in regulations, human behavior, and the traffic environment is therefore more important, and this means the variants are increasing tremendously. “Thus, alternate methods of validation are required, potentially including approaches such as simulation.” [1] “The approach of ESP homologation has demonstrated that virtual homologation could be a future method for application on ADAS and AD functions. This could be achieved by combining physical tests with more extensive simulation and ensuring senor quality by component homologation.” [2] The more complex the systems/the ODD, the more relevant simulation becomes. What is more, no one simulation tool can be used to test all aspects of the ADAS/AD system. Each has its strengths and area of focus. Manufacturers therefore use multiple simulation tools in pursuit of validating the safety of the full system.

Conventional on-road and proving ground testing are insufficient to fulfill these complex requirements and assure meaningful coverage of test cases and scenarios. To fill this gap, the use of virtual testing needs to shift beyond pure development in the direction of approval and homologation. However, the biggest challenge to incorporating virtual testing in this manner is acceptance of its toolchain and modeling. Validating simulation aims to demonstrate that simulations are accurate enough to fulfill their intended purpose.

The use and application of virtual testing in automated driving has been researched and discussed at length [3, 4, 5, 6, 7, 8, 9, 10]. An industry-led working group of IAMTS is tackling next steps to assist with the design and evaluation of simulations as an acceptable alternative to some physical testing. As a beginning of that effort, this paper outlines an approach to validate simulation tools, toolchains, and models. It combines established methods, standards, and regulations with new approaches to define an overall process that is sustainable, traceable, and efficient. In addition, it considers global and cross-regional applications.

Further efforts by this working group intend to build upon and further define this approach into a process that can be considered and applied by testing providers, manufacturers, certification bodies, and regulators. The ultimate goal of this effort is to overlay a process upon physically tested scenarios to determine whether it makes sense to perform that test in a specific virtual environment. Alternatively, it can be used to craft a virtual testing solution to satisfy that objective.

Table of Contents

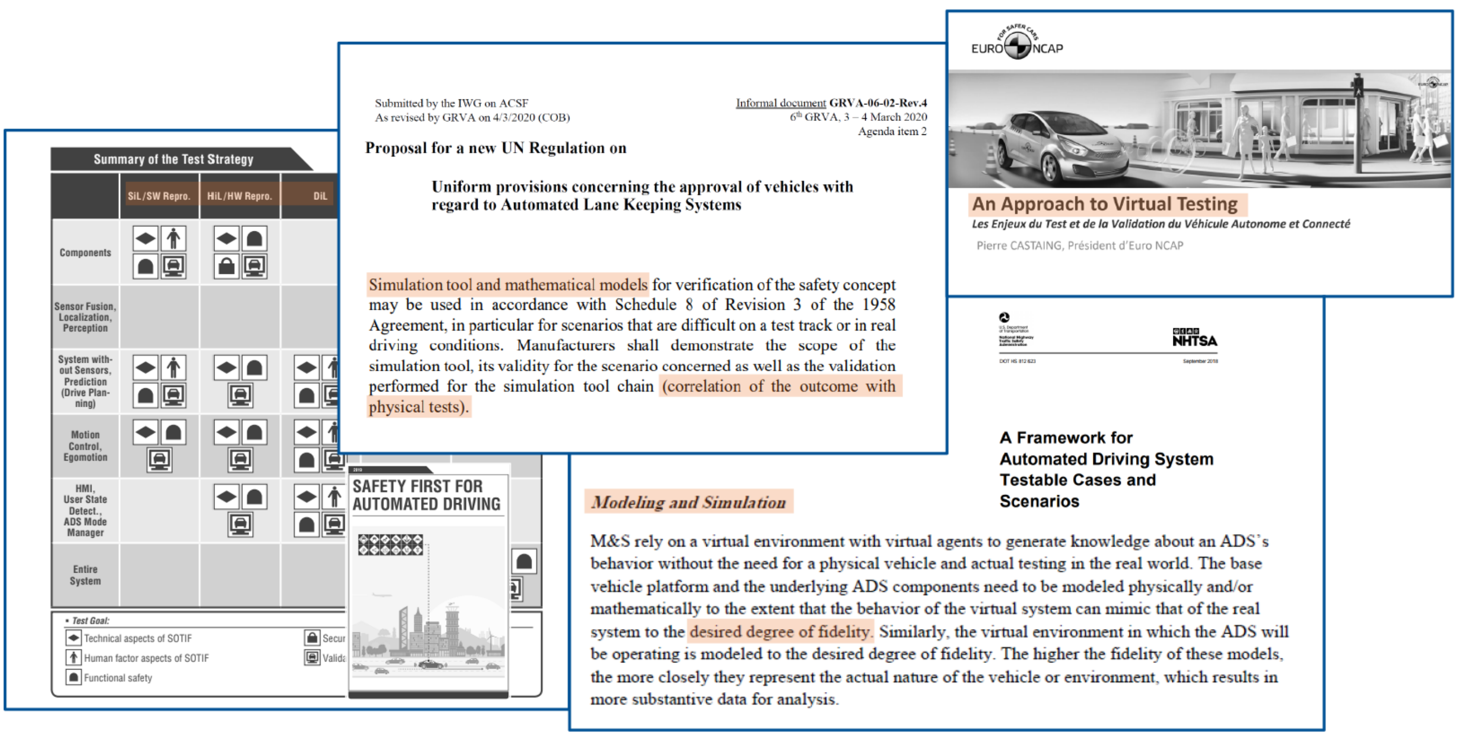
1. Challenges

The following elements drive the need for increased virtual testing in ADAS[1]/AD[2] development and approval:

- Test coverage: Covering each necessary scenario variation on real roads and proving grounds is impossible from perspectives of time, cost, and availability.
- Concept studies: Implementing each concept (e.g., sensor setup) in a physical environment is inefficient and must be done in an early stage of the development.
- Variant coverage: Covering each vehicle variant of a manufacturer for testing on real roads and proving grounds is impossible.
- Frontloading: Continuous validation along the development process is crucial, even before real prototypes are available.
- Calibration: Testing ODDs[3] may require different variations and calibrations, which can be a difficult or impossible task in a physical environment.
- Safe testing: Corner cases and critical scenarios are key elements in ADAS/AD validation. Not all of them can or should be performed in the real world.

Additional information and further background can be found in [11, 12, 13, 14].

FIGURE 1 Importance of Simulation in Standards and Regulations [5, 7, 8, 15]



---

1 Advanced driver assistance systems.
2 Automated driving.
3 Operational design domain.

Using virtual testing for ADAS/AD validation has many challenges. First, proof must be provided that the simulation toolchain correlates acceptably with reality. This is largely a question of desired accuracy. Is only system-level correlation required, or is a higher level of fidelity needed to test some subsystems? Different use cases are likely to require different levels of accuracy from the corresponding models. This point is important, because it raises the question of where the trade-off is between a simulation with a lot of modeling effort (higher costs upfront, cheaper execution), a purely physical test (all real components), and a cyber-physical test [16] with more real components (lower costs upfront, more expensive execution).

Second, virtual testing often entails using a combination of different tools that each target specific functions and components. These tools should be managed within an overall testing architecture that is traceable and efficient. It also must be flexible enough to accommodate significant changes. During the lifespan of the testing architecture, there will be adaptations to simulation models, requirements, tool versions, and even toolset combinations. Currently, there is no established validation process for virtual testing using multiple toolsets. Validation processes are available on a component level for single models (subsystems), so an all-encompassing process for highly integrated systems can be developed based on these. Still, there is a challenge tracing requirements, results, deviations, and anomalies centrally across such a system. This can be managed by bringing all the sub-models together into a centralized model and all parameters, results, etc., into a centralized database.

Third, one should expect and get the same outcome from a physical or virtual test (or combination thereof). This means establishing consistent KPIs[4] across environments using the real-world as the benchmark and extending them to pure simulation. Foremost, this ensures correlation between the two environments and requires testing the same scenarios across the spectrum of abstraction layers:

- Model-in-the-Loop (MiL)
- Software-in-the-Loop (SiL)
- Hardware-in-the-Loop (HiL)
- Driver-in the-Loop (DiL)[5]
- Vehicle-in-the-Loop (ViL)

Although this part of the validation process can seem cumbersome, it is essential to establishing confidence in the fidelity of the simulation. The result is that it facilitates the testing of variants in simulation that would be impractical in a physical environment [17], either because of the nature or sheer number of variations.

2. A Comprehensive Approach to Validating Simulation

For virtual testing methods to be used for ADS[6] validation, the toolchain itself must be verified and validated by establishing an acceptable correlation between virtual and physical testing.

The first phase of this process is preparation and involves, among other things, (1) selecting the relevant aspects for modeling, and (2) defining fidelity levels.

The second phase is execution, which includes defining a reasonable tolerance for the KPIs [18] (see Figure 2).

[4] Key performance indicator.
[5] DiL is commonly used, but Human-in-the-Loop would be better, as in case of AD no driver is required.
[6] Automated driving system.

FIGURE 2 Verifying and Validating the Virtual Testing Toolchain

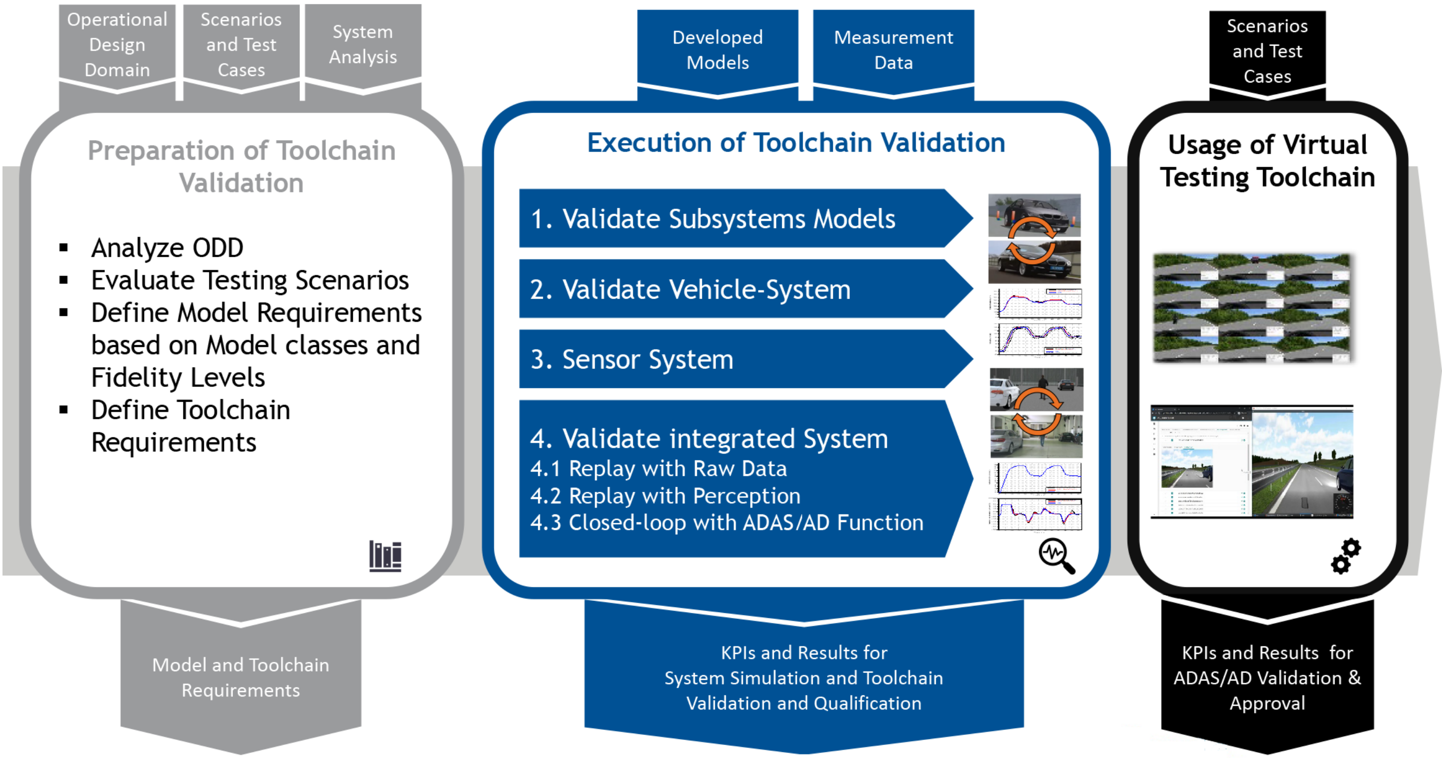
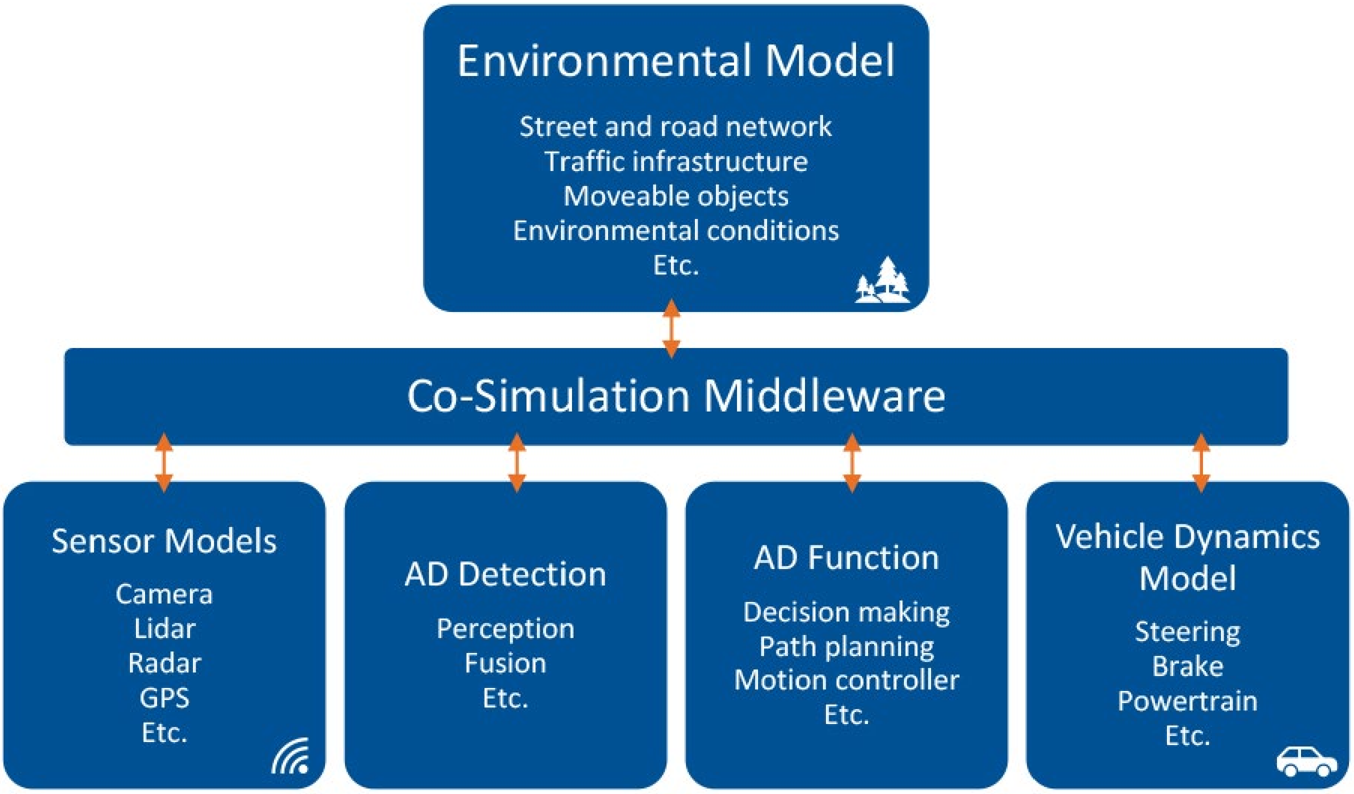


Figure 3 illustrates a basic simulation architecture. It consists of:

- Environmental model: Represents the street/road network, the traffic infrastructure, moveable objects, and environmental conditions with its interaction to other components (e.g., rain makes the road wet, which results in a different friction coefficient).
- Sensor models: Used to simulate the different sensor types (e.g., radar, LiDAR, camera, positioning).
- AD detection: This is the first part of the ADAS/AD system which should be tested. It consists of the perception and fusion algorithms. Input are the sensor raw data, output is the fused object list
- AD function: This is the second part of the ADAS/AD system. It consists out of the decision making, path planning, and motion controller.
- Vehicle dynamics model: Represents the virtual vehicle with all real components (e.g., steering, brake, powertrain, etc.).

FIGURE 3 Simulation Architecture



For each component/model, different fidelity levels are required. An approach to defining fidelity levels commonly used in vehicle dynamics [19] can likewise apply here to environmental and sensor models. Examples for classifying models under such a schema can be found in Appendix B.

Very importantly, there is also a co-simulation middleware layer which serves as a "backbone" connecting the different tools and models in a common interface. It handles issues of synchronization encountered with different step sizes, links solvers to tools, and couples simulated models with physical hardware like control units. A (tool independent) co-simulation middleware approach is preferable to cover these challenges. Although an alternative to a multiple toolchain would be to use a single tool which provides all kinds of models using one single solver, it is very unlikely to find as successful given that separate tools each provide their strengths when combined together in the area of ADS simulation.

2.1 Preparation of the Validation Process

In the preparation phase, all boundary conditions are defined. The ODD is analyzed and described to derive the requirements for the models. The breadth complexity and level of detail vary depending on the relevance, significance, and range of each factor. For example, if the ODD is limited to urban scenarios, vehicle speed is limited. This means that the vehicle dynamics model need not be validated for the whole range, but only up to a certain speed. Or, if the ODD excludes night operation, the sensor models do not need to be validated against low-light conditions. There are also often regional influences to consider. Using the same ODD, different operating speeds in Europe, the U.S., or China may require a different speed-range for validation of the vehicle dynamics model. As one can quickly see, the ODD is as important for the validating the toolchain as it is for the ADS.

Referring to the UNECE ALKS regulation proposal [4], the following list shows the relevance of the different components of the simulation models:

- Steering model (lane keeping and path driving)
- Brake model (including high dynamics; e.g., for emergency braking)
- ABS controller (in case of dynamic braking)
- Suspension model (movement of chassis and influences on field of view of sensors, special use case truck with trailer)
- Powertrain model and powertrain management (acceleration capabilities; e.g., for overtaking or highway entrance)
- Virtual lanes (relevant properties, reflections, interruptions, color)
- Sensor models (e.g., camera model with relevant effects which can be also simulated in the virtual environment)

To establish a baseline for the ODD, physical reference tests within the desired ODD are performed on a proving ground, real traffic environment, or both. These tests are performed with the desired active automated driving function enabled and—of course—a safety driver. It is during these reference tests that important effects and phenomena are often discovered.

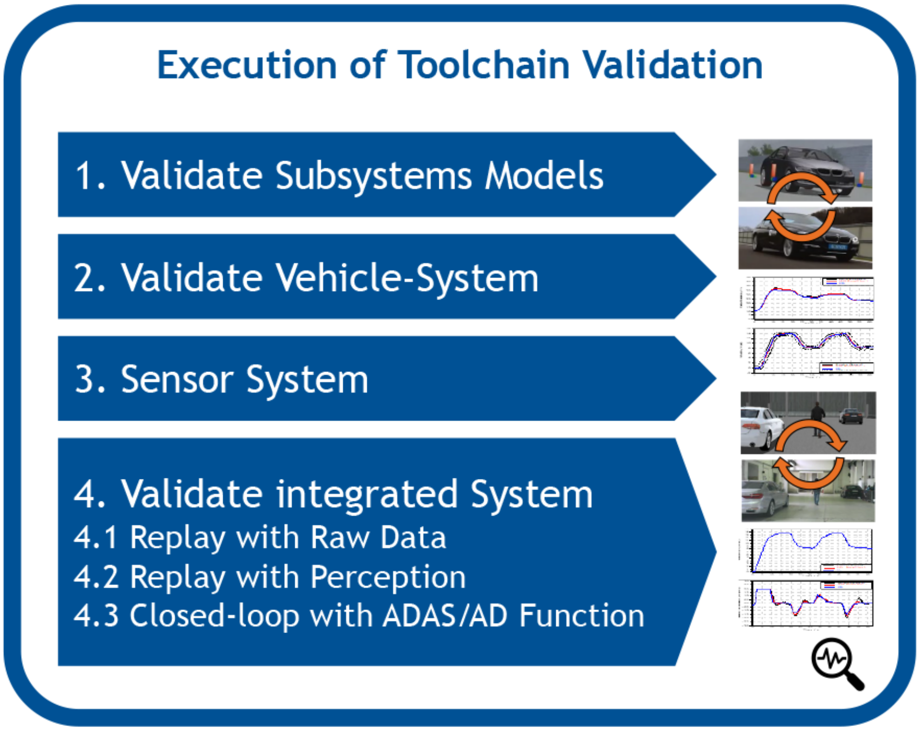
At minimum, the following data should be collected for validating the integrated system (see 2.2.4) and deriving the necessary level of fidelity for modeled components and functions:

- Reference sensor measurement system (higher accuracy than vehicle sensors)
- Vehicle data
- Sensor raw data
- Object lists (output of perception)
- Set values of the vehicle (acceleration pedal, brake pedal, steering or similar like accelerations)

2.2 Execution of the Validation Process

The next phase is execution of toolchain validation, which consists of four main steps. These are summarized below and addressed in more detail subsequently (see Figure 4).

FIGURE 4 Schema of the Execution Process



2.2.1 Step 1: Validation of the Subsystem Models

At a high level, subsystem models are the vehicle dynamics models, the environmental models, and the sensor models. The baselines for validation are the model classes and fidelity levels selected in the preparation phase. These describe the requirements and the effects the models must consider. These subsystems for the vehicle dynamics models are defined based on [19] for the environmental models based on the PEGASUS six-layer approach [20]. Validation of subsystems is typically performed by experts using open loop input-and-output tests. Appendix B provides some examples for specific use cases.

2.2.2 Step 2: Validation of the Vehicle System (Passive)

In the first integration step, the virtual vehicle is embedded in an environmental model. The fidelity of the environmental model is related to vehicle dynamics and mainly consists of a 3D road with road properties like friction, etc.

The validation tests focus on lateral and longitudinal vehicle dynamics and are also mainly open-loop:

- Longitudinal maneuver, e.g., maximum vehicle body pitch angle, braking with various pedal positions, coast down measurement.
- Lateral maneuver, e.g., maximum vehicle body roll angle, slowly increased steer maneuver at different vehicle speeds, slow weave steering maneuver at different vehicle speeds, step steer test, sine sweep test method, weave test method, transition test method.

An example is shown in B.3 of Appendix B.

2.2.3 Step 3: Validation of the Sensor System

In the third integration step, the sensor models are integrated with the relevant environmental model. The environmental model should be able to accommodate specific properties of the sensor model; e.g., for HDR processing, the simulator needs to run at a higher refresh rate. The validation tests are also mainly open-loop. Examples for are listed in B.4 of Appendix B.

2.2.4 Step 4: Validation of the Integrated System

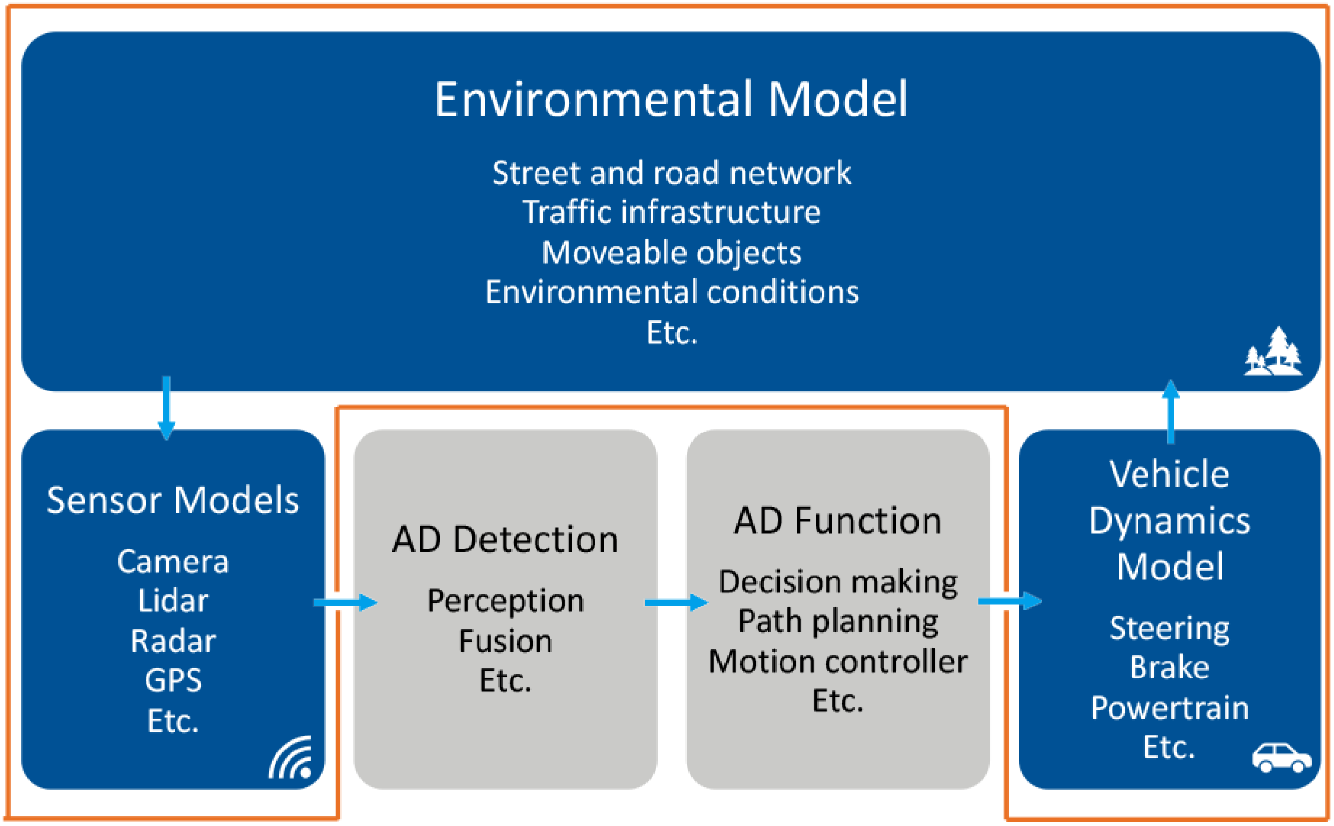
The final validation step is multi-part and combines the subsystems and sensor models with perception and function. Complexity should be introduced step-by-step in order to identify the cause of any deviations and anomalies.

The defined reference route(s) are designed in the simulation.

2.2.4.1 Step 4.1: Replay with Raw Data

Step 4.1 is used to determine correlation between the real-world raw sensor data and the simulation calculated sensor data. The virtual vehicle is not being driven by the autonomous driving function, but with the set values recorded by the real vehicle.

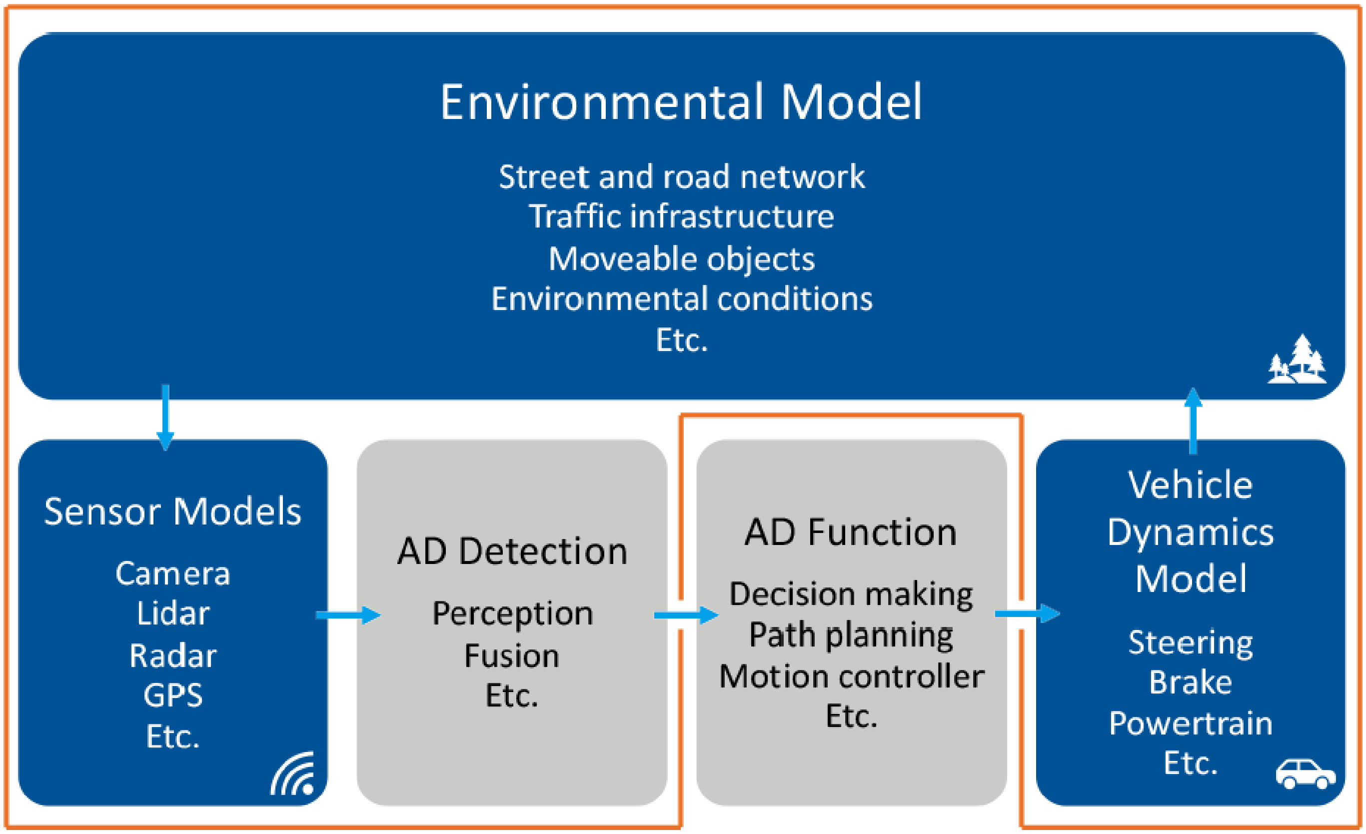
FIGURE 5 Replay for Evaluation of Sensor Raw-Data Output



Examples for effects and behavior analyzed/correlated in this step:

- Positioning of sensors (sensor setup)
- Chassis movements with sensors
- Geometrical effects in system (reduced field of view based on chassis movements; e.g., truck and trailer)
- System and simulation performance, including co-simulation of all parts (models, tools, numerical topics)
- Interface checking in virtual vehicle

FIGURE 6 Replay for Evaluation of Perception



2.2.4.2 Step 4.2: Replay with Simulated Perception

Step 4.2 determines the correlation between the perception of elements in the physical versus simulated environments. This step is run against the same scenario and holds constant in all respects the variables used in 4.1, except that it introduces into the simulation the same perception algorithm used in the preparation phase by the real-world vehicle. Steps 4.1 and 4.2 can be combined, as the physical test is the same (only the recorded data is different). The following are some examples of elements against which to measure and to observe for correlation and anomalous effects and behavior:

- Bounding boxes within FOV[7] from simulator, compare to bounding box output of perception stack.
- Classification of objects within field of view against simulation output of objection in FOV.
- Camera: Classification and localization of raw imagery, compare to same analysis of simulated data. Use same algorithm and compare outputs.
- Lidar: Object list, confusion matrix.
- Radar: Object list, confusion matrix.
- Sensor fusion object list. Compare against ground truth, confusion matrix.

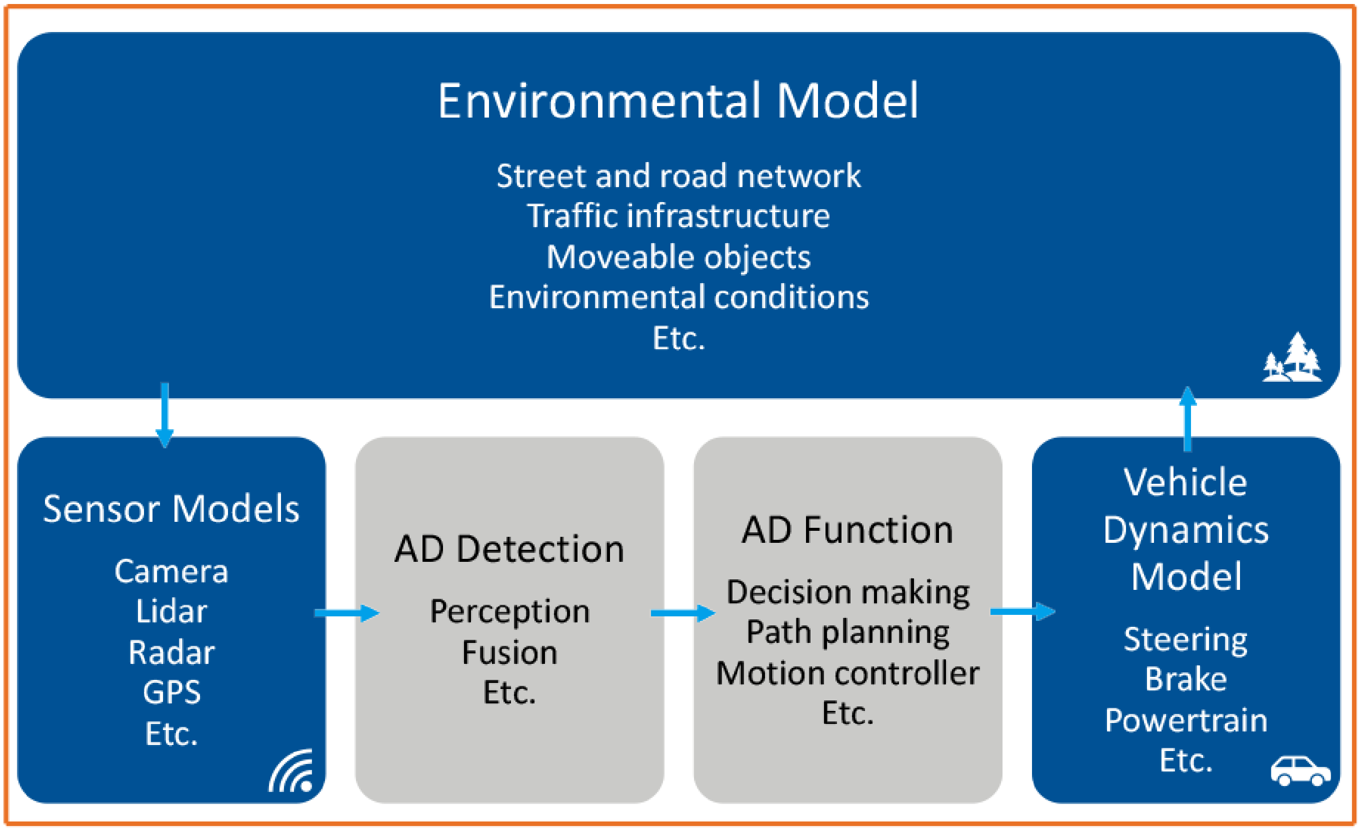
2.2.4.3 Step 4.3: Replay with Automated Driving Function

The final validation step evaluates the behavior of the vehicle in real driving versus simulation. This step is executed as in 4.2, but with the automated driving function of the simulation switched on. Beyond others, the control outputs (steering, throttle, brake) of the ADS function are measured and observed for correlation as well as the reaction of the vehicle; e.g., verifying if the track of the virtual vehicle matches to the recorded track.

---

[7] Field of view.

FIGURE 7 Setup for Evaluation of the Whole Toolchain



3. Discussion

As noted here, the use of multiple tools within a virtual testing architecture is very common and often necessary. Because a validation approach may vary depending upon the specific tools and even manufacturer-provided models used therein, it is our view that any ensuing regulations defining the use of simulation in testing automated systems should not be too prescriptive, but rather (a) set broader criteria for correlating with physical testing, and (b) permit in an individualized review of the specific approach taken. In general, such oversight should focus on the system as a whole and not at the level of simulating subsystems/components, not only to promote a neutral stance on the tools and approaches used, but to accommodate instances of “black boxes” which one can encounter when trying to decompose a proprietary tool or model at such a level.

The aim of this paper is to outline a first draft of a virtual testing toolchain validation process from a subsystem level up to a fully integrated simulation. It combines different views from different regions, companies, and research organizations working in this field. The outlined process described herein has been derived from best practices of the contributors considering existing standards and publications. This step-by-step approach ensures that correlations are traceable and deviations easily identified and quantified. This first iteration is a practical theory based on the knowledge and experiences of the contributors. A proof of the overall process has yet to be performed and will be a topic for future work. In particular, environment model and sensor model classifications, fidelity levels, and validation tests must be developed further. In addition, specific validation tests and their associated KPIs have not been addressed in this paper.

4. What's Next

For next steps, IAMTS will apply this process to concrete scenarios and use cases. The first application will be in relation to Automated Lane Keeping Systems (ALKS), for which UNECE will bring a first regulation into force beginning of 2021 ("UN Regulation on uniform provisions concerning the approval of vehicles with regards to Automated Lane Keeping System"). Of particular interest to the working group in this context is validation of camera and lane models. Specific aspects of the process to be refined in this effort include:

- Defining fidelity levels for the models (vehicle dynamics, environmental models, sensor models) for an ALK use case.
- Describing model effects, accuracy, etc.
- Describing a parameterization process (real-world measurement, extracting parameters, etc.).
- Describing a correlation process (including identifying key tests for an ALKS, KPIs and variable ranges).

5. About the International Alliance for Mobility Testing and Standardization™

IAMTS Vision:
Our vision is to create a global community of advanced mobility testing service providers with companies, organizations, and agencies in need of such services; to learn, develop, and share best practices to ensure consistent, replicable, and reliable testing; to maintain a global directory of physical, virtual, and cyber-physical testbeds and support and promote their audited capabilities; and to promote the rapid evolution of standards and certifications to ensure the safe deployment of advanced mobility systems and services.

IAMTS Mission:
Our mission is to develop and grow an international portfolio of advanced mobility testbeds that meet the highest quality implementation and operational standards.

6. Contact Information

To learn more about the International Alliance for Mobility Testing and Standardization™, please visit http://iamts.org.

Contact: info@iamts.org

7. Contributors

AVL: Tobias Dueser, Christian Gutenkunst—Lead and Contribution

CATARC: Qidong Zhao, Bolin Zhou—Contribution

Metamoto: Mark Hary—Contribution

NVIDIA: Barnaby Simkin, Ashley Reid—Contribution

Retrospect: Michael Woon, Aaron Stachewicz—Contribution

SAE International: John Tintinalli—Contribution

Shanghai SH Intelligent Automotive Technology Co., Ltd.: Charlie Cheng—Contribution

3D Mapping Solutions: Gunnar Graefe, Sebastian Tuttas—Contribution

Appendix A. Citations and Further Reading

[1] P. Koopman and M. Wagner, "Challenges in Autonomous Vehicle Testing and Validation," in 2016 SAE World Congress, Detroit, 2016.

[2] A. Kraus, H. Abdellatif, B. Schick, and S. Riedmaier, "Perspectives for simulation aided Homologation of highly automated vehicles," in 7th International Symposium on Development Methodology, Wiesbaden, 2017.

[3] OICA, "Future Certification of Automated Driving Systems," 01 02 2019. [Online]. Available: https://www.unece.org/fileadmin/DAM/trans/doc/2019/wp29grva/GRVA-02-27e.pdf. [Accessed 08 05 2020].

[4] UNECE, "World Forum for the harmonization of vehicle regulations (WP.29)," [Online]. Available: https://www.unece.org/trans/main/welcwp29.html. [Accessed 08 05 2020].

[5] UNECE, "Validation Method for Automated Driving (VMAD)," 20 08 2019. [Online]. Available: https://wiki.unece.org/pages/viewpage.action?pageId=60361611. [Accessed 08 05 2020].

[6] VVMethoden, "Research Project VVMethoden," 01 07 2019. [Online]. Available: https://vvm-projekt.de/. [Accessed 19 10 2020].

[7] M. Wood, M. Robbel, M. Maass, R. D. Tebbens, M. Neijs, M. Harb, J. Reach, K. Robinson, D. Wittmann, T. Srivastava, M. E. Bouzouraa, S. Liu, Y. Wang, C. Knobel, D. Boymanns, M. Löhning, B. Dehlink, D. Kaule, R. Krüger, J. Frtunikj, F. Raisch, M. Gruber, J. Steck, J. Mejia-Hernandez, S. Syguda, P. Blüher, K. Klonecki, P. Schnarz, T. Wiltschko, S. Pukallus, K. Sedlaczek, N. Garbacik, D. Smerza, D. Li, A. Timmons, M. Bellotti, M. O'Brien, M. Schöllhorn, U. Dannebaum, J. Weast, A. Tatourian, B. Dornieden, P. Schnetter, P. Themann, T. Weidner and P. Schlicht, "Safety First for Automated Driving," 02 07 2019. [Online]. Available: https://www.daimler.com/dokumente/innovation/sonstiges/safety-first-for-automated-driving.pdf. [Accessed 07 05 2020].

[8] NHTSA, "A Framework for Automated Driving System Testable Cases and Scenarios," 09 2018. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13882-automateddrivingsystems_092618_v1a_tag.pdf. [Accessed 08 05 2020].

[9] PEGASUS, "Research Project PEGASUS," 30 06 2019. [Online]. Available: https://www.pegasusprojekt.de/en/home. [Accessed 08 05 2020].

[10] ENABLE-S3, "Research Project ENABLE-S3," 05 2019. [Online]. Available: https://www.enable-s3.eu/. [Accessed 08 05 2020].

[11] T. Düser, H. Abdellatif, C. Gutenkunst and C. Gnand, "Approaches for the Homologation of Automated Driving," ATZ Electron Worldw (ATZelectronics worldwide), pp. 48-53, 04 10 2019.

[12] C. Gnand and T. Düser, "Homologation and Validation of Automated Driving Functions – It’s all About an Efficient Methodology and Process!," in International Symposium on Development Methodology, Wiesbaden, 2019.

[13] B. Koller and T. Düser, "Homologation und Validierung von automatisierten Fahrfunktionen," ATZ Extra, pp. 22-27, 12 03 2020.

[14] B. Simkin and T. Düser, "Vehicle-in-the-Loop and Virtual Testing," in NCAP VTA Workshop, Paris, 2020.

[15] P. Castaing, "An Approach to Virtual Testing," 03 2019. [Online]. Available: https://cdn.rohde-schwarz.com/fr/general_37/local_webpages/2019___automotive_tech_day_3_0/07_Euro_NCAP_Virtual_Testing.pdf. [Accessed 18 05 2020].

[16] DIN SAE SPEC 91381, Terms and Definitions Related to Testing of Automated Vehicle Technologies, 2019.

[17] S. Riedmaier, J. Nesensohn, C. Gutenkunst, T. Düser, B. Schick and H. Abdellatif, "Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions," in 11. Grazer Symposium Virtual Vehicle, Graz, 2018.

[18] European Parliament and the Council of the European Union, "Regulation (EU) 2018/858," 30 05 2018. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018R0858&from=EN. [Accessed 08 05 2020].

[19] ISO/DIS 11010-1, Passenger cars - Simulation model classification and taxonomy - Part 1: Vehicle dynamics.

[20] J. Bock, R. Krajewski, L. Eckstein, J. Klimke, J. Sauerbier and A. Zlocki, "Data Basis for Scenario-Based Validation of HAD on Highways," in 27th Aachen Colloquium Automobile and Engine Technology 2018 , Aachen, 2018.

[21] G. Herz, B. Schick, R. Hettel and H. Meinel, "Sophisticated Sensor Model Framework Providing Realistic Radar Sensor Behavior in Virtual Environments," in GSVF, Graz, 2017.

[22] ISO 17288-1, Passenger cars - Free-steer behaviour - Part 1: Steering-release open-loop test method.

[23] ISO 17288-2, Passenger cars - Free-steer behaviour - Part 2: Steering-pulse open-loop test method.

[24] ISO 21994, Passenger cars - Stopping distance at straight-line braking with ABS - Open-loop test method.

[25] ISO 14512, Passenger cars - Straight-ahead braking on surfaces with split coefficient of friction - Open-loop test procedure.

[26] ISO 9815, Road vehicles - Passenger-car and trailer combinations - Lateral stability test.

[27] ISO 12021, Road vehicles - Sensitivity to lateral wind - Open-loop test method using wind generator input.

[28] ISO 13674-1, Road vehicles - Test method for the quantification of on-centre handling - Part 1: Weave test.

[29] ISO 7401, Road vehicles - Lateral transient response test methods - Open-loop test methods.

[30] ISO 8725, Road vehicles - Transient open-loop response test method with one period of sinusoidal input.

[31] ISO 13674-2, Road vehicles - Test method for the quantification of on-centre handling - Part 2: Transition test.

[32] ISO 8726, Road vehicles - Transient open-loop response test method with pseudo-random steering input.

[33] ISO 3888-1, Passenger cars - Test track for a severe lane-change manoeuvre - Part 1: Double lane-change.

[34] ISO 3888-2, Passenger cars - Test track for a severe lane-change manoeuvre - Part 2: Obstacle avoidance.

[35] ISO 20119, Road vehicles - Test method for the quantification of on-centre handling - Determination of dispersion metrics for straight-line driving.

[36] ISO 9816, Passenger cars - Power-off reaction of a vehicle in a turn - Open-loop test method.

[37] ISO 7975, Passenger cars - Braking in a turn - Open-loop test method.

[38] FMVSS 126, Electronic Stability Control Systems.

[39] SAE J3016, Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.

[40] ISO 4138, Passenger cars - Steady-state circular driving behaviour - Open-loop test methods.

[41] A. Hartwecker, O. Al-Saidi and S. Müller, "Steering at the Powertrain Test Bench – A New Validation Method for Highly Automated Systems up to the Limits of Vehicle Dynamics," in 8th Int. Symposium on Development Methodology, Wiesbaden, 2019.

[42] M. Jokela, M. Kutila and P. Pyykönen, "Testing and Validation of Automotive Point-Cloud Sensors in Adverse Weather Conditions," Applied Sciences, 7 6 2019.

[43] K. Nidhi and S.M. Paddock, “Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?,” RAND Corporation, Santa Monica, CA, 2016, doi: 10.17249/RR1478. Available: https://www.rand.org/pubs/research_reports/RR1478.html.

Appendix B. Fidelity Levels

B.1 Overview of Environmental Fidelity Levels

The methodology introduced in [19] and the PEGASUS six-layer approach [20] were combined to derive this example of classifying fidelity levels for an environmental model. The following tables offer some examples. The first table illustrates a classification of different environmental layers such as infrastructure, conditions, data, and fidelity levels within each. The second table provides more detail of one of these: the environment: street layer (ESL) and its sub-levels.

TABLE B1 Example—Fidelity Levels of the Environmental Model

Model Fidelity Level	Description
ESL	Environment: Street layer
ESL 0	None
ESL 1	2D road network
ESL 2.1	3D road network
ESL 2.3	3D road network incl buildings
ESL 2.4	3D road network incl buildings and surface properties
ETI	Environment: Traffic infrastructure
ETI 0	None
ETI 1	Static traffic signs, lanes
ETI 2	Time/condition-dependent traffic signs
ETI 3	Complete traffic infrastructure
ETM	Environment: Temporal modifications
ETM 0	None
ETM 1	Temporal modifications of ESL and ETI
EMO	Environment: Movable objects
EMO 0	None
EMO 1.1	Basic traffic simulation vehicle with stationary movable object
EMO 1.2	Basic traffic simulation vehicle with replay movable object
EMO 1.3	Basic traffic simulation vehicle with reactive movable object
EMO 2.1	Advanced traffic simulation vehicle with stationary movable object
EMO 2.2	Advanced traffic simulation vehicle with replay movable object
EMO 2.3	Advanced traffic simulation vehicle with reactive movable object
EMO 3.1	High fidelity traffic simulation vehicle with stationary movable object
EMO 3.2	High fidelity traffic simulation vehicle with replay movable object
EMO 3.3	High fidelity traffic simulation vehicle with reactive movable object
EEC	Environment: Environment conditions
EEC 0	None
EEC 1	Rudimentary environment model
EEC 2	Basic environment model
EEC 3	Advanced environment model
EEC 4	High fidelity environment model
EDC	Environment: Data and communication
EDC 0	None

TABLE B2 Example—Deep Dive into the Level ESL

Model Type	Model Description	Description	Results	Common Modelling Methods/Typical Application Areas	Minimal Model Input	Minimum Model Output
ESL	Environment: Street layer					
ESL 0	None					
ESL 1	2D road network	Road geometry (x,y), friction coefficient	Basic geometric road network	OpenDRIVE, Aerial Imagery/Satellite Imagery	x-, y-coordinates	Position of vehicle (x,y) incl orientation (h)
ESL 2.1	3D road network	Road geometry, friction coefficient, elevation data, inclination data, lanes	General geometric road network	OpenDRIVE, Aerial Imagery/Satellite Imagery	x-, y-, z-coordinates, inclination at every point	Position of the vehicle (x,y,z) incl orientation (h,p,r)
ESL 2.3	3D road network incl buildings	Road geometry, friction coefficient, elevation data, inclination data, building footprints, lanes	General geometric road network	OpenDRIVE + 3D Assets (e.g., FBX), Aerial Imagery/Satellite Imagery, OpenStreetMaps	x-, y-, z-coordinates, inclination at every point, position, size, and geometry of the buildings' footprint	Position of the vehicle (x,y,z) incl orientation (h,p,r) and the position of buildings
ESL 2.4	3D road network incl buildings and surface properties	Road geometry, elevation data, inclination data, building footprints, road material, friction coefficient	General geometric road network and detailed surface information	OpenDRIVE + OpenCRG + 3D Assets (e.g., FBX), Aerial Imagery/Satellite Imagery	x-, y-, z-coordinates, inclination, road material and friction at every point, position, size, and geometry of the buildings' footprint	Position of the vehicle (x,y,z) incl orientation (h,p,r), the position of buildings and road surface properties

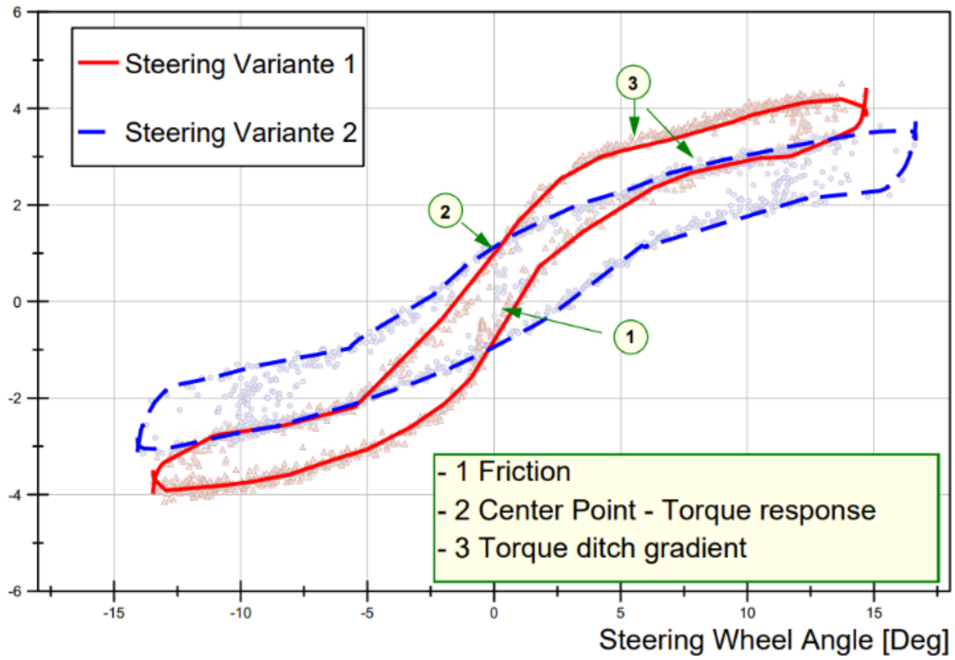
B.2 Examples of Step 1: Validation of the Subsystem Models

B.2.1 Vehicle Dynamics Sub-Models—Example Steering Model

Steering is one part of vehicle dynamics modeling (including braking, powertrain performance, chassis kinematics, etc.) and serves as a good example for subsystem validation and the breadth of complexity that may need to be considered.

The steering system has different properties which need to be considered depending on the use case. The simplest case is to consider only the ratio between steering angle and rack position. However, for many use cases, more complex models are required. For example, steering system models may also require the mass inertia, damping, and coulomb friction. All of these parameters should be modeled and validated on a subsystem level before integrating into a virtual vehicle. The parameter values are determined by physical testing, typically performed on steering test beds in an open loop operation. Figure B1 shows an example of different steering characteristics.

Figure B1 Example—Steering Model



B.2.2 Sensor Sub-Models

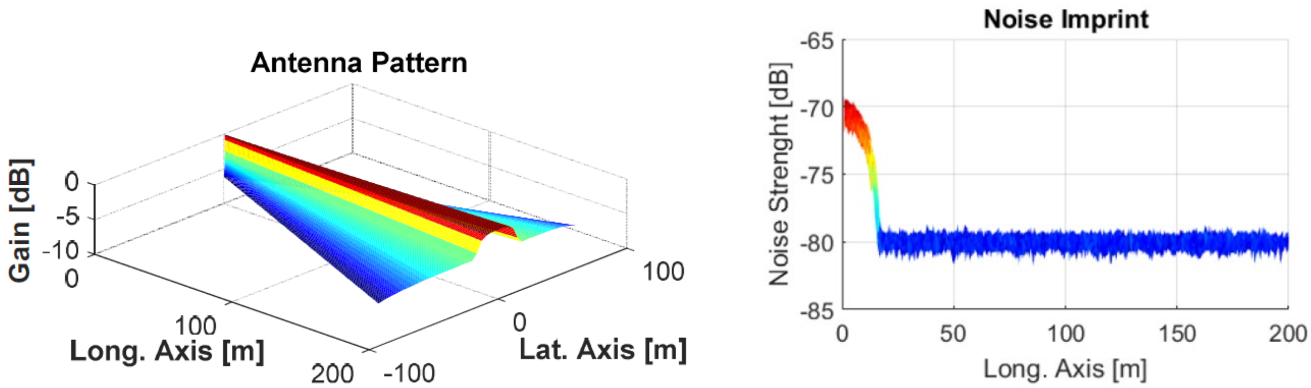
On a subsystem level, sensor models are validated against the manufacturer data sheet with physical tests validating or adding to those specifications. Physical tests should be performed in a simple environment using simple objects since sensor interactions with their environment can be extremely complicated and even more complex to model. The following are some example tests to gather comparison metrics:

Validating Radar Sub-Model:

- Fixed distance: Distance from fixed radar to fixed object.
- Non-fixed distance: Distance from radar to object, one or both in motion.
- Doppler between fixed radar and fixed objects: Verify that it is zero.
- Object width and height.
- Antenna characteristics.
- Field of view.
- Signal to noise ratio (over multiple ranges).
- Object velocity.

Figure B2 shows an example of a radar model considering noise imprint and antenna patterns.

Figure B2 Example—Radar Model [21]



Lidar Sub-Model:

- Distance range: Absolute, accuracy, repeatability.
- Detection range versus object reflectivity.
- Resolution: Horizontal, vertical.
- Field of view: Horizontal, vertical.
- Point cloud update rate.
- Etc.

Camera Sub-Model:

- HDR sensitivity, noise, gamma, MTF, color error.
- Height, width of the object in pixels.
- Range: Distance to the camera.
- Field of view: Horizontal, vertical.
- Intensity distribution comparison (simple histogram comparison).
- WDR, distortion, resolution and bandwidth.
- Etc.

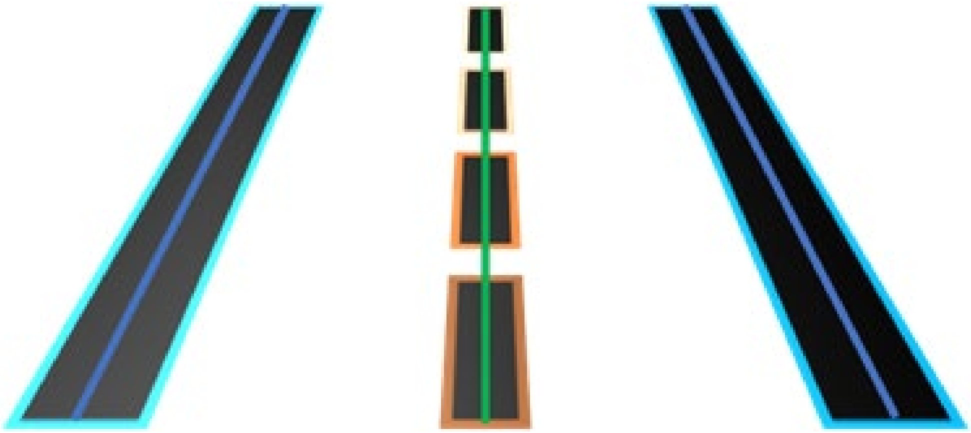
B.2.3 Environmental Sub-Models—Example Lanes

Lane markings can be characterized and modeled in different ways. Like the steering example, complexity can vary greatly and needs to be carefully considered for each use case to ensure satisfactory validation without "over-modeling." Using a broken line as an example (see Figure B3), it can be represented by a collection of single line segments (i.e., each painted portion is its own object of specific length and position), or by its entirety as a solid line of type "broken" (i.e., single object with properties defining the length of and gap). In both cases, there may be uses cases in which finite values defining painted portion length and gaps are insufficient and value ranges are needed. In addition, there are attributes such as color and reflectivity characteristics and state of deterioration (which could be a function of age).

The modeling fidelity required for a particular use case can vary greatly depending on what is being tested. In some cases, the line representation might need to match an exact outline of each marking (what a sensor can detect). In others, a representative line in the middle of the marking, with or without the line width, might suffice. Even for the latter, different representations are possible. The line for example can be a simple polyline with a certain sampling rate or a spline representation. In almost all cases, the most relevant modeling characteristic is the relative accuracy of the lane markings with respect to each other (e.g., lateral distance of the markings, which represents the lane width).

Then, there is the question of location. If a simulation is just about testing lane-keeping functionality, the absolute orientation of the markings may be irrelevant. The markings could be anywhere in a local coordinate system. However, if the subject of the automated system under-test references a position outside of lane markings to determine its behavior (either relative to other objects or in relation to a global or local coordinate system), the absolute or relative position of the lane markings within that context must be defined.

Figure B3 Example—Representation of Lanes and Markings

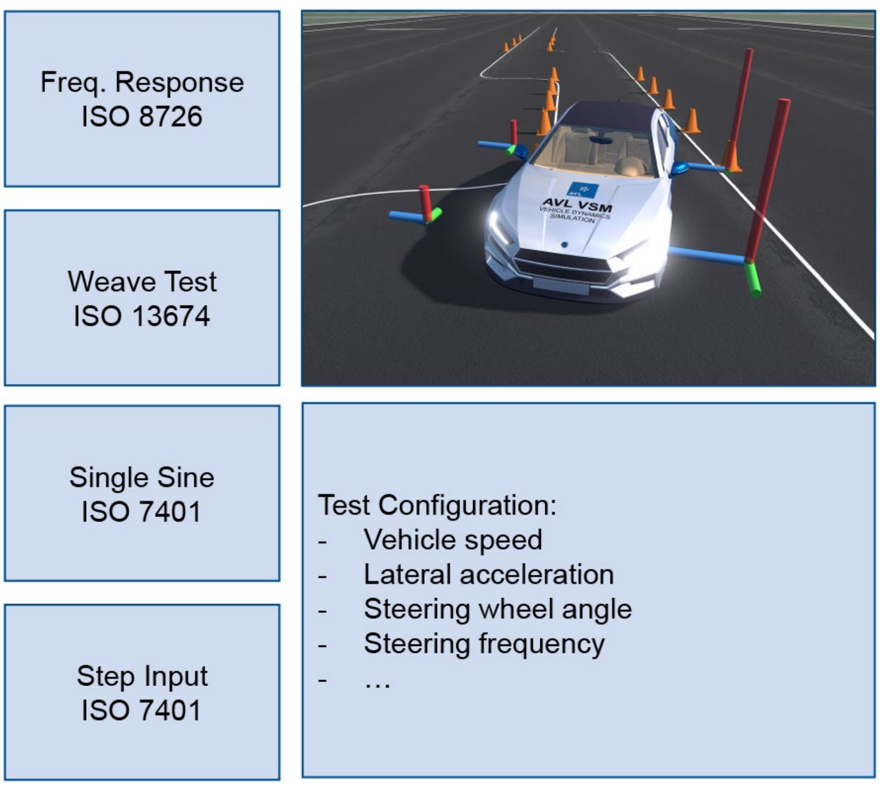


B.3 Examples of Step 2: Validation of the Vehicle System (Passive)

Depending on the use case, different standard tests already exist. Figure B4 shows an overview for the validation of the lateral behavior of the vehicle including the steering system. The vehicle dynamics model is integrated in the environmental model. For the validation of the vehicle system (passive), typically Pegasus Layer 1 is enough.

Especially in the area of vehicle dynamics, most of details are well covered at length in various publications, standards, etc. Due to this, it will not be described here more in detail here.

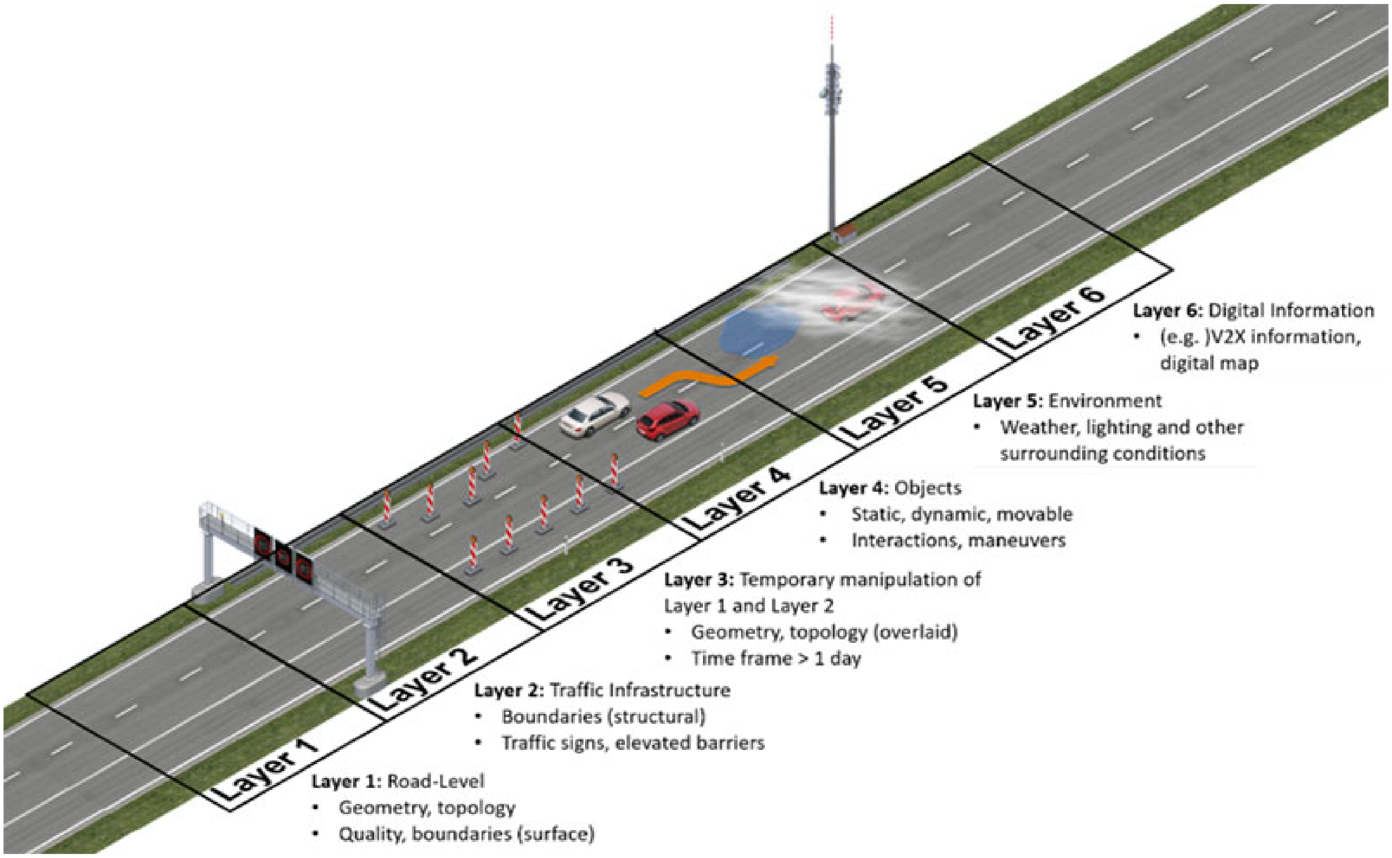
Figure B4 Example Vehicle System—Lateral Dynamics



B.4 Examples of Step 3: Validation of the Sensor System

As the ability to validate the sensor on the subsystem level is very difficult due to the complexity of sensor interaction with the environment, most sensor validation is done at the sensor system level where the sensor model is integrated with the environment model. If referring to the Pegasus Method, it may make sense to validate the sensor system with each of the scenario layers independently (see Figure B5), adding successive layers toward a full integration with the environment. This helps ensure correlation with each layer and facilitates traceability of deviations and anomalies. (Please note that although Pegasus Layer 5 is labeled as "Environment," we use the term generally to refer to all objects and conditions external to the vehicle under test.)

Figure B5 Model for a Systematic Description of Scenarios with Six Independent Layers [9]



In addition to validating the sensor model, this validation approach helps detect fidelity issues with the environment model itself, including object properties and values. When validating the sensor model against the environment model, one will discover that the two should balance to achieve best results. At one extreme, an ultra-high fidelity environment model paired with an ultra-low fidelity sensor model may produce unexpected (and undesirable) results in relation to the baseline. The converse is also true. Classifying the fidelity of the environment and validating the sensor system against the desired level(s) of that classification is very helpful for achieving this balance.

The following provides an example of classifying an environment model in relation to a camera sensor system for validation. It distinguishes the foreground, the background, and the target itself, and their characteristics relevant to the sensor system.

Figure B6 Foreground, Target, and Background Separation

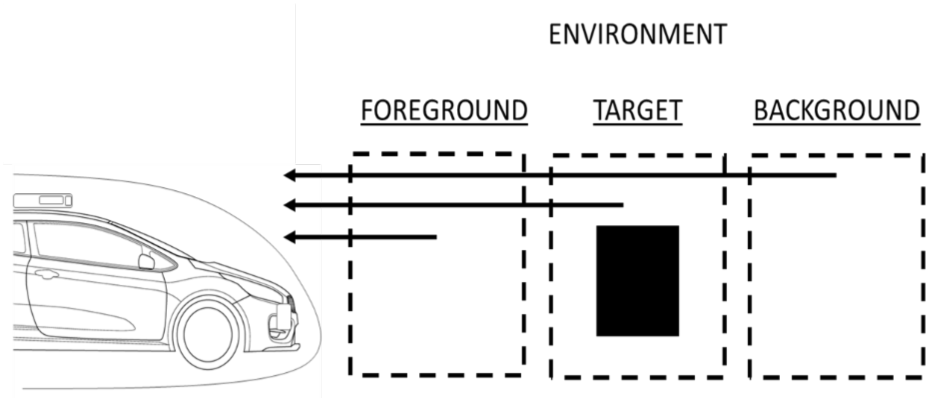


Table B3 Example Target Classification

Target Classification				
	T1	T2	T3	T4
Description	Always able to detect	Average to detect	Difficult to detect	Unable to detect
Examples	Nominal target w.r.t. perception function, e.g., standard test target, or ideal real-world object: - Standard test dummy - Perfectly visible appearance - Best pose to detect - Best target orientation and target trajectory for detection	Shown to be average, median, or common for real-word operating conditions: - 5th to 95th percentile male, female, and age groups - Slightly obscure pose - Varying orientations, e.g., arms over head	Worst-acceptable limit for real-world operating conditions: - Model of child's torso - Very faded lane markings - Highly obscure pose, e.g., radar deflecting - Incredibly fast vehicle, e.g., >2x the speed limit	All target definitions not meeting at least T3 are considered undetectable and considered PERC-E

Table B4 Example Background Classification

Background Classification			
	B1	B2	B3
Description	Ideal background conditions	Average background conditions	Difficult background conditions
Examples	Nominal background conditions for ideal testing or operating conditions: - Flat, well-surfaced pavement - Wide open free space - Large, flat, static, and evenly colored surfaces - No glaring sunlight, no radar interference or excessive echo	Shown to be average, median, or common for real-word operating conditions: - Vertical or horizontal curvatures - Ground clutter or moving debris or animals - Complex shapes, moving trees or people, and irregular color patterns - Some sunlight glare, some radar noise increase	Worst-acceptable background for real-world conditions: - Very hilly, cobblestones, or construction - Reflective surfaces, moving infrastructure - No contrast to target - Direct sunlight, or many glaring surfaces, radar jamming, high density precipitation

Table B5 Example Foreground Classification

Foreground Classification			
	F1	**F2**	**F3**
Description	Ideal foreground conditions	Average foreground conditions	Difficult foreground conditions
Examples	Nominal foreground conditions for ideal testing or operating conditions: - Free space, unoccluded - >1 km visibility conditions - No debris, no precipitation - Sufficient street lighting	Shown to be average, median, or common for real-word operating conditions: - Slight occlusion, objects such as other vehicles or VRUs, or infrastructure, e.g., poles, markers - Slight fog at medium distances - Heavy fog at near distances - Physical clutter	Worst-acceptable foreground for real-world conditions: - Very crowded, aliasing conditions - Heavy fog at medium distances, heavily falling snow or rain - Collected snow, rain, or leaves - Construction or unusual objects - Dirt on sensor or obstruction of sensor internally or externally