FKFS
RESEARCH IN MOTION.

Contributions of:

2024 AutoTest Technical Conference

Testing Hardware and Software in Automotive Development

16 – 17 October 2024

2024 AutoTest Technical Conference
Testing Hardware and Software
in Automotive Development
16 – 17 October 2024 | Stuttgart

FKFS
RESEARCH IN MOTION.

CONTRIBUTIONS 2024 AutoTest Technical Conference

Datengetriebenes Testen in der Automobilentwicklung I

Revealing Challenges for Automotive Interface Testing in the Application of Connectivity Systems
Alexander Prinz, BMW AG; Alexander Gallasch, BMW AG et al.

Ganzheitlicher Ansatz für den durchgängigen Einsatz virtueller Prototypen im Entwicklungsprozess
Jan Stehle, IPG Automotive GmbH et al.

Entwicklung und prototypische Umsetzung von automatisierten Testverfahren nach AFGBV
Toralf Trautmann, Hochschule für Technik und Wirtschaft Dresden (HTWD)

Datengetriebenes Testen in der Automobilentwicklung II

Methodik und Tools zur automatisierten Auswertung von Testdaten
Tim Martini, Hochschule Esslingen

Herausforderungen und Lösungen für die durchgehende Digitalisierung des Testprozesses
Stefan Unterschütz, Werum Software & Systems AG

Testfall-Erkennung in Erprobungsfahrten
Christopher Kober, Mercedes-Benz AG et al.

Virtuelle Absicherung automatisierter und vernetzter Fahrfunktionen I

Automatisierte NCAP Tests mit ASAM OpenSCENARIO
Jakob Kaths, Vector Informatik

Virtueller Test in der praktischen Anwendung
Felix Strauß, Bertrandt Group et al.

Entwicklung und Integration standardisierter virtueller Testszenarien mit TRONIS
Karl Schreiner, TWT GmbH Science & Innovation et al.

Virtuelle Absicherung automatisierter und vernetzter Fahrfunktionen II

Validation and Verification of Lidar System: AI-Generated Point Cloud and Object Sensor Model
Ahmed Yousif, Valeo Detection Systems GmbH

Scenario Generation for Testing Automated Driving Systems
Lukas Birkemeyer, Karlsruher Institut für Technologie (KIT) et al.

Advances in multi-abstraction distributed vECU co-simulation technology for automotive software testing
Markus Wedler, Synopsys et al.

2024 AutoTest Technical Conference

Testing Hardware and Software in Automotive Development

16 – 17 October 2024 | Stuttgart

FKFS
RESEARCH IN MOTION.

KI als Testgegenstand und Testmethode

KI-Lösungen zur Überwachung, Beurteilung und Verkürzung von Prüfläufen
Alwin Tuschkan, AVL List GmbH

Moving towards explainable AI by utilizing a holistic lifecycle assurance framework
Lukas Klose, Keysight Technologies Deutschland GmbH et al.

Testprozesse und Technologie I

Modulares Power HIL Konzept: Flexibles Testen und Validieren bei voller Leistung
Manuel Fischer, dSPACE GmbH

Thermische Testsimulation der CCS-Dose für große Ladeströme zur Überprüfung von Optimierungsansätzen
Jochen Krings, Daimler Truck et al.

Importance of Interoperability Testing in Smart and Bidirectional Charging Systems
Jan Roßa, P3 Automotive GmbH et al.

Testprozesse und Technologie II

Hochautomatisierte Test- & Integrationspyramide von Softwaremodultest bis Gesamtfahrzeugintegration
Sebastian Dietze, tracetronic GmbH Maximilian Leicht, BMW AG et al.

Glaubwürdigkeit von Software-in-the-Loop-Testsystemen - Potentiale und Herausforderungen
Anne Kieneke, Dr. Ing. h.c. F. Porsche AG et al.

Tests und Fahrversuche für sehr leichte elektrische Nutzfahrzeuge
Andreas Daberkow, Hochschule Heilbronn

Revealing Challenges for Automotive Interface Testing in the Application of Connectivity Systems

Dr. Alexander Prinz
BMW AG
Munich, Germany
alexander.prinz@bmw.de

Hanxiang Zhu
BMW AG
Munich, Germany
hanxiang.zhu@bmw.de

Alexander Gallasch
BMW AG
Munich, Germany
alexander.gallasch@bmw.de

Abstract: This paper provides an overview of test methods for interface testing in Vehicular Connectivity Systems (VCS) using Cellular Vehicle-to-X (C-V2X) technology. After giving an introduction in the model-based test approaches in the field of automotive electronic control unit testing, we define a generic architecture for C-V2X applications. With the help of previous foundation, we identify the relevant components for Vehicle-to-Network (V2N) applications. Subsequently, we apply different test approaches based on simulation methods and test-interfaces as well as fleet-measurements. After analyzing the fleet data, we also consider the application of AI (Artificial Intelligence) based methods to optimize the performance and the reliability of the test results. In conclusion, we identify missing test approaches and -strategies and define optimizations for tooling, test-setups, data quality and measurement features for AI test-applications.

Automotive-Test, Design-for-Test, Test-Interface, Electronic Control Unit (ECU), Vehicle to X (V2X)

1 Introduction

X-in-the-loop (XiL) test methods are an inherent part of the test and development of electronic systems in the field of automotive production. Reference [FHW16] summarizes the constraints for the successful application of Hardware-in-the-loop (HiL) testing in the field of driver assistance systems:

- Closed-Loop HiL testing combines a real-time simulation with an unit-under-test (UuT). The test reaction of the electronic control unit is comparable to the real system behaviour. (e.g. behaviour inside the car)
- The injection describes the interface between simulation and UuT.
- For driver assistance, HiL-testing uses the injection on a hardware level and can be directly applied for e.g. sensor-testing. This means the sensor will be stimulated or the measured data will be provided to the control loop of the UuT using bus interfaces.
- The injection on the software level depends on the availability of interfaces in the software architecture of the UuT. The software-injection point has to ensure the real-time test operation on the HiL-test-bench.

The presented constraints are also transferable to other domains in the field of automotive electronic-control-units (ECU) like entertainment, power-train or suspensions. The sensors in the different application-fields might vary.

Contribution: 2024 AutoTest Technical Conference | 16 - 17 October 2024 | Stuttgart

The so-called unit-under-test in the HiL-benches differs from the development phase of the vehicle project. Starting with component HiL-testing in an early stage, also the combination of several ECUs in a bundle will ensure that errors will be determined before system integration in the vehicle. The main drawback of bundle tests compared to single ECU-tests is the traceability of the error to the root cause. For that reason, the automotive industry has implemented interface-tests in its HiL-portfolio. In [Fil19] the main aspects of interface-testing are described:

- These interfaces are mostly communication interfaces between different ECUs that represent a vehicle function.
- When testing an interface one counterpart of the interface will be replaced by a simulation. Latter has to fulfill the requirements of real-time application, interface-stability and parameterization.
- The validation of interface-models can be more difficult than simulation-models of whole ECUs because repercussions can´t be eliminated every time and the availability of measurement data is not ensured in all processing layers.

To create the development of a safe driving environment, extensive testing is conducted throughout the different phases of a vehicle project. The keys to these innovative technologies are high safety, high efficiency, high reliability, and low latency. C-V2X is one such technology. It is a wireless technology which is utilized in autonomous driving and intelligent transportation systems (ITS), which entirely relies on communication systems, therefore, interface-testing is the ideal method to check and ensure its functionality.

V2X technology is actually not a recent concept. The establishment of standards of V2X was first proposed in 1999 by the US Federal Communications Commission (FCC) [FCC99]. In the first stage of this technological development, people attempted to achieve interconnectivity within a local area network using Dedicated short-range communications (DSRC) technology in the U.S. or ITS-G5 (Car2Car) in Europe, which is based on IEEE 802.11p. With the continuous development of communication technology, the emergence of LTE (Long-Term Evolution) greatly improved people's lives. C-V2X was then born with it and is a 3rd Generation Partnership Project (3GPP) standard by using the fourth generation LTE (4G) or the fifth generation (New Radio 5G NR) connectivity for signal transmission and reception. This technology enables real-time communication and coordination between vehicles, infrastructure, pedestrians, and the network, ultimately contributing to a safer and more efficient transportation system. C-V2X can be divided into four types (see Figure 1) [MKOI18].

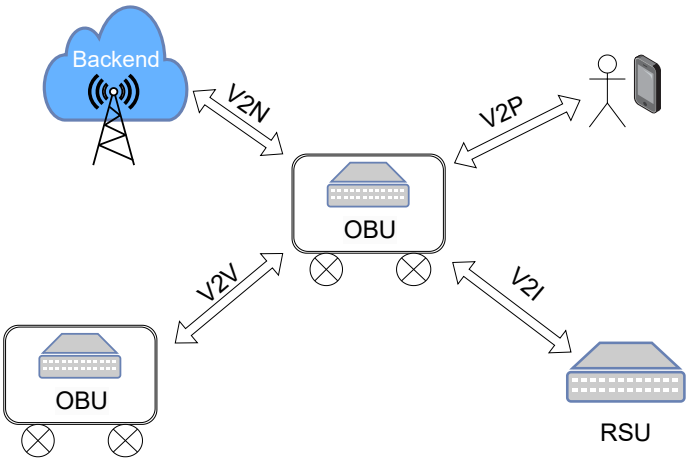


Figure 1: Applications of C-V2X

1.1 Device to device (D2D)

- V2V (Vehicle to Vehicle): Communication between vehicles to exchange the driving data, e.g., the distance between each other or the decision of lane changing.

- V2I (Vehicle to Infrastructure): Communication between vehicle and RSU (Road Side Unit) that the road infrastructure is connected with, e.g., traffic light, road sign and roadside sensors.
- V2P (Vehicle to Pedestrian): Communication between vehicle and pedestrian (their mobile devices) to remind both parties to pay attention to the traffic situation to avoid casualties.

For D2D, the communication is realized without utilizing the cellular network. The interface PC5 (Proximity Communication) at 5.9 GHz is used. It is a proximity Service which enables direct communication within a local region.

1.2 Device to network (D2N)

- V2N (Vehicle to Network): Communication between vehicle and backend to achieve better traffic efficiency through such as cloud services.

For D2N, cellular networks are used for V2N Applications. It is a Uu interface or also called a radio interface between the User Equipment (UE) and Node B. Node B is the cell tower better known as a base station.

2 Architecture of C-V2X

Figure 2 shows the scenario of the overall C-V2X system. ”Connectivity” is the most important aspect here. Each unit is connected to at least one other unit. This is made possible by either wireless or wired connections and their underlying network protocols. In this chapter, the main components of the architecture as well as their interfaces are described.

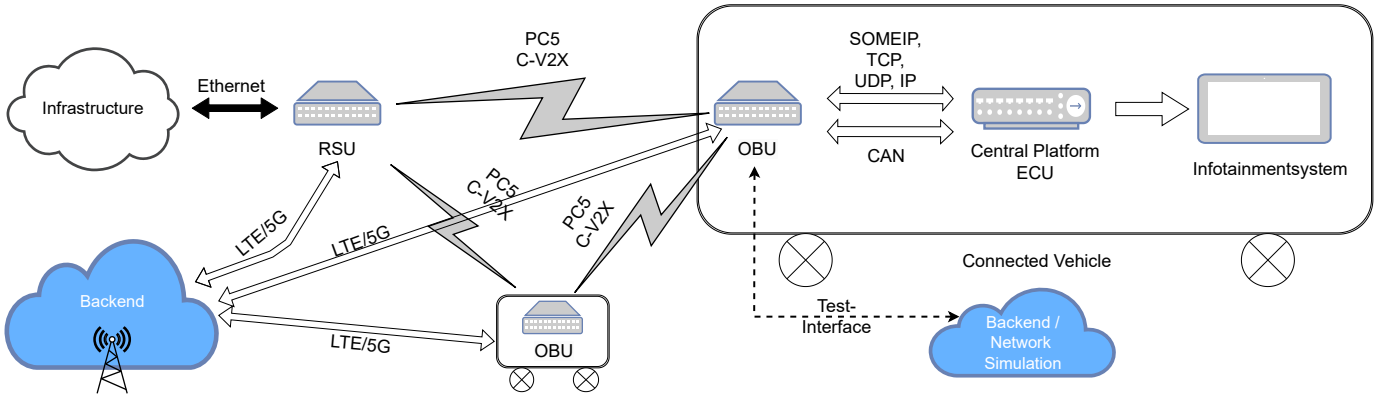


Figure 2: C-V2X System Architecture (with reference to [MVH21])

2.1 Roadside-Unit (RSU)

The RSU is a wireless transmitting device that enables sidelink communication (D2D) through the PC5 interface, eliminating the need for cellular networks. PC5 interface builds local communication within a short

range (max. $1\,\mathrm{km}$). In this predefined area, the RSU can broadcast real-time information from the connected units (infrastructure like traffic lights) to vehicles. Additionally, if an eSIM-card (embedded Subscriber Identity Module) is integrated in the RSU, the data can also be transmitted directly over the cellular network [MVH21].

2.2 On Board Unit (OBU)

The OBU is a wireless communication device installed in vehicles. With the OBU, the performance of autonomous driving can be significantly improved through direct communication between the vehicle's OBU and RSUs, as well as other OBUs. The most important information, such as the vehicle's location, direction, and speed, is sent and received here.

C-V2X can address all possible connectivity challenges. Sidelink communication over the PC5 interface ensures low latency and spatial resource reuse within the local field. Outside its local coverage, the Uu interface via LTE/5G bridges the gaps and aids in more efficient centralized control [GAA18]. Combined with autonomous driving systems (Data Analysis from Camera, LiDAR, Radar, GNSS, etc.), C-V2X can solve many challenges encountered during driving: platooning (maintaining a safe distance between platoons of vehicles), extended sensors (exchange of sensor data), advanced driving (semi-automated or fully-automated driving), and remote driving (drive for disabled passenger or in dangerous environment) [TrGPP18b]. In this paper, V2N communication via the Uu interface is focused on automobile connectivity, such as entertainment services.

To realize V2N communication, an OBU integrated with an eSIM card is a prerequisite. By connecting a small single-board computer with an LTE module, we have built our own OBU device and controlled it to test connectivity while the vehicle is in motion. Doppler shift was not considered because the driving speeds were approximately within 120 kilometers per hour (km/h) where LTE network performance can be considered unaffected if speeds are below 120 km/h [TrGPP12]. As the vehicle is not in a static position, the connected base station/Node B is changing over time for real driving scenarios. Therefore the connected vehicle experiences varying network quality and speed rates.

3 Interface- and Function-Testing by example of V2N

This chapter contains various aspects of the development, testing and launch of connected vehicles based on V2N-technology. Each method has various scopes and, therefore, contributes differently to the functionality and quality of connected services. As V2N-connection quality and data rates are important for the development of connected vehicles, we describe V2N-traffic simulations (HiL) for the early development phase. The testing goals, including functional and stress tests, will be taken into account. In the second step, the V2N-System will be launched and LTE-raw-data from real traffic scenarios will be evaluated. In the end, high-level V2N-data-rates will be analyzed. The quality of the collected data will be considered for applications like big data and artificial intelligence model approaches. The ultimate goal in future work is to achieve a V2N-network speed and quality model to promote and maintain the development of connected vehicles that rely on V2N-technology.

3.1 V2N-Traffic Functional and Stress Test Simulation

As a first step, the presented electronic control units in the connected vehicle, as shown in Figure 2, were tested in a HiL-setup. To fulfil functional as well as stress tests, we considered a simulated V2N connection for flexibility. The network simulation is connected to the OBU-system with the test-interface in Figure 2. It includes an interpreted network communication model based on OSI-Layer four up to seven. When running the test-setup, different requests and responses were triggered by the simulation, and the OBU received and interacted with the simulated backend / network-simulation with the given test-interface. Metrics such as jitter and latency for repetitive request-response communication, packet loss for high-frequency request queues, and connection loss can be evaluated in different simulated communication scenarios for the V2N vehicle (see Figure 3). The evaluation was implemented with a self-developed software. Latter already includes a framework for some of the required transport protocols and is easily adapted and modified to support application specific protocols. It delivers an illustrated test report, which gives the tester a quick overview of performance and results.

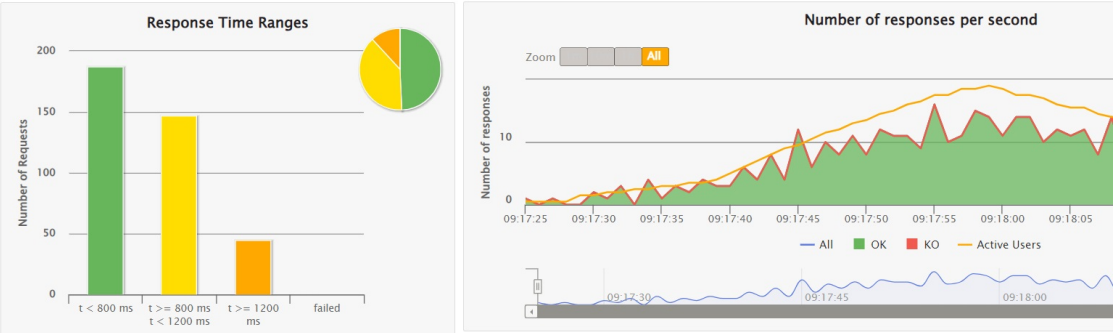


Figure 3: Report and metrics for response times from the V2N-Traffic HiL-Test

3.2 V2N-Measurement and Evaluation for OBU

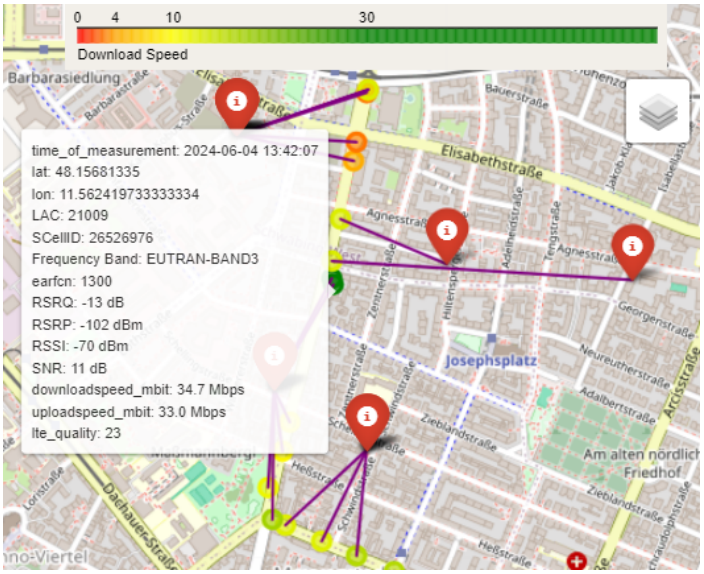


Figure 4: V2N-Traffic Data Analysis: Tower-Assignment and LTE-Raw Data. MNO: DE-Telekom

Beside simulation, also real vehicle measurements, with reference to the measurement setup in Figure 2, will be elaborated. The hardware part consists of LTE communication devices made up of components Raspberry Pi 4 integrated with a GPS antenna, and the Simcom Sim7600X (LTE Module) as the OBU. The selected MNO (Mobile Network Operator) is DE-Telekom. In different scenarios, we conducted multiple tests specifically targeting V2N communication:

- An urban route with the best infrastructure
- An inter-district urban route with moderate infrastructure
- A long-distance route with weaker infrastructure

In Figure 4, a static measurement in an urban area was considered. The visualized LTE-Raw data provides an overview of the parameters which reflect the overall quality of the current signal. The most important values here are RSSI (Received Signal Strength Indicator), RSRP (Reference Signal Received Power), RSRQ (Reference Signal Received Quality), and SNR (Signal-to-noise ratio). Besides the SNR value, they depend on each other sequentially. RSSI assesses the average total received power observed within the measurement bandwidth across N ressource blocks (RBs). This power indicated by the carrier RSSI encompasses contributions from co-channel serving and non-serving cells. RSRP represents the power of the LTE Reference Signals distributed across both the full bandwidth and narrowband. Generally, the RSRP value determines which base station the LTE module will connect to. RSRQ is a C/I (Carrier-to-Interference) type of measurement and reflects the quality of the received reference signal. The RSRQ measurement provides supplementary information when RSRP alone is insufficient for cell reselection decisions [PBM+21] [AM17].

The circles represent the positions where we conducted our tests. Depending on upload and download data rates tested in megabits per second (Mbps), they are assigned different colours (as seen at the top of Figure 4). The connected 4G base stations (red pin needles) are found with the help of detected cell ID (SCellID) and are linked to these positions by purple line segments. The switches between different cell towers can be observed in the Figure 4. The key requirement for connected apps and services inside the vehicle is to prevent time-limited failures. For entertainment services, for example, the required buffers have to be sufficiently large to provide seamless usage. The requirements can be derived with big-data fleet analysis to be aware of the worst-case scenarios. Using the real data we collected, we conducted a detailed analysis. During the initial check of the data's validity, we already found significant discrepancies, which led to a loss of accuracy. Due to the importance of the RSRP value described earlier, it was analysed first versus the RSSI value on which it depends (Equation 2). A linear relationship between the two values is to be expected since both units are logarithmic, and N remains constant because the channel bandwidth has not changed according to the recorded data. However, what we discovered through the data is that the RSRP values fluctuate by ±10 dBm for each RSSI value, which makes it scientifically unusable. It could be due to the Sim7600H hardware not always functioning stably, resulting in the data not always being scientifically accurate, which is also noticeable when we compare the LTE quality values with the RSSI values. According to 3GPP [TrGPP18a], the two are always clearly defined in relation to each other, which is not reflected in the recorded data.

3.3 V2N-Network Speed-Test Verification and Elaboration of the data quality for AI-Applications

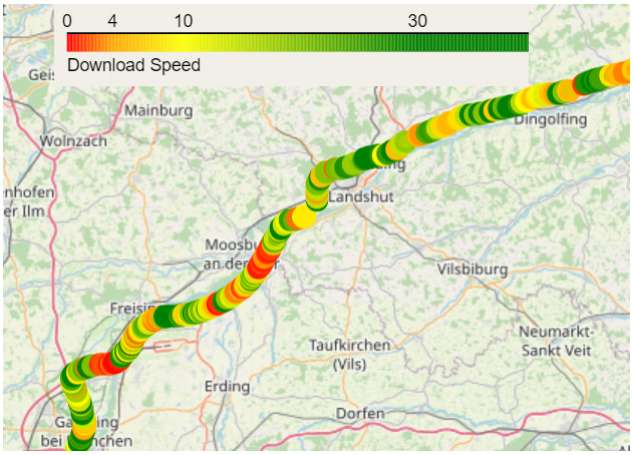


Figure 5: V2N-Traffic Data Analysis: Download-Rate over Driving-Position. MNO: DE-Telekom

After the collection of raw data and further processing have been finished, the following features became available (See Table 1), which we have used to train an AI model to predict the stability of a V2N system. Three different algorithms have been chosen, which are listed in Table 2. Even though we used the same inputs as those in some published works [ATSM+23] [EAAz+], we were not able to reproduce their results. The accuracy of both models we trained using KNN and RF with Grid Search was around 66%. What has already been noticed during the research is that there is no clear scientific solution to consistently predict LTE network performance. The selected models in these two example papers produced very different results. The possible reasons for this phenomenon

could be the significant differences in 4G/5G infrastructure across different regions and the individual differences in LTE modules produced by various companies. We believe that more importantly, additional features are needed for more accurate learning and prediction, which are listed in Table 3. The first also the most important one is every RSRP value of available surrounding eNodeBs which is not only needed for understanding the feasibility of V2N on different road segments but also for calculating approximate SINR values that provide a more accurate reflection of the actual network environment [DBT15]. There are many LTE Modules on the market that have the capability to obtain the RSRP values of four dominating surrounding eNodeBs, which is unfortunately not our case. Cell load also plays a role, varying depending on the time of day. If the load exceeds 70%, the signal quality will deteriorate [SVY06]. A complete digital twin can be considered as the ultimate goal that many fields aim to achieve. Details like signal reflections or wall penetration can be accurately simulated to ensure theoretical plans become a reality. A famous ongoing project is Nvidia's Aerial Omniverse Digital Twin [Cor24].

Table 1: Overview of the available parameters in V2N-Measurement data.

Available Features	Description
Radial distance to eNodeB	The absolute distance between the measurement point and eNodeB, where eNodeB is LTE base station.
RSSI	RSSI (dBm) $= 10 \cdot \log_{10}\left(\frac{P(\text{mW})}{1\text{mW}}\right)$ (1)
RSRP	RSRP (dBm) $=$ RSSI (dBm) $- 10 \cdot \log_{10}(12 \cdot N)$(2)
RSRQ	RSRQ (dB) $= 10 \cdot \log_{10}(N) +$ RSRP(dBm) $-$ RSSI(dBm)(3) where N is the number of resource blocks (RBs) as per channel bandwidth
SNR	SNR (dB) $= 10 \cdot \log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right)$ (4)
Upload rate (Mbps)	Measurement of the average performance of the internet connection by calculating the upload speed between the device and the selected server
Download rate (Mbps)	See upload rate above

Table 2: Overview of used AI-Algorithms with reference to [HTF09]

Used algorithms	Description
K-Nearest Neighbors (KNN)	KNN classifies an instance by analyzing the labels of the k closest data points in the feature space.
Random Forest (RF) with GridSearchCV	RF combines the predictions of multiple decision trees, each built on a different subset of the training data. GridSearchCV uses cross-validation to choose the best one.
Neural Network (NN)	NN consists of multiple layers of neurons with activation functions, learning from data through forward and backward propagation, adjusting weights to minimize error.

Table 3: Overview of the required parameters in V2N-Measurement data. (Design-for-Test)

Required Features	Description
Available eNodeBs	The corresponding RSRP values of available surrounding eNodeBs.
Signal-to-interference-plus-noise ratio (SINR)	According to the research in [DBT15], with corresponding RSRP values of available surrounding eNodeBs, SINR can be well approximated.
Cell load	The performance of cells at different time intervals.
3-D environment model	A complete digital twin of infrastructure and construction.

4 Conclusion

Summing up above, the testing of V2N-technology for connected vehicles is challenging and requires various test-solutions and data-analysis. First, we have proposed an approach for HiL-based functional and stress testing, which requires suitable design-for-test interfaces, simulation models and test-automation. Then, we have discussed various V2N-vehicle fleet data. We elaborated that the LTE-raw data could not be verified with the given physical dependencies. This could have it´s root cause to measurement-effects that were not considered so far, such as cell load, construction layout of nearby buildings, weather, and other disturbing factors. In the end, the evaluation of the V2N network speed over a long driving period provides a good overview of the overall vehicular connectivity performance. These measurement data were used for training AI algorithms, as shown in Table 2. The currently achieved data quality and amount were not sufficient for a successful application to predict download- and upload-rates. Future work will focus on the repeatability and stability of LTE raw data and download- and upload-rates. This supports the development of future applications for vehicular connectivity, as a seamless connection with high data rates is mandatory at all times.

Acknowledgment

At this point we appreciate the collaboration of all colleagues that worked on this project. Namely, we thank Sebastian Buresch; BMW AG, Jonas Schmitz; BMW AG and Julian Wiesner; BMW AG for their effort.

References

[AM17] Nadia Awad and Is-Haka Mkwawa. The impact of the reference signal received power to quality of experience for video streaming over LTE network. *2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT)*, pages 192–196, 2017.

[ATSM+23] A. Al-Thaedan, Z. Shakir, A. Y. Mjhool, et al. Downlink throughput prediction using machine learning models on 4G-LTE networks. *International Journal of Information Technology*, 2023.

[Cor24] Nvidia Corporation. NVIDIA Aerial Omniverse Digital Twin. https://developer.nvidia.com/aerial-omniverse-digital-twin, 2024. [Online; accessed 18-August-2024].

[DBT15] Milutin Davidovic, Sanja Bjekovic, and Igor Tomic. On the impact of network load on LTE network downlink performance. 06 2015.

[EAAz+] Habiba Elsherbiny, Hazem M. Abbas, Hatem Abou-zeid, Hossam S. Hassanein, and Aboelmagd Noureldin. 4G LTE Network Throughput Modelling and Prediction. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*.

[FCC99] FCC Federal Communications Commission. FCC ALLOCATES SPECTRUM IN 5.9 GHz RANGE FOR INTELLIGENT TRANSPORTATION SYSTEMS USES Action Will Improve the Efficiency of the Nation's Transportation Infrastructure. https://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html, 1999. [Online; accessed 22-December-2023].

[FHW16] Marius Feilhauer, Juergen Haering, and Sean Wyatt. *Current Approaches in HiL-Based ADAS Testing*, volume 9. 2016.

[Fil19] Thies Filler. *Entwicklung einer Methodik für die durchgängige Integration von Hardware und Softwaremodellen in Simulationen für Fahrfunktionen*. Springer-Verlag, Berlin Heidelberg New York, 2019.

[GAA18] 5GAA 5G Automotive Association. Cellular-Vehicle-to-Everything (C-V2X): today and next steps. https://www.gsma.com/iot/wp-content/uploads/2020/07/02_5GAA_Maxime-Flament.pdf, 2018. [Online; accessed 13-January-2024].

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. 2 edition, 2009.

[MKOI18] Zachary MacHardy, Ashiq Khan, Kazuaki Obana, and Shigeru Iwashina. V2X Access Technologies: Regulation, Research, and Remaining Challenges. *IEEE Communications Surveys and Tutorials*, 20(3):1858–1877, 2018.

[MVH21] Lili Miao, John Jethro Virtusio, and Kai-Lung Hua. PC5-Based Cellular-V2X Evolution and Deployment. *Sensors*, 21:843, 2021.

[PBM+21] Gubtha Mahendra Putra, Edy Budiman, Yonatan Malewa, Dedy Cahyadi, Medi Taruk, and Ummul Hairah. 4G LTE Experience: Reference Signal Received Power, Noise Ratio and Quality. In *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pages 139–144, 2021.

[SVY06] Iana Siomina, Peter Varbrand, and Di Yuan. Automated optimization of service coverage and base station antenna configuration in UMTS networks. *IEEE Wireless Communications*, 2006.

[TrGPP12] 3GPP The 3rd Generation Partnership Project. 3GPP TS 36.300 version 9.9.0 Release 9, 2012.

[TrGPP18a] 3GPP The 3rd Generation Partnership Project. 3GPP TS 27.007 version 15.2.0 Release 15, 2018.

[TrGPP18b] 3GPP The 3rd Generation Partnership Project. Study on Enhancement of 3GPP Support for 5G V2X Services. https://www.3gpp.org/ftp/Specs/archive/22_series/22.886/, 2018. [Online; accessed 13-January-2024].

Virtuelle Validierung von Fahrfunktionen entlang einer kontinuierlichen Werkzeugkette

Dr. Michael Kochem, Jan Stehle
IPG Automotive GmbH
Fautenbruchstraße 46
76137 Karlsruhe
michael.kochem@ipg-automotive.com
jan.stehle@ ipg-automotive.com

Abstract: Moderne Fahrzeuge werden durch neue Softwarearchitekturen immer komplexer und deshalb häufig als „Software-defined Vehicle“ bezeichnet. Die Entwicklung, Validierung und spätere Aktualisierung sind mithilfe des realen Fahrversuchs nur sehr eingeschränkt und in einigen Fällen gar nicht realisierbar beziehungsweise wirtschaftlich unrentabel. Mithilfe der virtuellen Fahrzeugentwicklung kann in jeder Entwicklungsphase bis hin zum realen Test eine durchgängige Validierung der Software auf Basis reproduzierbarer Tests erfolgen.

Um diesen Ansatz in bestehende Prozesse zu integrieren, ist eine übergeordnete Ebene für den einfachen Zugang zur Simulation erforderlich. Diese muss eine Möglichkeit zur intuitiven Verwaltung und Versionierung von Datenmengen, Simulationsmodellen und Software bieten. Als Werkzeug für das Workflow-Management in der Simulation unterstützt VIRTO zahlreiche Testmethoden. Daten, Testabläufe und Ergebnisse können für jede Entwicklungsphase wiederverwendet und der Entwicklungsaufwand erheblich minimiert werden.

1 Einleitung

In der Vergangenheit haben sich Fahrzeuge vor allem durch ihre Auslegung auf Basis verschiedener Hardware-Komponenten unterschieden, heute jedoch wird die Software immer mehr zum zentralen Unterscheidungsmerkmal und zum bestimmenden Faktor für die Eigenschaften eines Fahrzeugs.

Die kontinuierliche Weiterentwicklung der Fahrzeugsoftware ist aufgrund ständig neuer Anwendungen und Möglichkeiten erforderlich – auch dann, wenn sich das Fahrzeug bereits im Feld befindet. Darüber hinaus ist eine deutliche Verkürzung von Entwicklungszeiten (time-to-market) gefordert, da Fahrzeugbesitzer*innen zunehmend Softwareupdates (idealerweise Over-the-Air, kurz OTA) über den gesamten Produktlebenszyklus erwarten.

Diese Updates ermöglichen auch noch nach mehreren Jahren Fehlerbehebungen, Verbesserungen oder sogar neue Fahrzeugfunktionen. Dadurch kann jedes Fahrzeug zu einem Unikat werden. Aus Sicherheitsgründen ist es jedoch zwingend erforderlich, alle neuen Funktionen vor ihrem Einsatz im Feld ausgiebig zu testen und abzusichern. Dies wiederum ist mit hohen Kosten und einem enormen Aufwand für die Absicherung und Validierung verbunden.

Zur Überprüfung neuer Softwarefunktionen per Simulation haben sich verschiedene Methoden etabliert − beispielsweise Open-Loop- beziehungsweise Data-Replay-Testing. Bei dieser Methode können neue Algorithmen mit einer großen Menge von Messdaten, die auf RADAR-, LIDAR- oder Ultraschallsensoren basieren, sowie Fahrzeugmessdaten beaufschlagt und getestet werden. Im scheinbaren Gegensatz dazu steht das szenarienbasierte Testen, bei dem sich ein virtuelles Fahrzeug – ausgestattet mit der gleichen Sensorik und Aktorik wie das reale Fahrzeug – in einer simulierten Umgebung bewegt. Dies wird auch als Closed-Loop-Testing bezeichnet, da dabei alle Regelkreise geschlossen sind. Eine Zwischenstufe bilden gemessene Szenarien, die bei Bedarf mit künstlichen Objekten angereichert werden können. Eine weitere Zwischenstufe stellen künstliche Szenarien dar, die aus echten Szenarien generiert wurden, um die gleiche Situation in variablen Versionen darzustellen.

2 Die Rolle des virtuellen Fahrversuchs

Da Fahrfunktionen nicht nur innerhalb der eigenen Domäne, sondern auch im interdisziplinären Kontext fehlerfrei funktionieren müssen und Tests idealerweise im Gesamtfahrzeug durchgeführt werden sollten, ist deren Absicherung mit enormen Anforderungen verbunden. Hinzu kommt, dass Fahrzeuge und ihre Funktionen in immer kürzeren Zeitabständen aktualisiert und optimiert werden. Die Entwicklung, die Validierung und die spätere Aktualisierung mithilfe realer Fahrversuche ist daher nur sehr eingeschränkt und in einigen Fällen gar nicht realisierbar beziehungsweise wirtschaftlich unrentabel – eine Anpassung und Optimierung der Fahrzeugentwicklungsprozesse ist erforderlich.

Ein geeigneter Lösungsansatz ist die virtuelle Fahrzeugentwicklung. Dabei werden auf Basis eines virtuellen Fahrzeugs und/oder einer virtualisierten Umgebung die relevanten Entwicklungsbereiche in der Simulation abgedeckt, indem die zu testende Software in ein virtuelles Gesamtfahrzeugmodell integriert wird.

Um eine Durchgängigkeit des virtuellen Fahrzeugs (Digital-Twin) bis zum End-of-Life des realen Fahrzeugs zu gewährleisten, ist eine frühzeitige Integration unerlässlich. Deshalb ist es entscheidend, dass die Simulationsmodelle basierend auf konsistenten Daten zunehmend an Reife gewinnen. Eine durchgängige Validierung der entwickelten Software auf Basis reproduzierbarer Tests sollte in jeder Entwicklungsphase bis hin zum realen Test erfolgen.

2.1 Übergeordnete Datenverwaltungs-/Workflowmanagement-Ebene

Ein geeignetes Werkzeug für das Workflow-Management muss in erster Linie die Simulation mit diversen Testmethoden wie MIL/SIL/HIL/VIL, Open- und Closed-Loop-Tests sowie Prüfgeländetests mit realen Targets unterstützen. Entscheidend ist, dass die Daten- und Workflowverwaltung für alle Testmethoden geeignet sein muss. Die Mindestanforderung jedoch ist, dass vergleichbare Entwicklungswerkzeuge, beispielsweise die 0D- und 1D-Simulation, dazu in der Lage sein müssen, die erforderlichen Daten auszutauschen oder in einem Prozess zusammenzufassen.

Um dies zu gewährleisten, ist eine übergreifende Ebene der Datenverwaltung erforderlich. Diese muss die involvierten Mitarbeitenden dazu befähigen, die anfallenden Datenmengen zu verwalten, zu überprüfen und den Benutzer*innen und Anwendungen auf einfache Weise zur Verfügung zu stellen. Darüber hinaus muss die Software eine intuitive Verwaltung und Versionierung der anfallenden Daten, Simulationsmodelle und Software ermöglichen, siehe Abb. 1.

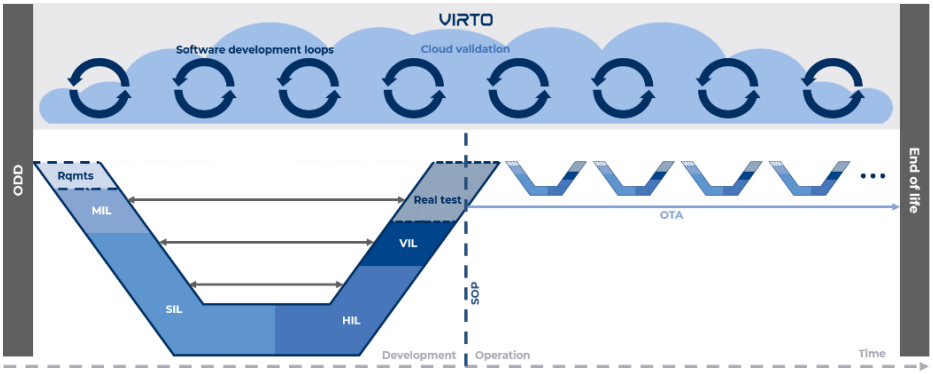


Abb. 1. Übergeordnete Datenverwaltung zur Unterstützung des Entwicklungs- und Maintenance-Prozesses, primär der Simulation

Das vorrangige Ziel ist es, die Simulation auch für Personen ohne Simulationserfahrung nutzbar zu machen. Den Schlüssel hierzu bilden konsistente Daten und Modelle des Fahrzeugs, der Komponenten, der Umgebung und der Manöver und Tests. Die Datenqualität steigt im Laufe der Entwicklung mit den Erkenntnissen aus dem Feld.

Sinnvollerweise sollte sich dies automatisch in den Simulationsmodellen widerspiegeln. Die in den Entwicklungsprozess eingebundene Datenverwaltungs- und Workflowmanagement-Software führt zur Etablierung der Simulation als essenzieller Bestandteil konventioneller Entwicklungsprozesse.

2.2 Optimierung des Entwicklungsprozesses

Nach der vorgelagerten Festlegung der Operational Design Domain (kurz ODD) [1, 2] für die bestimmungsgemäße Funktionsweise des Fahrzeugs oder einer Funktion ergeben sich die Anforderungen an das Gesamtsystem und die Teilsysteme. Auf dieser Grundlage erfolgt anschließend die Definition der Lastenhefte.

Klassischerweise würde die Entwicklung des Fahrzeugs oder einer bestimmten Funktion ausschließlich auf der Basis vorhandener Hardwarekomponenten inklusive realer Steuergeräte erfolgen. Diese Vorgehensweise ist allerdings zeitaufwendig und daher nur schwer mit einem agilen Software-Entwicklungsprozess vereinbar. Daher sollte die virtuelle Entwicklung frühzeitig eingebunden zu einem zentralen Bestandteil der Entwicklung werden. Dieser Prozess kann durch ein Daten- und Workflow-Management in der Cloud unterstützt werden, wobei die Simulation im Idealfall nahtlos integriert wird, sodass sie unbemerkt zum Einsatz kommt.

Anschließend erfolgt die Durchführung der verschiedenen Testmethoden. Dabei ist stets eine Durchgängigkeit der Daten des virtuellen Prototyps und der Umgebung gewährleistet. Die Eingangsdaten sowie die neu generierten Daten werden in der Cloud verwaltet, zueinander in Beziehung gesetzt und im Rahmen des Workflowmanagements für alle Testmethoden bereitgestellt. Um eine kontinuierliche Absicherung zu erreichen, können parallel dazu wiederholt Simulationen in großer Zahl in der Cloud durchgeführt werden. Sämtliche Daten, Testabläufe und Ergebnisse können für jede Entwicklungsphase wiederverwendet werden, was zu einem erheblich minimierten Entwicklungsaufwand führt. Dieser Prozess kann bis zum End-of-Life des Fahrzeugs nahtlos auf die Entwicklung von Software-Updates übertragen werden.

Darüber hinaus können Daten aus dem Feld zur Verbesserung des Simulationsprozesses genutzt werden. Die eingangs erwähnten Fahrzeugunikate können so ohne nennenswerten Aufwand mithilfe der Simulation getestet und freigegeben werden, bevor die Software auf das reale Fahrzeug übertragen wird.

2.3 Parallelisierung

Idealerweise bietet die beschriebene Umgebung auch Möglichkeiten für eine einfache Parallelisierung. Voraussetzung dafür ist, dass alle relevanten Komponenten wie Fahrzeug, Umgebung, virtuelle Steuergeräte, etc. wie bei der SiL-Simulation virtualisiert zur Verfügung stehen.

Wenn dies gegeben ist, können mit einem entsprechend hohen Parallelisierungsgrad sehr viele Tests in kurzer Zeit durchgeführt werden. Dies kann etwa durch skalierbare Rechenleistung und eine Automatisierung in der Cloud beziehungsweise mithilfe von Cloud-Technologien erfolgen [3].

Neben der eigentlichen Simulationsdurchführung sind automatisierte Auswertungen erforderlich. Aufgrund der hohen Simulationsanzahl werden sehr große Datenmengen erzeugt. Diese können eine vergleichbare Größe haben, wie die Datenmengen, die von realen Fahrzeugen im Feld aufgezeichnet werden. Eine übersichtliche Aggregation der im Zuge der Simulationen erzeugten sowie der gemessenen Daten ist unerlässlich.

Ähnlich zur DevOps-Schleife wird so ein sich wiederholender (automatisierbarer) Simulationskreislauf etabliert: Eine Software(-komponente) wird in den virtuellen Versuchsträger integriert, im Integrationstest überprüft und dann für eine Vielzahl von Tests (Simulationen) in der Cloud bereitgestellt. Die automatisiert ausgewerteten Ergebnisse der Simulationen werden anschließend zur Verbesserung der Software(-komponenten) genutzt, sodass der Kreislauf von neuem beginnt, siehe Abb. 2.

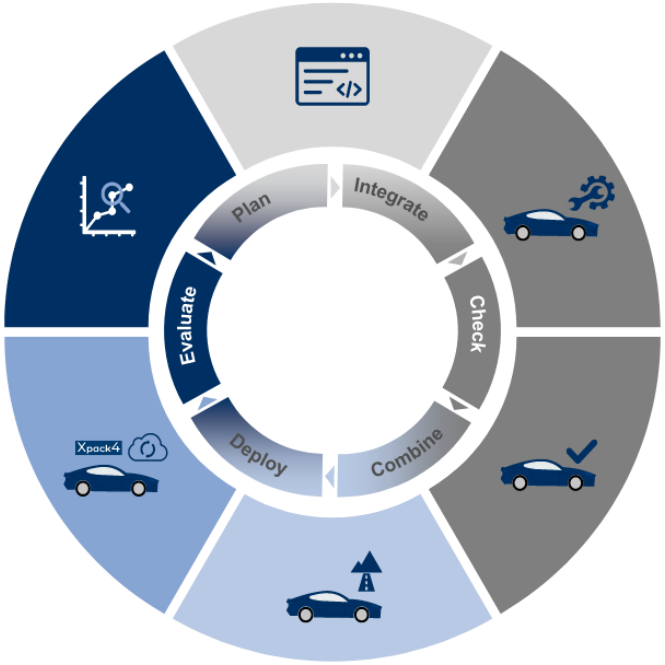


Abb. 2. Simulations-„DevOps“ als Grundvoraussetzung für die Parallelisierung

3 Praktische Anwendung der dargestellten Toolkette

Im Folgenden wird am Beispiel der Entwicklungs- und Testumgebung VIRTO eine Entwicklungsmethode skizziert, mit der den beschriebenen vielfältigen Herausforderungen der modernen Fahrzeugentwicklung begegnet werden kann. Als modulare App-Suite stellt VIRTO eine Sammlung einzelner Tools dar, die je nach Bedarf unabhängig voneinander oder als Gesamtwerkzeugkasten eingesetzt werden können, siehe Abb. 3.

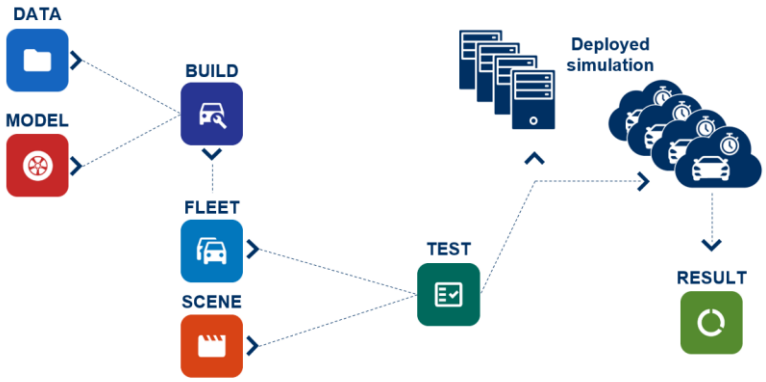


Abb. 3. Anwendungsübersicht der Entwicklungs- und Testumgebung VIRTO

Da in den meisten Unternehmen in der Regel bereits eine Vielzahl von Werkzeugen eingesetzt wird und individuelle Prozesse etabliert sind, bestand das vorrangige Ziel bei der Entwicklung darin, Lücken im Entwicklungsprozess zu schließen, um einen durchgängigen Testprozess zu ermöglichen. Entscheidend war, dass sowohl alle denkbaren Problemfelder berücksichtigt werden als auch eine flexible Anpassung an bestehende Entwicklungsprozesse realisierbar ist.

Die Entwicklungs- und Testumgebung ermöglicht die Umsetzung der genannten Anforderungen unabhängig von Art und Größe des Unternehmens. Durch eine konsistente Entwicklungsumgebung führt dies zu einem reibungslosen Daten- und Workflowmanagement für eine vollständig nachverfolgbare Simulation.

Nachfolgend wird anhand ausgewählter Anwendungsbeispiele gezeigt, wie sich die jeweiligen Prozessschritte vereinfachen lassen.

3.1 Reduzierung der Komplexität von Simulationen

Bei der Durchführung von Simulationen ist selten eine exakte Abbildung des realen Verhaltens erforderlich. Die Annahme, dass einfache Modelle grundsätzlich auch eine geringe Komplexität aufweisen, ist allerdings eine häufige Fehleinschätzung. Um genau abwägen zu können, welcher Detaillierungsgrad benötigt wird und welcher sich unmittelbar auf die geforderte Auswertung auswirkt, ist ein ausgeprägtes Systemverständnis erforderlich. Das modulare Zusammenspiel der verschiedenen Anwendungen bietet an dieser Stelle ein großes Unterstützungspotenzial. So liefert die integrierte Datenbanklösung wertvolle Einblicke in den jeweiligen Entwicklungsstand und erhöht so das Vertrauen in die Parameterdatenbasis.

Die Modellverwaltung hingegen steuert dokumentierte und verifizierte Submodelle unterschiedlicher Detaillierungsgrade aus den verschiedenen Fachbereichen [4] bei, während die Lösung für den Modellaufbau den automatisierten Aufbau mittels vordefinierter Konfigurationen übernimmt, die auf langjährig erprobter Simulationsmethodik basieren.

Auf diese Weise erstellte Fahrzeuge werden in einer virtuellen Flotte aus Gesamtfahrzeugen gesammelt. Dies erlaubt es, gezielt geeignete Fahrzeugmodelle für Tests zu verwenden, ohne sich mit der komplexen Modellierung und Parametrierung auseinandersetzen zu müssen. In Verbindung mit der Anwendung für den Modellaufbau kann diese Gesamtfahrzeugflotte stets aktuell gehalten werden, da der Aufbau automatisiert erfolgt, sobald neue Parameterdaten oder Modelle zur Verfügung stehen. Dies ermöglicht einen einfachen Zugriff auf ein zentrales, automatisiert gepflegtes Modell-Repository und damit erhebliche Synergien in der virtuellen Fahrzeugentwicklung.

Anhand einer aufgebauten Flotte unterschiedlichster Fahrzeugmodelle, die nach Anwendungsfällen aufgegliedert sind, unterstützt die Flottenverwaltung die gezielte Suche nach einem geeigneten Simulationsmodell für den jeweiligen Anwendungsfall.

3.2 Etablierung eines übergeordneten Workflows

Im Laufe des Entwicklungsprozesses erstellen verschiedene Fachbereiche zahlreiche Simulationsmodelle von Systemen oder Bauteilen. Einige der Simulationsmodelle werden innerhalb des Unternehmens, andere von Zulieferern entwickelt und bereitgestellt. Der bereichsübergreifende Austausch von Modellen oder Modelldaten (Parametern) ist, insbesondere in größeren Organisationen, häufig mit Schwierigkeiten verbunden. Darüber hinaus wird meist dem eigenen Modell ein höheres Vertrauen entgegengebracht als Modellen aus bereichsfremden Quellen.

Die Entwicklungs- und Testumgebung kann hier mit Anwendungen für die Modell- und Flottenverwaltung als wertvoller Multiplikator agieren, indem sie einen zentralen Speicherort für eine Vielzahl von Modellen bereitstellt. Dabei kann es sich sowohl um mechanische und hydraulische Subsystemmodelle als auch um Modelle der Steuergeräte (mit oder ohne modellierter Aktorik) handeln.

Die Anwendungen zur Testdurchführung und Ergebnisanalyse bilden außerdem die Eckpfeiler, um aus den Simulationsergebnissen Handlungsschritte abzuleiten. Die Simulationen können aus den aufgebauten Fahrzeugen aus der Flottenverwaltung und den Szenarien aus der Szenarienverwaltung individuell zusammengestellt werden.

Die Ausführung der Simulationen auf geeigneten IT-Infrastrukturen wie Cloud- oder HPC-Rechnern wird von der Anwendung für die Testdurchführung verwaltet. Diese Vorgehensweise ermöglicht jederzeit einen Einblick in die Ergebnisgenerierung. Die Ergebnisse werden in der Ergebnisverwaltung gespeichert und visualisiert.

Diese Durchgängigkeit bringt auch den großen Vorteil der Nachvollziehbarkeit mit sich, da dem Ergebnis entnommen werden kann, welche Ausgangsdaten der Simulation zugrunde lagen. Damit leistet die Entwicklungs- und Testumgebung einen wesentlichen Beitrag zur Erhöhung der Skalierbarkeit und ermöglicht es den Systementwickler*innen, sich auf die Innovation zu konzentrieren.

4 Zusammenfassung & Ausblick

Da sich Software zum entscheidenden Differenzierungskriterium im Fahrzeug entwickelt hat, sind ganzheitliche Lösungen zur Beherrschung der damit verbundenen Komplexität gefragt. Das vorrangige Ziel ist es, alle notwendigen Testfälle abzudecken und die Entwicklungszeiten deutlich zu verkürzen. Die Entwicklungs- und Testumgebung VIRTO wurde speziell für diese Anforderungen entwickelt und setzt dabei auf gängige Workflows, sodass Lücken im Entwicklungsprozess gezielt geschlossen werden können und eine durchgängige Lösung entsteht. Dies führt zu einer Optimierung bestehender Arbeitsabläufe, während die durchgängige Entwicklungsumgebung eine nachvollziehbare Simulation gewährleistet.

Das Ziel sollte eine weitgehend automatisierte Validierung durch die Simulationsdurchführung sein, insbesondere von sicherheitsrelevanten Fahrfunktionen. Die durchgängige Nachverfolgbarkeit und Nachvollziehbarkeit sind wichtige Merkmale, um zukünftige, rein simulationsbasierte Freigaben neuer Softwareversionen zu ermöglichen. Die Anbindung an Requirements-Managementsysteme sowie die erweiterte Einbindung der beschriebenen, automatisierten Prozesse in den erweiterten Kontext des Produktentwicklungsprozesses können künftig zu einer weiteren Steigerung der Entwicklungseffizienz und letztendlich der Produktqualität führen [5, 6].

5 Literaturverzeichnis

[1] SAE International, “The key to autonomous vehicle safety is ODD”, [Online]. https://www.sae.org/news/2019/11/odds-for-av-testing, 2019, letzter Zugriff 05.08.2024

[2] Erz, Jannis; et al., "Towards an Ontology That Reconciles the Operational Design Domain, Scenario-based Testing, and Automated Vehicle Architectures," 2022, *2022 IEEE International Systems Conference (SysCon)*, Montreal, QC, Kanada, S. 1-8, doi: 10.1109/SysCon53536.2022.9773840.

[3] Niederbrucker, Gerhard., et al., „Clouds Ahead – The Transformation of Vehicle Development and Data Management Processes”, 2021, In: Bertram, T. (eds) Automatisiertes Fahren 2020. Proceedings. Springer Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-34752-9_15

[4] ISO 11010-1:2022-04, Straßenfahrzeuge - Personenkraftfahrzeuge – Klassifizierung Simulationsmodelle - Teil 1: Fahrdynamik

[5] Gebhard, Bernd and Rappl, Martin, "Requirements Management for Automotive Systems Development," SAE Technical Paper 2000-01-0716, https://doi.org/10.4271/2000-01-0716, 2000

[6] Dick, Jeremy.; Hull, Elizaeth; Jackson, Ken:. Requirements Engineering. https://doi.org 10.1007/978-3-319-61073-3, 2017

Entwicklung und prototypische Umsetzung von automatisierten Testverfahren nach AFGBV

Trautmann, Toralf[1]; Mendt, Franziskus[2]; Irrasch, Tom[1]; Blenz, Konstantin[3]

[1] University of Applied Sciences Dresden (HTW), Friedrich-List-Platz 1, 01069 Dresden, Germany
toralf.trautmann@htw-dresden.de
[2] ZAFT e.V. an der HTW Dresden, Friedrich-List-Platz 1, 01069 Dresden, Germany
franziskus.mendt@htw-dresden.de
[3] TraceTronic GmbH, Stuttgarter Str. 3, 01189 Dresden, Germany
konstantin.blenz@tracetronic.de

Kurzfassung: Mit der „Autonome-Fahrzeuge-Genehmigungs-und-Betriebs-Verordnung – AFGBV“ sind die Randbedingungen für die Typzulassung und den täglichen Betrieb autonomer Fahrzeuge geregelt. Das Problem besteht aktuell in der technischen Umsetzung der Verordnung für ausgewählte Nutzungsszenarien. Eine besondere Herausforderung stellt hierbei die Durchführung der in §13 Absatz 7 geforderten täglichen erweiterten Abfahrkontrolle dar. An der Hochschule für Technik und Wirtschaft Dresden (HTWD) erfolgt in Zusammenarbeit mit der tracetronic GmbH die technische Umsetzung dieser Abfahrkontrolle anhand ausgewählter Einsatzszenarien. Ziel ist es, einen Handlungsleitfaden für Hersteller, Prüforganisationen und Dienstleister zu entwickeln. Besonderes Augenmerk gilt dabei der zeitsynchronen Aufzeichnung und dauerhaften Speicherung der erhobenen Daten. Diese unterscheiden sich je nach Anwendungsfall erheblich und sind besonders für die Klärung der Schuldfrage im Falle eines Unfalls von grundlegender Bedeutung. Auf Basis eines prototypischen Aufbaus des Testfeldes und der Laboreinrichtung an der HTWD erfolgt die Bewertung der Praxistauglichkeit dieses Handlungsleitfadens in den Bereichen Planung des Testablaufs, Steuerung der Testdurchführung und Protokollierung sowie die Identifikation fehlender Komponenten.

1 Einleitung

Autonome Fahrzeuge werden in den nächsten Jahren in unterschiedlichen Ausprägungen die individualisierte Mobilität dramatisch ändern. In Deutschland sind solche Fahrzeuge seit dem 24.06.2022 zulassungsfähig ([SVG24] und [AFG24]). Die entsprechende Verordnung (Autonome-Fahrzeuge-Genehmigungs- und- Betriebs-Verordnung - AFGBV) regelt dabei sowohl die Zulassung der Fahrzeuge als auch die Anforderungen an den täglichen Betrieb. Solche Fahrzeuge stellen die aktuell höchste Stufe der Automatisierung dar (Stufe 5) und sind ohne Präsenzfahrende im Zulassungsbereich nutzbar. In der Verordnung sind zahlreiche Vorgaben sowohl für die Typzulassung als auch den sicheren Fahrbetrieb vorgegeben. Die HTWD verfügt mit dem Prüffeld für automatisierte Fahrfunktionen über eine technische Einrichtung, die eine Entwicklung und den Test entsprechender Fahrfunktionen in einem realen Versuchsfahrzeug ermöglicht.

Im Projekt ist zunächst die Realisierung einer prototypischen täglichen Betriebskontrolle für ein umgebautes Versuchsfahrzeug (BMW i3) vorgesehen. Nachdem diese für ausgewählte Funktionen realisiert wurde, ist der Testbetrieb für einen Shuttle-Service vorgesehen. Hierfür soll in Kooperation mit der FSD Fahrzeugsystemdaten GmbH entweder für dieses Fahrzeug oder ein anderes Fahrzeug die Zulassung nach AFGBV für einen eingeschränkten Betriebsbereich erfolgen. Auf Grund der Lage der HTWD bietet sich hierfür eine Fahrt vom Prüffeld zum Dresdner Hauptbahnhof an. Damit kann ein praxisrelevanter Test auch im Rahmen von Lehrveranstaltungen erfolgen.

2 Anforderungen der AFGBV

In der Verordnung werden alle Bereiche des Betriebes von autonomen Fahrzeugen geregelt. Dies beginnt bei der Typzulassung, die jetzt auch die festgelegten Betriebsbereiche umfasst, und endet bei den Anforderungen und Pflichten von Hersteller, Halter und Technischer Aufsicht. Besonders die umfangreichen Anforderungen an den Halter (§ 13) sind für den sicheren Betrieb von großer Bedeutung. Dort heißt es beispielsweise (Zitat):

> *(7) Die erweiterte Abfahrkontrolle im Sinne des Absatzes 1 Nummer 2 beginnt mit einer Probefahrt, um die Systeme zu aktivieren. Im Anschluss an die Probefahrt werden folgende Bereiche überprüft:*
> 1. *Bremsanlage,*
> 2. *Lenkanlage,*
> 3. *Lichtanlage,*
> 4. *Reifen/Räder,*
> 5. *Fahrwerk,*
> 6. *sicherheitsrelevante elektronisch geregelte Fahrzeugsysteme sowie die Sensorik zur Erfassung externer und interner Parameter und*
> 7. *mechanische Fahrzeugsysteme für die aktive und passive Sicherheit.*

Es gibt keine weiteren Aussagen zur Parametrierung der jeweiligen Systeme, beispielsweise eine Mindestfahrgeschwindigkeit oder eine Objektkonstellation für Umfeldbedingungen. Im Rahmen des Projektes sollen daher an verschiedenen Beispielsituationen der technische und zeitliche Aufwand sowie die Mindestmenge an dauerhaft zu speichernden Informationen ermittelt werden. Mit der Erweiterung der Software ***ecu.test*** der Firma tracetronic GmbH soll dann ein Werkzeug zur Planung, Durchführung und Verwaltung für den professionellen Einsatz zur Verfügung gestellt werden.

3 Konzeption

Bei der Konzeption des Projektes war zu berücksichtigen, dass nicht allein der Forschungsgegenstand (AFGBV-Vorgaben) betrachtet werden muss, sondern die neuen Testfälle- und verfahren auch für aktuelle Serienentwicklungen direkt eingesetzt werden können. Zudem ist eine Nutzung in der Lehre seitens der HTWD erforderlich, da nur so ausreichend Zeiten für die praktische Erprobung zur Verfügung stehen. Der grundlegende Ablauf wurde daher in dieser Weise in Absprache mit den Projektpartnern festgelegt:

- Start im Werkstattbereich des Fahrzeug-Technikums,
- Durchführung einer oder mehrerer Versuche auf dem HTWD-Prüffeld,
- Rückfahrt ins Technikum zur Durchführung der weiteren Tests.

Als eine besonders gut geeignete Funktion hat sich dabei eine automatische Notbremsung auf ein stehendes oder bewegliches Zielobjekt (Person) herausgestellt. Eine ausführliche Darstellung zur Relevanz im Bereich automatisierter Shuttle-Busse (Projekt RABus) und zur systematischen Bewertung solcher Funktionen findet sich in [GAO23]. Insbesondere in Verbindung mit einem Fußgängerüberweg gibt es hier verschiedene Szenarien mit hoher praktischer Relevanz. Bei der Kollision mit einem Cruise-Robotertaxi im Jahr 2023 war eine ähnliche Situation letztlich der Auslöser für die nachfolgende Einstellung der Erprobungsfahrten [KOO24]. Im Projekt sollen daher 4 Varianten realisiert und detailliert untersucht werden. Diese sind in Abb. 1 dargestellt.

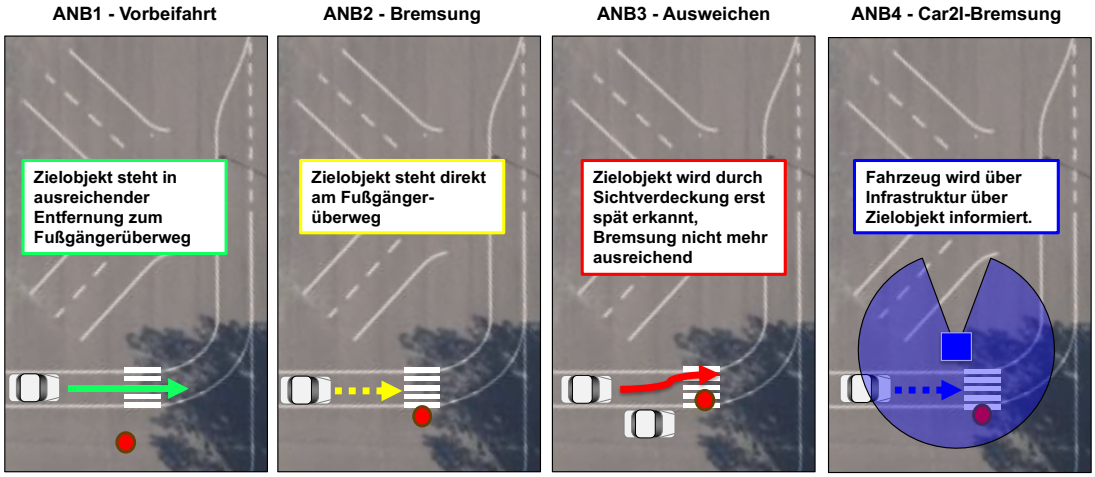


Abbildung 1: Ausgewählte Testfälle für die Funktion Automatische Notbremse (ANB)

An einem Fußgängerüberweg kommt es dabei nicht nur auf die Kollisionsvermeidung an, sondern ebenso auf die sichere Interpretation der Fußgängerintention. Autonome Fahrzeuge werden auf Sicherheit hin optimiert werden. Dabei kann die Situation entstehen, dass ein Fußgänger zwar direkt am Überweg steht, trotzdem aber nicht losläuft. Hierfür müssen sichere Lösungen gefunden werden, z.B. eine langsame Weiterfahrt mit Sicherheitsabstand (seitliches Ausweichen bei Möglichkeit) oder eine eindeutige Signalisierung für den Fußgänger. Zusätzlich zu diesen fahrzeugautarken Auslösungen ist eine Veh2I-Kommunikation mit einer Verkehrsüberwachung vorgesehen. Hierzu ist eine mobile Einheit der Firma NewSense Engineering GmbH (München) vorhanden, die zukünftig über ein WLANp-Netzwerk relevante Objektdaten an das Versuchsfahrzeug übermitteln wird.

4 Technische Umsetzung

Die Basis des Prototyps bildet ein umgebauter BMWi3. Durch einen Zusatzmotor am Lenkgetriebe erfolgt die Querführung des Fahrzeugs. Für die Längsführung im Komfortbereich wurde das Fahrpedal elektronisch manipuliert. Durch Rekuperation sind so Beschleunigungen bis etwas a_L = -2,0 m/s^2 möglich. Eine Notbremsung kann durch einen Manipulator am Parkbremsschalter erfolgen. Dieser Aufbau bietet im Gegensatz zur Busmanipulation die Möglichkeit der Nutzung in anderen Serienfahrzeugen. Der Signalflussplan inklusive des verwendeten Lidars (für ANB und Lokalisierung) ist in Abb. 2 dargestellt.

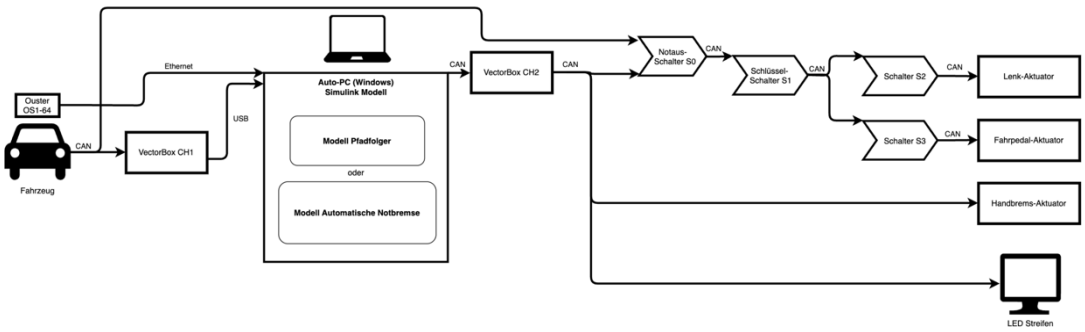


Abbildung 2: Signalflussplan des Versuchsträgers (BMW i3, IRR24)

Da das Fahrzeug eine Aussnahmegenehmigung für Stufe 3 und 4 für Sachsen besitzt, ist eine Notabschaltung integriert. Ein zusätzlicher LED-Leuchtstreifen kann sowohl für die Insassen oder auch für umgebende Personen zur Information genutzt werden. Verschiedene Ausprägungen werden vor allem im Rahmen der Veranstaltungen im Lehrgebiet Kfz-Mechatronik (Prof. Toralf Trautmann) implementiert und im Realversuch getestet. Verschiedene Beispiele für bereits erfolgte Realisierungen finden sich auf dem youtube-Kanal des Instituts (youtube: MechLabDE).

Für die erste Stufe der Realisierung wurde zur Querführung ein Pfadfolge-Regler genutzt. Da aktuell noch keine echtzeitfähige Positionierung über den Lidar erfolgen kann, wird die aktuelle Position nur auf Grund der Fahrzeug-Eigenbewegungssensorik ermittelt. Dies reicht aber aus, um die in Abb. 3 dargestellt Runde auf dem HTWD-Prüffeld mit $v < 10$ km/h abzufahren. Voraussetzung ist aber hierfür eine exakte Startpositionierung im Garagenbereich und eine genau bekannte Beladung.

Nach der Einfahrt in den Garagenbereich sind verschiedene automatische Tests vorgesehen, die einen besonderen Bezug zur periodischen Hauptuntersuchung haben. Für die FSD Fahrzeugsystemdaten GmbH (Dresden) wurde beispielsweise eine Methode entwickelt, um bei der Einfahrt eines zu prüfenden Fahrzeugs die Scheinwerfereinstellung effektiv zu ermitteln [DEG20]. Solche Testverfahren eignen sich sehr gut zur Integration in den vorgesehenen Ablauf.

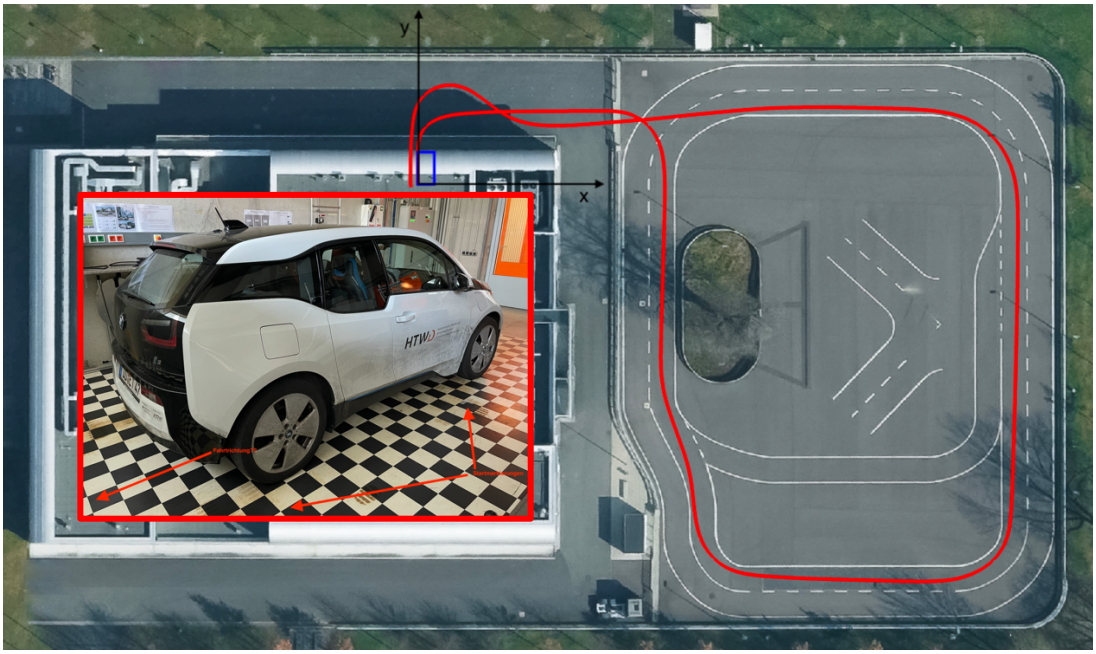


Abbildung 3: Beispiel für eine automatisierte Testfahrt (IRR24).

5 Erweiterte Untersuchungen und Einbindung der Steuerungssoftware

Neben dem BMW i3 als autonomem Fahrzeug soll die konzipierte Steuerungssoftware auch für automatisierte Tests an anderen Versuchsfahrzeugen eingesetzt werden. Daher wurde die ANB-Eigenentwicklung auf Basis von Lidar-Sensorik in einem modularen Aufbau mit der Software Matlab/Simulink erstellt. Hierdurch ist es sehr einfach möglich, Algorithmen je nach Anwendungsfall zu wechseln (z.B. eine einfache Erfassung für Praktikumsversuche) und die Steuerung über ***ecu.test*** auszuführen. Die Vernetzung ist in Abb. 4 dargestellt.

Die umfangreiche Vorverarbeitung der Lidar-Daten erfolgt dabei auf einem zusätzlichen CAR-PC. Für bestimmte Modell (Velodyne oder Ouster) können dabei die in Matlab implementierten Treiber genutzt werden. Für andere Modelle erfolgt die Vorverarbeitung in einem ROS2-Knoten. Die Anbindung an ein Simulink-Modell ist dann ebenfalls durch die ROS2-Unterstützung möglich.

Ein wichtiger Untersuchungsschwerpunkt im Bereich der Lidar-Sensorik ist das Verhalten bei schlechten Witterungsverhältnissen. Bedingt durch das Funktionsprinzip kann es zu funktionalen Einschränkungen kommen. Erste systematische Untersuchungen haben aber zumindest für statische Situationen keine gravierende Änderung aufgezeigt [TRA24]. Mit dem neuen Versuchsaufbau sind solche Untersuchungen nun auch dynamisch möglich. Aus den Ergebnissen kann dann abgeleitet werden, ob eine aufwendige Witterungssimulation (z.B. Regen) für eine tägliche Abfahrkontrolle zwingend notwendig sein wird.

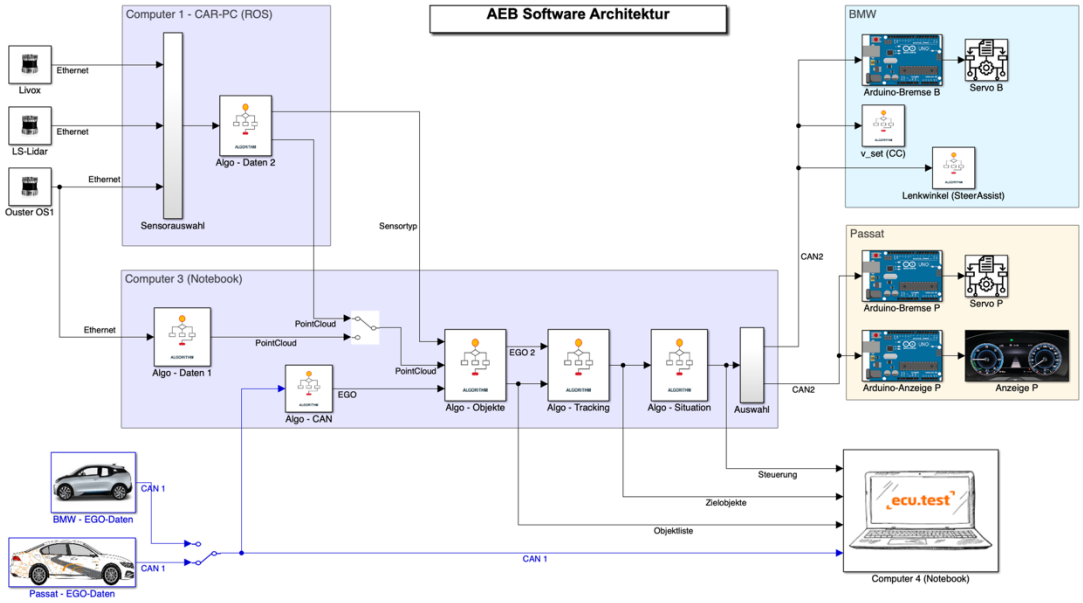


Abbildung 4: Signalflussplan des modularen Prototyps.

Der grundlegende Ablauf für einzelne Tests ist folgendermaßen geplant:

- 3D-Simulation des Ablaufs inklusive Umfeldsensorik zur Ermittlung der notwendige Parametrierung,
- Durchführung der Versuchsfahrten mit Messdatenaufzeichnung,
- Auswertung und Protokollierung,
- Abgesicherte Ablage der relevanten Daten.

Besonders wichtig ist in diesem Zusammenhang die passende Auswahl der für die jeweilige Situation relevanten Daten. Gerade beim Einsatz von umfangreicher Umfeldsensorik treten sehr große Datenmengen auf. Es muss daher eine selektive Auswahl erfolgen, um bei möglichst geringem Speicherbedarf die sichere Funktionalität der sicherheitskritischen Systemen bei Bedarf nachweisen zu können. Neben den Entwicklungsbereichen der Fahrzeughersteller ist für die Ausgestaltung der Entwicklung auch die Mitarbeit von Kfz-Sachverständigen vorgesehen.

In der gegenwärtigen Startphase ist immer die Anwesenheit einer Präsenzperson vorgesehen. Im weiteren Projektverlauf wird diese durch teleoperierte Eingriffe ersetzt.

6 Zusammenfassung und Ausblick

An der Hochschule für Technik und Wirtschaft Dresden (HTWD) entsteht in enger Zusammenarbeit mit der tracetronic GmbH ein Testfeld für autonome Fahrzeuge. Im Fokus steht dabei die Ableitung praxisrelevanter Abläufe für die vorgeschriebene tägliche Abfahrkontrolle. Ein modularer Aufbau erlaubt dabei auch die Nutzung einzelner Testelemente für Fahrzeuge mit geringerer Automatisierungsstufe.

7 Literaturverzeichnis

[SVG24] https://www.gesetze-im-internet.de/stvg/StVG.pdf (Abruf: 01.09.2024)

[AFG24] https://www.gesetze-im-internet.de/afgbv/__1.html (Abruf: 01.09.2024)

[GAO23] Gao, K., Weinrich, U., Riemer, T., and Reuss, H.-C., "Technical Evaluation of the Obstacle Detection for Automated Shuttle Buses" SAE Technical Paper 2023-01-1227, 2023

[KOO24] Philip Koopman, "Lessons from the Cruise Robotaxi Pedestrian Dragging Mishap" Preprint: Accepted by IEEE Reliability Magazine, 2024

[IRR24] Tom Irrasch, "Entwicklung eines Funktions-Prototypen für die erweiterte Abfahrkontrolle autonomer Fahrzeuge laut AFGBV" Diplomarbeit HTW Dresden, 2024

[DEG20] Degenkolbe, Marcus; Trautmann, Toralf, "Prüfsystem zum Prüfen eines Scheinwerfers eines Fahrzeugs und Verfahren davon" Deutsche Patentanmeldung DE102019111210A1, 2020

[TRA24] Trautmann, Toralf, "Prüfsystem zum Prüfen eines Scheinwerfers eines Fahrzeugs und Verfahren davon" Deutsche Patentanmeldung DE102019111210A1, 2020

[NSE24] https://www.newsense-engineering.de (Abruf: 01.09.2024)

A Method and a Set of Tools for Automated Evaluation and Aggregation of Test Data

Tim Martini[1]✉, André Kempf[2], Michael Auerbach[1], André Casal Kulzer[3]

[1]Labor Fahrzeugantrieb,
Hochschule Esslingen,
Kanalstraße 33, 73728 Esslingen a.N.
{tim.martini, michael.auerbach}@hs-esslingen.de

[2]Entwicklung Betriebsstrategie,
Mercedes-Benz AG,
Mercedesstraße 120, 70372 Stuttgart
andre.a.kempf@mercedes-benz.com

[3]Lehrstuhl Fahrzeugantriebssysteme,
IFS Universität Stuttgart,
Pfaffenwaldring 12, 70569 Stuttgart
andre.kulzer@ifs.uni-stuttgart.de

Abstract: Faced with the desired shift-left concept in automotive development, it is essential to automate and digitise the development process. It is common practice in automotive software engineering to implement DevOps processes, such as continuous testing. In the context of the off-car development method, extended data from test drives and in-service operations are utilised as stimuli for continuous off-car robustness testing. The robustness testing complements regression testing by using the exploratory nature of road tests. This allows the software to be analysed in a real-world operational context at an early stage in the development process. The implementation of automated road testing in simulations and on test benches enables the execution of a significant number of measurements, the evaluation of which can prove to be a challenge. In addition to the assessment of functional tests, other characteristics, such as the quantity of the change of states and the availability of electrical energy, are also relevant for behaviour of a performance-hybrid vehicle in real-world conditions. This work presents a method and tools for automated evaluation and aggregation of data obtained from off-car testing. Object-oriented approaches and interactive dashboards are employed for data and code management, as well as for visualisation purposes. An efficient analysis is enabled, supporting data-driven development. The use of software-in-the-loop simulations and the implementation of analysis classes ensure a consistent evaluation throughout the product development process in different development environments.

1 Research Motivation and the Challenge of Off-Car Testing

The complex nature of control systems in modern vehicles, which involve a multitude of actors and interactions, requires a comprehensive and intensive testing process. As it is not feasible to test all operational conditions, the software system’s functionality can only be proven within a covered range [Wo18, My12]. Intensive testing of the system reduces the likelihood of errors occurring during in-service operation. Road testing, which is a highly resource-demanding process, is of pivotal importance. In addition to functional tests, the vehicle's behaviour is tested subjectively and exploratively under varying boundary conditions. However, due to shorter development cycles and limited availability of test vehicles, there is diminishing time available for road testing. Therefore, it is essential that the vehicle software reaches a high level of maturity before the road testing phase in order to ensure an efficient use of time. The

goal is to continuously increase the software's maturity throughout the product development process (PDP). [Ma22]

In order to guarantee this, the continuous integration, deployment, and testing processes, as defined by the DevOps methodology [HPJ20], will be employed. A new software or calibration dataset is tested against defined requirements. To extend the test coverage and subject the evolving system to exploratory analysis and error investigations at an early stage of its development, off-car robustness testing is implemented as part of the off-car development method (OCM) in the PDP [Ma22].

The OCM employs data derived from both test drives and in-service operation to emulate the conditions of road tests in simulations and on test benches. Published works on this topic employ Markov chains to aggregate speed measurement reports from test drives into compressed, equivalent advance profiles [FIG20]. By emulating road tests, the system behaviour can be evaluated in a non-road environment, subjected to comprehensive analysis, and adapted to real-world operational requirements. The off-car test (OCT) used in this paper is a 2000-kilometre road test comprising of 100 individual trips varying in length. The total distance of OCTs and, consequently, the number of individual trips can be scaled as required. A probabilistic approach to the configuration of the trips in question ensures that a different OCT is conducted if the software or calibration dataset is modified.

The accumulation of data through the fast-moving nature of continuous off-car testing presents a challenge for analysis. A time series based interpretation of results requires a significant amount of resources for this quantity of data. To capture insights and identify potential avenues for enhancing system behaviour, it is essential to employ efficient methods.

This work proposes a solution to the challenge of off-car testing, namely a method and tools for automated data evaluation and aggregation. In order to automate and standardise the evaluation of data across the PDP, an object-oriented approach to data and code management has been integrated into the tool chain (see section 2). The tools have been designed with the specific purpose of aggregating vehicle‘s internal measurement signals, calibration data, and calculation instructions for the usage of evaluation and visualisation. In accordance with user predefined calculation instructions, the results of the OCTs are presented in the form of interactive dashboards, accompanied by the calculation of key figures, as described in section 3. The paper concludes with a comparison of two calibration datasets. The examples are demonstrated through software-in-the-loop (SiL) simulations [SZ16, Sy24] and a control system for the change from electric to hybrid drive of a plug-in performance hybrid vehicle.

2 Consistent Evaluation and Aggregation: Data and Code Management

The reusability of calculation procedures and the aggregation with data are of fundamental importance for efficient and error-free analysis of the evolving vehicle software. In order to implement these principles, object-oriented programming (OOP)

in Python [St18] is employed. One of the core concepts of OOP is the definition of classes, which serve as blueprints for instances. The classes designed for the OCM describe methods for the evaluation and visualisation of measurements. Measurement-specific data are transferred to an instance as attributes, which may be single or multi-measurement data. The following pseudo-code outlines the structure and the functionalities of an example analysis class utilised in this paper. [St18]

Pseudo-Code: Analysis Class for Automated Evaluation and Visualisation of Electric Drive Characteristics
Input: Measurements in diverse formats from a multitude of development environments
Return: Measurement-specific instance with the class functions as implemented methods

```
1: Class Analysis:
2:         Function Initialise Instance():
3:                 Assign measurement(s) to instance
4:         Function Read Time Series():
5:                 Process diverse data formats
6:                 Save defined signals in a standardised data format
7:         Function Evaluate Electric Drive Characteristics():
8:                 Calculate the number of starts of the internal combustion engine per trip
9:                 Calculate the minimum value of the state of charge of the traction battery per trip
10:        Function Visualise Electric Drive Characteristics():
11:                Plot the quantity of starts of the internal combustion engine as a histogram
12:                Plot the minimum values of the state of charge of the traction battery as a histogram
13:        Function Read Calibration():
14:                Save calibration parameters of selected functions of the vehicle software
15:        …
```

When an OCT is conducted as a SiL simulation, an analysis instance is generated for each individual measurement. The analysis class may be calculated in simulations or, alternatively, posterior instances may be assigned to measurements. In addition to the storage of time series data from the control device, signals and key figures (e.g., lines 8 and 9 of the analysis class) are calculated to analyse the system. Furthermore, calibration parameters of selected functions are read and assigned to an instance. A single result file is created, comprising measurement-specific instances. These instances contain the time series data of the control devices, the calculated signals and key figures, the employed calibration data, and the methods utilised for the evaluation and visualisation of the measurement data and calibration parameters. The file is stored in Pickle format [Py24] for archiving.

In addition to its utility in SiL environments, OOP has the potential to enhance efficiency and reduce error probability in a multitude of development environments. The analysis class may be employed in a variety of additional development applications. The processing of diverse data formats (see pseudo-code, line 5) and the capacity for posterior assignment allows measurements to be generates from any environment as instances. The evaluation of simulations, test benches and vehicle measurements remains consistent. In the OCM framework, the above analysis class is also employed to analyse measurements documented within a big data infrastructure [IS23]. Alongside testing, the signals and key figures of OCTs can be used to determine target functions and facilitate automated calibration (see [Ma22]). Figure 1 provides an overview of the various development applications of the analysis class in the PDP.

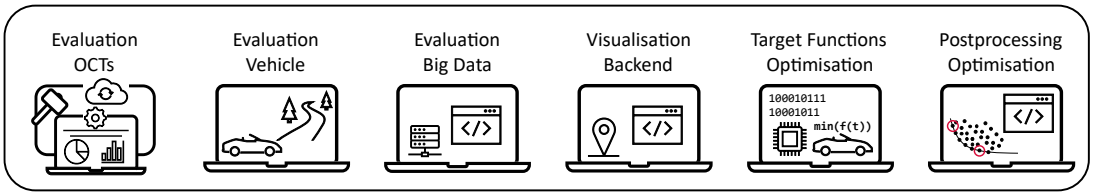


Figure 1: Overview of various development applications and environments of the analysis class in the PDP

Following the successful execution of an OCT, the only prerequisites for analysis are the Pickle file and a Python installation. All components for visualisation in interactive dashboards and the calculation of key figures are contained in the result file. This enables efficient and error-free analysis of the software under development.

3 Visualisation and Key Figures for Data-Driven Decision Making

Analysing large amounts of complex and fast-moving data, such the output from an OCT during vehicle software development, is a challenge. Visual preparation is a key to manageability and the integration of dashboards enables interaction with the data. This supports the search for errors and their causes while allowing for data-based software adjustments and calibration. [De24]

Using a set of analysis class methods, it is feasible to create a dashboard that is customised to a specific function by integrating the user predefined components, such as the histograms described in pseudo-code lines 11 and 12. A summarised interactive visual representation of data enables efficient comprehension of system behaviour and eases the investigation of anomalies [De24]. Figure 2 schematically illustrates the structure of a dashboard designed for off-car robustness testing of the control system that is switching between electric and hybrid drive. The dashboards are built using the Python library Dash [Pl24a]. The data is based on 100 individual trips with a total distance of 2,000 kilometres, which represents the intended real-world operation of the system under development.

The dashboard shows a list of measurements performed with information on each individual trip. The table list of measurements offers the possibility of filtering, for example, by the dynamics of a trip in the form of the relative positive acceleration index (RPA, [Tu15]). Furthermore, the dashboard updates the displayed graphs and calculations according to the filtered trip selection. In addition to histograms and key figures, the diagrams also display the associated calibration maps and corresponding operating points.

Time series data can be analysed by employing a predefined method of the instance using the Python library ASAMMDF [Hr23]. Measurements can be exported as MDF files [AS24] and analysed with common tools. The export can be initiated from the dashboard via a download button or through command line. Alternatively, interactive time series plots for selected signals can be generated. Using the Python library Plotly [Pl24b], the analysis class can deliver plots as an HTML file [Wo24].

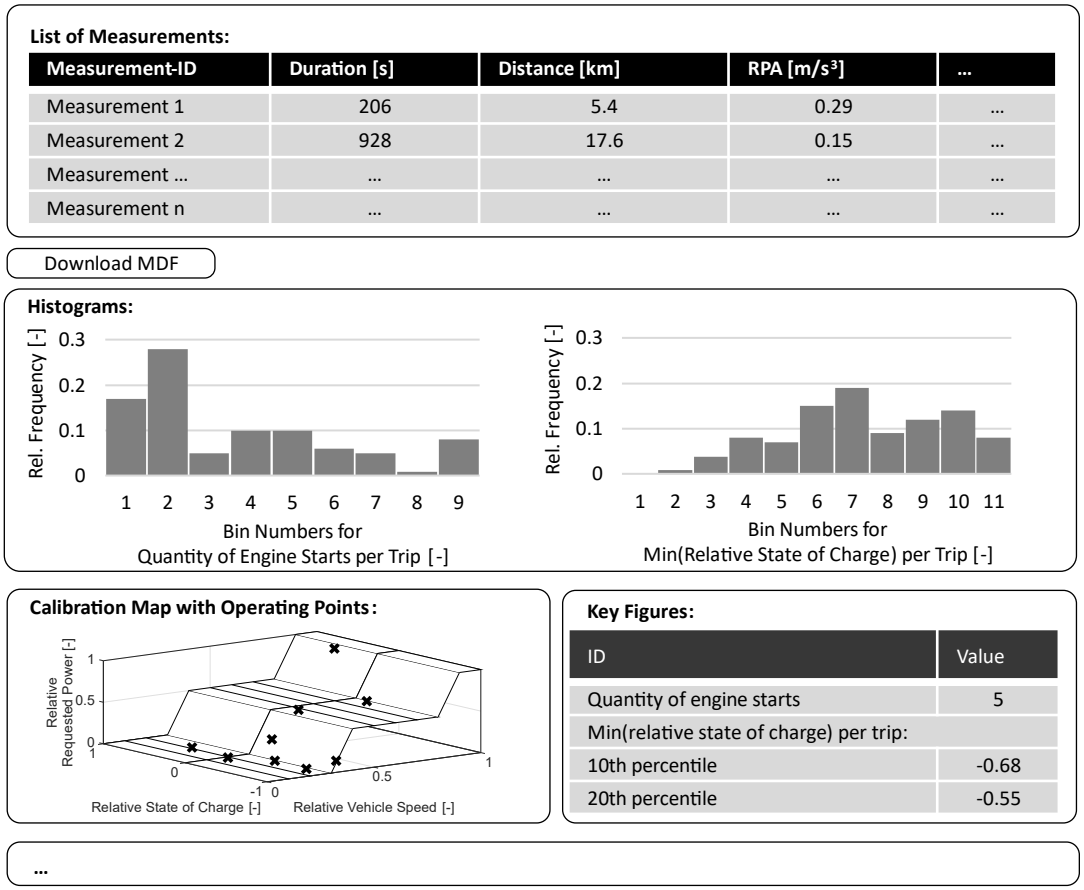


Figure 2: Schematic dashboard for analysing the control system for the change from electric to hybrid drive

The previously mentioned open-source Python libraries along with the Pickle module provide a platform for the analysis of a multitude of simulations and test bench measurements. Upon generation of new software or datasets, an analysis of the implemented modifications can be conducted based on the intended real-world operation within a few hours. To illustrate the power of the aforementioned method and tools, the results of an example variation of calibration data is presented in the following section.

Demonstrating Method and Tools: Results of an Example Off-Car-Test

The calibration of the control system in charge of changing drive states from electric to hybrid has a significant impact on the subjective perception of vehicle behaviour in performance hybrids. The degree of freedom of operation permitted by the drive train allows for a multitude of variants to be implemented. The calibration determines the utilisation of electrical energy stored in the traction battery and the extent of the change of states. The following section presents a comparison of two example calibration datasets and examines the resulting system behaviour as in-service operation. The example describes the results of an OCT for the charge-sustaining operation [UNR154] of a performance hybrid vehicle. The modified calibration map is shown in terms of the requested power as function of the vehicle speed and the state of charge of the traction battery.

Both variants concentrate on binary operation, divided into two speed ranges. In order to prevent the internal combustion engine from operating at an inefficient partial load, the low speed range is usually run in an all-electric mode. At higher speeds, a hybrid operation is employed to facilitate recovery of electrical energy through load point shifting. Moreover, the avoidance of partial load operation is extended by boundary conditions, namely performance potential and comfort. The differentiation between the variants is based on the targeted separation speed for the binary mode ranges. Calibration dataset A is designed with performance in mind, with the objective of maintaining sufficient electric energy for accelerations in the traction battery. This is achieved by starting the internal combustion engine at speeds of 30 kilometres per hour. Calibration dataset B is designed to provide a more comfortable system behaviour with less frequent changes of states than dataset A. This requires a system to reduce the number of internal combustion engine start and stop events in urban areas. The targeted engine start-up speed is therefore defined as 50 kilometres per hour.

In order to quantify the comfort of the system, the number of starts of the internal combustion engine is analysed. The key figure for the performance potential is described by the 10th percentile ($x_{0.1}$) of the lowest occurring state of charge of the traction battery per trip, thereby representing the minimum electrical energy that can be made available in 90 percent of cases. To calculate these values, OCTs of each variant are simulated in SiL environments. Based on the method discussed above and the tools for automated evaluation and aggregation of data, a pickle file is available after a few hours of simulation time.

Once the pickle file and a Python installation have been prepared, the results can be analysed. Figure 3 illustrates components of the analysis class used for the creation of dashboards. The left-hand side of the plot displays the comfort quantification, represented in a histogram showing the specific number of start-ups of the internal combustion engine per trip. The histogram on the right-hand side of the plot represents the lowest charge states of the traction battery per trip, which quantifies the performance potential. The state of charge is normalised to an example charge-sustaining set point for the purpose of comparing variants in this paper. The class centres for categorising the number of starts are arranged in an arithmetic sequence with a difference of two, ranging from zero to 20 starts. The relative state of charge exhibits a class step size of 0.1 within an interval of minus one to zero.

Comparing calibration dataset A to B, there is a shift towards fewer starts per trip and an increase in trips where the internal combustion engine is not started at all. The proportion of time spent in all-electric driving mode is 2.25 times higher with calibration dataset B compared to dataset A (see the left plot, bin zero). Overall, the number of starts reduces by 22 percent. An increase in the proportion of electric driving results in a greater demand for electric energy from the traction battery, which has two consequences in real-world operations. Firstly, the 10th percentile drops to -0.68 with calibration dataset B compared to the calibration dataset A value of -0.55 for the minimum relative state of charge. Secondly, in dataset B, the value of -0.55 corresponds to the 20th percentile ($x_{0.2}$). Therefore, with dataset B, the number of trips with the same minimum available electric energy is 10 percent lower than with dataset A.

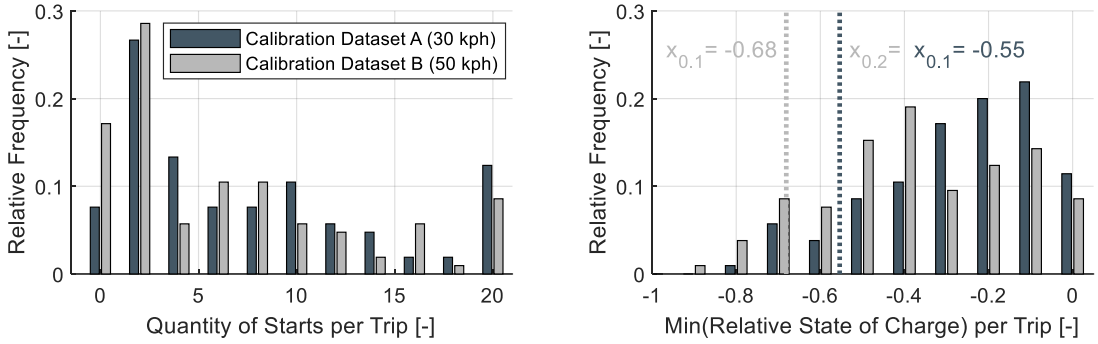


Figure 3: Quantification of the characteristics of the dataset variants in real-world operation – comfort (left) and performance potential (right)

This initial assessment serves to comprehend the potential advantages and disadvantages of the proposed modifications at a glance. A plan of action for subsequent steps can be devised based on the quantified characteristics. In this example, calibration dataset B results in a reduction in performance potential in comparison with dataset A. However, it also demonstrates the potential to reduce the quantity of change of states. To investigate a potential trade-off between comfort and performance, it is recommended to employ the interactive dashboards for a more comprehensive analysis. Based on the insights gained, modifications can be made to the calibration data as appropriate.

The off-car testing equivalent to in-service operations enables the quantification of potential advantages and disadvantages of modifications prior to costly road tests. The aggregation of time series data, key figures, and evaluation methods, along with the aggregation of multiple instances within a Pickle file, facilitates an efficient analysis of data produced. The insights thus obtained can then be employed to inform decision-making processes. In the event that a tested variant fails to meet the required specifications, the calibration data can be adjusted in subsequent iterations without the need for road tests. Following a successful off-car demonstration, the calibration data can be deployed and further refined on development vehicles.

4 Conclusion

The method and tools presented in this paper facilitate the efficient analysis and data-driven decision-making of new software versions and calibration datasets. In order to ensure an automated and consistent evaluation, an object-oriented approach is utilised. An analysis class has been designed to provide blueprints for the evaluation and visualisation of measurements taken in a variety of development environments. Following the successful completion of an off-car test, the specific instances of individual trips, along with time series data, key figures, calibration parameters, and predefined methods are aggregated in a result file. The data is then analysed using interactive dashboards.

In order to utilise the exploratory nature of road testing at an early stage of the product development process, an off-car imitation of road testing is employed as part of the OCM. The relocation of test drives to simulations and test benches allows a multitude

of measurements to be conducted in an efficient manner. However, the fast moving and rapid accumulation of data presents a challenge when off-car testing is conducted on a regular basis. To address this issue, the method and the aforementioned tools presented in this work are integrated into the OCM. Consequently, a few hours after a modification is made, key figures for the system behaviour in an in-service-equivalent operation can be provided and the system can be analysed. An example variant comparison demonstrates the power of the employed tools for efficiently analysing system behaviour. To quantify performance potential and comfort, key figures are calculated based on the state of charge of the traction battery and the frequency of internal combustion engine starts.

Further development of the OCM will entail extending the application of the analysis class to automated calibration. Additionally, key figures of off-car testing will be employed in the determination of target functions.

The presented work on automated data evaluation and aggregation plays a pivotal role in the success of automated, data-driven development of automotive software.

References

[AS24] ASAM e. V.: MDF (Measurement Data Format). https://www.asam.net/standards/detail/mdf/, accessed 16 Feb 2024.

[De24] Demirbaga, Ü. et al.: Big Data Analytics. Theory, Techniques, Platforms, and Applications. Springer Nature Switzerland; Imprint Springer, Cham, 2024.

[FIG20] Förster, D.; Inderka, R. B.; Gauterin, F.: Data-Driven Identification of Characteristic Real-Driving Cycles Based on k-Means Clustering and Mixed-Integer Optimization. IEEE Transactions on Vehicular Technology 3/69, pp. 2398–2410, 2020.

[HPJ20] Halstenberg, J.; Pfitzinger, B.; Jestädt, T.: DevOps. Ein Überblick. Springer Fachmedien Wiesbaden, Wiesbaden, 2020.

[Hr23] Hrisca, D.: ASAMMDF Documentation. https://asammdf.readthedocs.io/en/latest/, accessed 16 Feb 2024.

[IS23] Inmon, B.; Srivastava, R.: Rise of the Data Lakehouse. Building the Data Lakehouse. Technics Publications, Sedona, 2023.

[Ma22] Martini, T. et al.: Methodik und Tools zur Betriebsstrategieentwicklung für Performance-Hybride. 9. AutoTest Technical Conference, Stuttgart, 2022.

[My12] Myers, G. J.: The Art of Software Testing. J. Wiley & Sons, Hoboken, 2012.

[Pl24a] Plotly Technologies Inc.: Dash Documentation & User Guide. https://dash.plotly.com, accessed 13 Aug 2024.

[Pl24b] Plotly Technologies Inc.: Collaborative Data Science. https://plotly.com/python/, accessed 13 Aug 2024.

[Py24] Python Software Foundation: Pickle – Python Object Serialization. https://docs.python.org/3/library/pickle.html, accessed 13 Aug 2024.

[St18] Steyer, R.: Programmierung in Python. Ein kompakter Einstieg für die Praxis. Springer Vieweg, Wiesbaden, Heidelberg, 2018.

[Sy24] Synopsys, Inc.: Silver – Virtual ECU | Synopsys Verification. https://www.synopsys.com/verification/virtual-prototyping/silver, accessed 2 Aug 2024.

[SZ16] Schäuffele, J.; Zurawka, T.: Automotive Software Engineering. Springer Fachmedien Wiesbaden, Wiesbaden, 2016.

[Tu15] Tutuianu, M. et al.: Development of the World-Wide Harmonized Light Duty Test Cycle (WLTC) and a Possible Pathway for its Introduction in the European Legislation. Transportation Research Part D: Transport and Environment 40, pp. 61–75, 2015.

[UNR154] UNR154:18.08.2021, E/ECE/TRANS/505/Rev.3/Add.153/Rev.1.

[Wo18] Wolf, F.: Fahrzeuginformatik. Springer Fachmedien Wiesbaden, Wiesbaden, 2018.

[Wo24] World Wide Web Consortium: HTML Standard. https://html.spec.whatwg.org/multipage/, accessed 16 Feb 2024.

Herausforderungen und Lösungen für die durchgehende Digitalisierung des Testprozesses

Dr.-Ing. Stefan Unterschütz, Dipl.-Ing. (FH) Thomas Rönpage

IT-Lösungen für Tests
Werum Software & Systems AG
Anna-Vogeley-Straße 20
21337 Lüneburg
stefan.unterschuetz@werum.de
thomas.roenpage@werum.de

Abstract: In der Produktentwicklung werden zunehmend nicht nur die Ergebnisse aus Validierungs- und produktbegleitenden Tests herangezogen, sondern auch Daten, die während des Betriebs erfasst werden. Die Speicherung dieser Daten erfolgt verteilt auf unterschiedlichsten Medien und Systemen sowie auf cloudbasierten Lösungen. Um diese Daten effektiv nutzen, auswerten und miteinander in Beziehung setzen zu können, verwenden Entwickler und Datenanalysten verschiedenste Systeme – von Eigenentwicklungen über spezifische Lösungen bis hin zu KI-basierten Systemen. Für ganzheitliche Betrachtungen müssen die Testdaten mit weiteren Informationen aus MES (Manufacturing Execution System), PLM (Product Lifecycle Management) und den Daten aus dem Anforderungsmanagement verknüpft werden. In diesem Artikel werden Konzepte für den Aufbau moderner, skalierbarer und anpassbarer Lösungen für einen datengetriebenen Testprozess vorgestellt, die eine umfassende Digitalisierung des Versuchs- und Messdatenmanagements im Prüffeld und bei Feldversuchen ermöglichen.

1 Einleitung

Ziel der Digitalisierung ist es, Daten und Prozesse in eine digitale Form zu überführen, um Daten effektiver zu nutzen, den Automatisierungsgrad zu erhöhen, Geschäftsprozesse effizienter zu gestalten, neue Geschäftsmodelle zu entwickeln und die Entscheidungsfindung zu unterstützen. Während die Digitalisierung bereits seit vielen Jahren in unzähligen Systemen der Unternehmen Einzug gehalten hat, liegt der Fokus aktuell auf einer durchgängigen Digitalisierung von Systemen und Prozessen, um nachhaltige Konzepte aufzubauen und agil und resilient auf sich ständig ändernde Rahmenbedingungen reagieren zu können.

Ein aktuelles Beispiel aus dem Bereich der Produktentwicklung ist der digitale Zwilling, der das virtuelle Abbild eines physischen Objekts darstellt. Im Idealfall umfasst er alle Informationen, die während des Lebenszyklus eines Objekts anfallen, von Simulationsdaten über Fertigungsdaten bis hin zu Sensordaten aus dem realen Betrieb.

Als Anbieter von Softwarelösungen im Bereich Testprozesse und Testautomatisierung hat Werum viele Digitalisierungsprojekte seiner Kunden realisiert. Dieses Paper beschreibt die zentralen Herausforderungen dieser Projekte in den Branchen Automotive, Aerospace und Consumer Electronics und skizziert die Lösungsansätze, die zum Erfolg geführt haben. Der Fokus liegt dabei auf einer durchgängigen Digitalisierung des Testprozesses (Abbildung 1), es werden aber auch Anknüpfungspunkte für angrenzende oder unterstützende Prozesse aufgezeigt.

Abbildung 1: Wesentliche Schritte des Testprozess

2 Herausforderungen

Um den Testprozess vollständig zu digitalisieren, müssen eine Reihe von Herausforderungen überwunden werden, die teils allgemeiner Natur und teils spezifisch für Testprozesse sind.

Im Testprozess fließen Daten aus einer Vielzahl von Systemen zusammen, was den Prozess grundsätzlich sehr komplex macht. Beispiele sind: IIOT, MES, Ressourcenmanagement, PLM, ERP, Auftragsmanagement, Feldtests, Datenanalyse, Anforderungsmanagement sowie Excel und E-Mail. Bei einer durchgängigen Digitalisierung müssen neue Systeme und Prozesse nahtlos mit bestehenden Systemen zusammenarbeiten, obwohl Daten, Prozesse und Tools häufig heterogen sind.

Der Austausch zwischen den Systemen ist teilweise eingeschränkt, da die IT-Infrastruktur auf verschiedene Netze (Labor, Büro) oder Standorte verteilt ist und IT-Sicherheitsregeln zu beachten sind.

Eine weitere Herausforderung ist die Skalierbarkeit. Moderne Lösungen müssen mit immer größeren Datenmengen umgehen können, da sich die Variantenanzahl erhöht, eine höhere Testtiefe immer mehr Messdaten erzeugt und auch immer neuere Analyseverfahren zusätzliche Daten generieren.

Bei der Einführung eines neuen Systems müssen auch existierende Altdaten berücksichtigt werden, was eine große Herausforderung darstellen kann. Diese Altdaten liegen auf verschiedenen Systemen, in verschiedenen Formaten und enthalten meist unvollständige Metainformationen.

Zusätzlich muss das Thema Cybersecurity bei der Entwicklung neuer Systeme mit hoher Priorität berücksichtigt werden. Hierbei geht es um den Schutz vor Cyberangriffen, Datenverlust und anderen Sicherheitsbedrohungen.

Auch verschiedene Compliance-Anforderungen müssen eingehalten werden. Beispiele hierfür sind Qualitätsregularien, Datenschutzgesetze und -vorschriften wie die EU-Datenschutzgrundverordnung (DSGVO), ASPICE, TISAX, ISO 27001, ISO/IEC 17025 oder auch Exportkontrollen.

Digitalisierungsprojekte betreffen viele Personen in verschiedenen Abteilungen und Positionen. Involvierte Mitarbeiter benötigen umfassendes fachspezifisches und technisches Wissen, um die Digitalisierung erfolgreich durchführen zu können. Auch müssen teilweise Befindlichkeiten zwischen Abteilungen überwunden und eine hohe Akzeptanz bei den Anwendern erreicht werden.

Der Übergang zu neuen digitalen Lösungen erfordert oft Investitionen in neue Hardware, Software und Infrastruktur. Die TCO-Betrachtung (Total Cost of Ownerschip) muss auch die laufenden Kosten wie beispielsweise Kosten für die bestehende Infrastruktur, Betrieb, Wartung und Weiterentwicklung berücksichtigen.

Für die skizzierten Herausforderungen werden in den folgenden Kapiteln Lösungsansätze für eine durchgängige Digitalisierung erläutert. Der Fokus liegt dabei auf der Harmonisierung, Standardisierung, der Architektur und dem Rollout.

3 Harmonisierung und Standardisierung

Für eine durchgehende Digitalisierung sind die Themen Harmonisierung und Standardisierung von Datenmodellen und Prozessen unerlässlich (HAN21). Sie sind wichtige Erfolgsfaktoren für eine skalierbare, erweiterbare und wartbare Systemlandschaft. Leider stellt dieses Thema auch eine der größten Herausforderungen dar, da es in die „Hoheitsgebiete“ verschiedener Abteilungen eingreift. Deshalb muss eine gemeinsame Lösung gefunden werden. Bestimmte Abläufe sind in den Abteilungen absolut notwendig und müssen erhalten bleiben, aber übermäßiger Protektionismus sollte vermieden werden. Dazu empfiehlt sich eine Benutzerbefragung, eine Prozessanalyse und die Betrachtung der aktuellen Systeme. Die Einbeziehung wichtiger Entscheider und Key-User ist essenziell.

Ein einheitliches Datenmodell bedeutet nicht, dass alle Stellen mit exakt den gleichen Daten arbeiten müssen. Vielmehr geht es darum, sich auf einige Konstanten zu einigen. So sollten z.B. einheitliche Identifikatoren für die Prüflinge verwendet werden, was u.a. die Umsetzung des digitalen Zwillings ermöglicht.

Insbesondere für Messdaten, die abteilungsübergreifend genutzt werden sollen oder deren Verfügbarkeit über mehrere Jahre gewährleistet sein muss, ist eine Standardisierung erforderlich. Hierfür eignen sich Standardformate wie MDF4, HDF5 oder TDMS. Diese Formate werden langfristig von verschiedenen Werkzeugen unterstützt.

Bei prozessübergreifenden Systemen ist es wichtig, ein führendes System klar zu definieren. Dies ist besonders entscheidend, wenn Daten redundant in verschiedenen Systemen vorgehalten werden. So muss festgelegt werden, welches System Änderungen unter bestimmten Statusbedingungen durchführen darf. Zum Beispiel sollten nach der Übermittlung eines Testauftrags vom Testmanagementsystem an die Prüfstandsautomatisierung keine weiteren Modifikationen mehr an der Parametrisierung des Testauftrags durch das Testmanagementsystem erfolgen.

4 Architektur

In gewachsenen, heterogenen Unternehmensstrukturen ist eine komplette Neuentwicklung aller Geschäftsanwendungen nahezu unmöglich. Stattdesen bietet sich eine Integrationsarchitektur an, die den Datenfluss zwischen verschiedenen Systemen und Anwendungen innerhalb eines Unternehmens ermöglicht. Hierzu werden im Folgenden wichtige Konzepte vorgestellt.

4.1 Infrastruktur

Bei der Überlegung, ob der Betrieb on-premise oder in der Cloud erfolgen soll, wird in vielen Fällen ein Mischbetrieb sinnvoll sein. Auch wenn die Vorteile von cloud-basierten Lösungen durch hohe Kostenkontrolle, vielfältige und flexible Angebote überwiegen und auch den globalen Betrieb wesentlich vereinfachen, gibt es bei einigen Anwendungen gute Gründe, diese on-premise zu betreiben. So können Regularien wie die Datensicherheit, aber auch hohe Leistungsanforderungen z.B. bei der Messdatenerfassung den on-premise Einsatz erfordern.

Darüber hinaus bietet die Unterstützung von Containerisierung wie bspw. durch Docker viele Vorteile. So können Orchestrierungsplattformen wie Kubernetes zur Skalierung eingesetzt werden und die Laufzeitumgebung kann einfach aktualisiert werden. Auch das Ausrollen neuer Software über Continuous Integration/Delivery Pipelines ist damit möglich.

Um einen autarken Betrieb zu gewährleisten, muss eine lose Kopplung der Systeme realisiert werden. So kann z.B. eine Prüfstandsautomatisierung einen Testauftrag inkl. Prüfprogramm komplett laden und cachen, anstatt diesen sukzessive zur Laufzeit abzurufen. Durch Caching werden auch temporäre Netzwerkausfälle überbrückt.

Da im Testprozess verschiedene Dienste aus unterschiedlichen Netzwerksegmenten miteinander kommunizieren müssen, sind Sicherheitsaspekte von Firewalls und Netzwerkfreigaben zu berücksichtigen. Während das Testmanagement im Büronetzwerk stattfindet, das in der Regel mit dem Internet verbunden ist, erfolgt die Testdurchführung durch die Prüfstandsautomatisierung im stärker gesicherten Labornetzwerk. Die Kommunikationsrichtung sollte immer vom sicheren zum unsicheren Netzwerk verlaufen.

4.2 Schnittstellen und Datenaustausch

Damit verschiedenen Systeme im Testprozess ihre Daten digital austauschen können, sind leicht zugängliche, gut dokumentierte und robuste Schnittstellen erforderlich. Hierfür hat sich die Webtechnologie REST etabliert, die eine breite Unterstützung vieler Programmier- und Skriptsprachen ermöglicht und durch Einfachheit, Flexibilität und Sicherheit überzeugt.

Um einen effizienten Datenaustausch über die Schnittstellen zu ermöglichen, ist die Einführung eines standardisierten Datenmodells (Abschnitt 3) notwendig. Insbesondere Altsysteme weichen jedoch von Standards ab und können oft nicht mehr angepasst werden. Hier kann ein Konnektor helfen, um eine (teil-)automatische Verknüpfung mit anderen Systemen zu erreichen. Der Konnektor dient dabei als Vermittler und Transformator zwischen zwei Diensten, wobei die Systeme selbst entkoppelt bleiben, was ein großer Vorteil ist.

Konnektoren werden typischerweise verwendet, um z.B. verschiedene Prüfstände oder Prüfstandsautomatisierungen mit einem Testdatenmanagementsystem zu verbinden. Die Konnektoren können die Daten aus den verschiedenen Prüfständen in eine einheitliche Struktur überführen, bevor sie an das Testdatenmanagementsystem übergeben werden. Die Daten müssen nicht immer sofort übertragen werden. Im Hinblick auf den autonomen Betrieb und die Ausfallsicherheit ist es sinnvoll, die Daten lokal zu puffern und gebündelt zu übertragen.

Neben Konnektoren bietet sich auch ein Datenaustausch über Publish/Subscribe-Mechanismen bzw. Eventbusse an. Typische Vertreter im Bereich der Testprozesse sind OPC-UA, MQTT, DDS. Bei einem Bus kann sich ein Daten- oder Ereignisproduzent mit einem Topic-Namen registrieren und dann bei Bedarf Messdaten, Ereignisse oder auch Statusinformationen publizieren. Interessierte Konsumenten können dann asynchron auf diese Daten zugreifen.

Abbildung 2 zeigt beispielhaft, wie die Kommunikation in einer größeren Systemlandschaft aussehen kann.

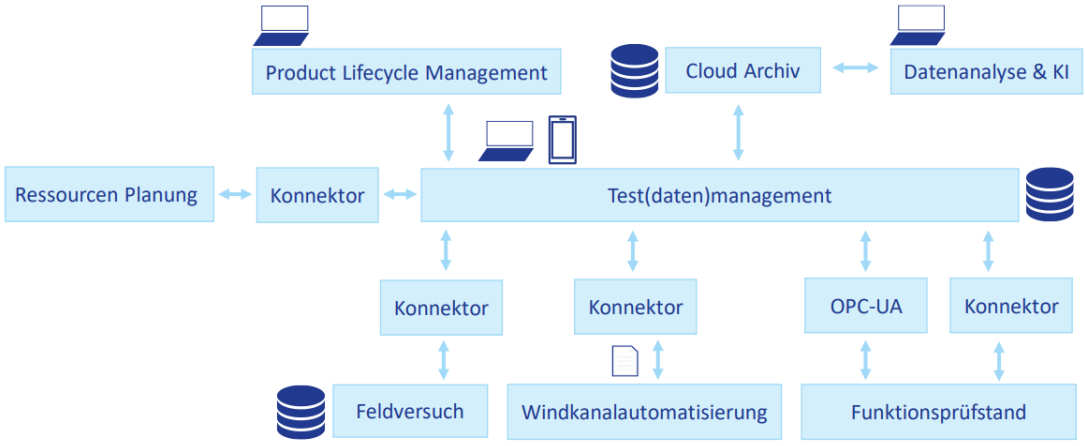


Abbildung 2: Beispielhafter Systemaufbau

4.3 Data-Ingestion und Data-Analytics

In den vorherigen Abschnitten lag der Fokus auf Systemen und der Kommunikation zwischen diesen. In diesem Kapitel wird beschrieben, wie solche Systeme im Sinne einer Toolchain fachlich zusammenarbeiten. Für den Testprozess sind die Themen Datenerfassung bzw. Datenimport sowie Datenspeicherung und -verarbeitung von großer Bedeutung. Hier bietet sich der Einsatz einer Data Analytics Toolchain an (BOH20), um die Daten für interdisziplinäre Auswertungen einheitlich zur Verfügung zu stellen. Ein Beispiel dafür ist in Abbildung 3 dargestellt. Die Toolchain dient als Brücke zwischen Datenproduzenten und Datenkonsumenten.

Der Import von Daten bzw. die Eingabe von Ergebnissen kann automatisch oder manuell erfolgen. Der automatische Upload kann über Übergabeverzeichnisse oder Web-Schnittstellen erfolgen. Die Übertragung erfolgt kontinuierlich im Stream oder in Batches. Letzteres wird z.B. bei Versuchsfahrten verwendet, bei denen das Versuchsfahrzeug keine kontinuierliche Verbindung zum Datenspeicher hat und die Daten zunächst temporär sammelt (ZEU23).

Die Weiterverarbeitung umfasst die Bereinigung, Anreicherung und Harmonisierung der Daten sowie die Aufbereitung der Metadaten. Die Daten werden vorzugsweise in Object Stores oder Dateisystemen gespeichert, während die Metadaten in relationalen Datenbanken verwaltet werden. Die Qualität der Metadaten ist dabei entscheidend, da sie für die Suche und Auswertung verwendet werden. Die Metadaten sollten daher den Test umfassend und standardisiert beschreiben, beispielsweise durch die Angabe der genutzten Ressourcen.

Die Datenanalyse und -auswertung ist ein zentraler Bestandteil der Werkzeugkette. Durch die Integration unterschiedlicher Analysesoftware, können neben der explorativen Analyse einzelner Daten auch standardisierte Auswertungsroutinen realisiert werden. Darüber hinaus eröffnen Technologien wie Machine Learning und KI neue Anwendungsmöglichkeiten. So können durch Predictive Maintenance Anomalien in Datensätzen frühzeitig und automatisch erkannt werden.

Der Zugriff und die Verteilung der Daten erfolgt über gesicherte Weboberflächen und Webschnittstellen. Dies ermöglicht unter anderem die Einschränkung der Sichtbarkeit auf bestimmte Datensätze aber auch die Einhaltung von Compliance-Anforderungen.

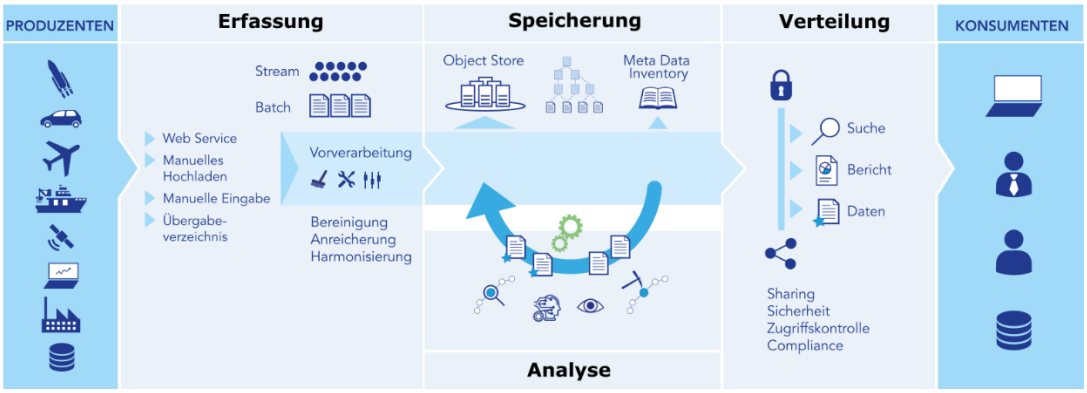


Abbildung 3: Funktionaler Aufbau der Data Analytics Toolchain

4.4 Compliance - Sicherheit - Governance

Neben den funktionalen Anforderungen an Systeme und Anwendungen sind heute weitere Anforderungen im Bereich Compliance, Security und Governance zu erfüllen. Insbesondere bei der Beschaffung neuer Software sollten die nachfolgenden Funktionen bereits integriert sein, um die Zukunftssicherheit zu gewährleisten.

Für die Authentifizierung werden in der Regel zentrale Single-Sign-On-Dienste verwendet, die über Standardprotokolle wie SAML, OAuth2 oder OpenID Connect angesprochen werden. Verschiedene Dienste können hier auf einheitliche Benutzerkonten zurückgreifen. Dies ist für den Benutzer komfortabel und reduziert den Administrationsaufwand erheblich. Für eine zentrale Konfiguration der Autorisierung, d.h. die Prüfung von Berechtigungen im Testprozess, ist ein ähnlicher Ansatz möglich. Teilweise werden jedoch sehr komplexe Berechtigungsprüfungen benötigt, sodass die Prüfung von einem konkreten Dienst übernommen werden muss.

Im Umgang mit Testdaten sind Nachvollziehbarkeit, Transparenz und Auditierbarkeit erforderlich. Dafür müssen Mechanismen integriert werden, um Änderungen an den Daten zu protokollieren. Diese Anforderungen sind insbesondere für Qualitätsstandards wie ISO 17025 relevant.

Generell muss während des gesamten Testprozesses eine gute Daten-Governance gewährleistet sein. Diese umfasst unter anderem die Sicherstellung der Richtigkeit, der Qualität und des Zugangs zu den Daten. In Bezug auf die DSGVO müssen weitere Funktionen wie Zugriffsschutz, Anonymisierung und Löschfunktionen implementiert werden.

Zusätzlich müssen Sicherheitsrichtlinien umgesetzt werden, um einen sicheren Betrieb der Software zu gewährleisten. Dazu gehören der Einsatz von Firewalls, Verschlüsselung und regelmäßige Sicherheitsüberprüfungen, um die eingesetzten Softwarebibliotheken und Systeme auf Sicherheitslücken zu überprüfen und die Integrität und Vertraulichkeit der Daten zu gewährleisten. Hervorzuheben ist hier das Open Worldwide Application Security Project (OWASP), das Richtlinien, Tools und Best Practices für den sicheren Betrieb von Software zur Verfügung stellt.

5 Rollout

Besonders in größeren Projekten ist ein Rollout als „Big Bang“ ausgeschlossen. Beim „Big Bang“ findet der Wechsel auf die neuen Systeme in kürzester Zeit ohne Rückfallmöglichkeiten auf vorherige Systeme statt. Das Risiko von Produktionsausfällen und Datenverlusten ist hoch, da selbst umfangreichste Vorabtests nie alle Eventualitäten abdecken können.

Erfolgreiche Rollouts bedienen sich der folgenden Konzepte:

- Start mit einem Pilotbetrieb oder Parallelbetrieb mit bestehenden Systemen
- Schrittweises Onboarding von Personen, Abteilungen und Projekten

- Schrittweise Bereitstellung von neuen Funktionen
- Schrittweise Integration von Systemen bzw. Konnektoren zu den Systemen
- Falls erforderlich, schrittweise Übernahme von Altdaten
- Aufbau von Test-, Qualitäts- und Produktivsystem für das Ausrollen von Software.

Zudem ist ein angemessenes Erwartungsmanagement wichtig. Fehlen zum Start wichtige Funktionen, die später hinzugefügt werden sollen, dann muss den Beteiligten klar gemacht werden, dass bestimmte Funktionen erst in späteren Releases unterstützt werden. Wenn man den Erstnutzern die Möglichkeit gibt, mit ihrem Feedback aktiv an der Verbesserung teilzunehmen, erhöht das die Akzeptanz. Auch sind eine gute Einweisung und Wissensmanagement wichtig für Akzeptanz und Produktivität.

6 Zusammenfassung & Ausblick

In diesem Paper wurden verschiedene Konzepte für eine durchgängige Digitalisierungslösung im Testprozess vorgestellt. Aus Werums langjähriger Erfahrung in den verschiedensten Industriebereichen haben sich diese Konzepte als Best Practice erwiesen. Allerdings hat sich auch gezeigt, dass es keine „Patentlösung“ gibt, sondern dass stets individuelle Kundenanforderungen berücksichtigt werden müssen. Zudem sollte die Digitalisierung eher als kontinuierlicher Prozess und weniger als einmalige Aktivität verstanden werden.

Aktuell steht neben der digitalen Transformation auch die KI-Transformation im Fokus. Bei der Definition der Unternehmensprozesse und dem Aufbau neuer Systeme sollte bereits heute Künstliche Intelligenz berücksichtigt werden. Die Bandbreite möglicher Use Cases ist vielfältig: Von einfachen Wissensabfragen oder Korrekturaufgaben bis hin zur Ausführung komplexer Datenextraktionen, Auswertungen und Datenanalysen ist vieles möglich. Besonders Tätigkeiten, die sich gut beschreiben lassen, können mit großen Sprachmodellen (LLMs) automatisiert werden.

7 Literaturverzeichnis

[BOH20] Hendrik Bohlen und Stefan Unterschütz, From Data to Company Values, ATZelectronics worldwide 15, 2020

[HAN21] Björn Hansen, Standardisierte Lösungen in komplexen Systementwicklungsprozessen, ATZextra 6, 2021

[ZEU23] Uwe Zenker und Thomas Rönpage, Milliarden Messwerte im Validierungstest, HANSER automotive 2, 2023

Testfall-Erkennung in Erprobungsfahrten

Julian Fuchs (1), Christopher Kober (2)

(1) FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10–14, 76131 Karlsruhe
(2) Mercedes-Benz AG, E/E Testing Prüfstand
Werk 059, HPC L334, 71063 Sindelfingen
fuchs@fzi.de
christopher.kober@mercedes-benz.com

Abstract: Zur Absicherung assistierender und automatisierender Fahrzeugfunktionen werden bei der Mercedes-Benz AG Erprobungsfahrten und Fahrzeugtests basierend auf Testfallkatalogen eingesetzt. Bei Erprobungsfahrten, seien sie real oder im Hardware-in-the-Loop-(HiL)Umfeld, werden große Mengen an Fahrzeugdaten aufgezeichnet. Diese Erprobungsfahrten beinhalten zu einem großen Teil die gleichen Manöver und Fahrszenarien, die in Prüfgeländeversuchen oder Systemtests am HiL-Prüfstand durchgeführt und analysiert werden. In diesem Beitrag soll eine Methodik vorgestellt werden mit deren Hilfe Testfälle aus Erprobungsfahrten extrahiert und ausgewertet werden können. Dadurch kann eine höhere Testabdeckung erreicht und Testzeit eingespart werden. Die Methodik umfasst die Auswahl repräsentativer Erprobungsfahrten, die Extraktion relevanter Testfälle aus den aufgezeichneten Daten, die Abstraktion der Testfälle aus den Testfallkatalogen sowie die Entwicklung und Anwendung von Analyse- und Auswertungsalgorithmen. Die Evaluation erfolgt anhand verschiedener Kriterien wie Testabdeckung, Testzeitersparnis und Überprüfung der Genauigkeit der extrahierten Testfälle im Vergleich zu den ursprünglichen Testfallkatalogen prototypisch am Beispiel diverser Fahrfunktion der Mercedes-Benz-AG.

1 Einleitung

Die fortschreitende Komplexität moderner Fahrzeugsysteme und der stetige Anstieg der Anforderungen an Sicherheit und Zuverlässigkeit erfordern eine immer umfassendere Validierung und Verifikation der eingesetzten Technologien. Um sicherzustellen, dass alle relevanten Situationen abgedeckt werden, ist es entscheidend, eine hohe Testabdeckung zu gewährleisten. Die bisherigen Ansätze im Testprozess sind jedoch oft zeit- und ressourcenintensiv, insbesondere wenn Tests auf dem Prüfgelände durchgeführt werden müssen [1] [2] [3]. Zudem besteht die Herausforderung, dass nicht alle realen Fahrbedingungen in den definierten Testfällen erfasst werden.

Hier setzt die entwickelte Methodik an, die darauf abzielt, aufgezeichnete Realdaten aus Erprobungsfahrten systematisch auf das Vorkommen von Testfallinhalten zu untersuchen. Durch diese Methodik wird es möglich, bereits in realen Fahrsituationen durchgeführte Tests zu erkennen und so die Testabdeckung zu erhöhen. Dadurch können nicht nur Testaufwände reduziert werden, sondern es wird auch eine effizientere Nutzung der vorhandenen Ressourcen ermöglicht.

2 Herausforderungen und Zielsetzung

Die Implementierung dieser Methodik steht jedoch vor mehreren Herausforderungen:

- Abstraktion der Testfälle: Testfälle aus den bestehenden Testfallkatalogen müssen so abstrahiert werden, dass eine automatisierte Erkennung ihres Inhalts in den aufgezeichneten Realdaten möglich ist. Dies erfordert eine sorgfältige Analyse und Modifikation der Testfallbeschreibungen, um die essenziellen Merkmale herauszufiltern und sie auf ein Niveau zu bringen, das für die automatisierte Verarbeitung geeignet ist.
- Integration in bestehende Datenpipelines: Die Methodik soll in die bestehende Datenpipeline integriert werden, was technologische und organisatorische Anpassungen mit sich bringt. Die nahtlose Einbindung in bestehende Prozesse und Systeme ist entscheidend, um den maximalen Nutzen zu erzielen und Reibungsverluste zu vermeiden.
- Datenmengen und Visualisierung: Die Menge an Daten, die bei Erprobungsfahrten generiert wird, stellt eine weitere Herausforderung dar. Um die relevanten Informationen effizient zu verarbeiten und darzustellen, müssen Konzepte und Mockups für eine geeignete Visualisierung entwickelt werden. Diese sollen es den Ingenieuren und Entwicklern ermöglichen, die wesentlichen Informationen schnell zu erfassen und fundierte Entscheidungen zu treffen.

Das übergeordnete Ziel ist die kontinuierliche, systemunabhängige Erkennung und Bewertung von Testfällen, unabhängig von der jeweiligen Testinstanz und der zu bewertenden Funktionalität.

3 Konzept zur Abstraktion von Testfällen zur Anwendung auf realen Daten

Normalerweise bestehen Testfälle aus mehreren Testschritten. Diese Testfälle werden sequentiell nacheinander ausgeführt. Testfahrer erhalten ihren klaren Testauftrag und führen diesen aus. Es wird dabei meist nur eine Funktionalität gleichzeitig bewertet. Zusätzlich werden Erprobungsträger in Abend- und Wochenenderprobungen genutzt, um Daten aufzuzeichnen und Auffälligkeiten zu finden. Hierbei fallen große Datenmengen an, deren manuelle Auswertung mit einem großen zeitlichen und ressourcentechnischen Aufwand verbunden ist [4].

3.1 Ermittlung von Aktionen und Bewertungen auf Basis der Testspezifikation

Als Eingangsgröße in das entwickelte Konzept dienen Testspezifikationen und aufgenommene Zeitreihen, der Fahrzeugbusse (oder auch von Referenz Sensorik). Dabei werden zunächst die bestehenden Testspezifikationen genutzt und die einzelnen Testschritte unterteilt. Für jeden Testschritt werden Aktionen und Bewertungen definiert. Die Aktionen werden auf Basis der Stimulationen im Testschritt (bspw. Fahrer schnallt sich ab) festgesetzt. Die Bewertungen beschreiben den idealen Soll-Zustand der Systemfunktionalität. Jede Aktion hat einen Start- und einen Endzeitpunkt, genauso wie eine Bedingung (Aktivierungsbedingung), die erfüllt sein muss, damit die Aktion aktiv sein kann (bspw. Fahrer ist abgeschnallt). Jede Bewertung hat genauso wie eine Aktion einen Start- und Endzeitpunkt und eine Bedingung, die erfüllt sein muss, damit die Bewertung aktiv ist, hat aber zusätzlich noch ein Bewertungskriterium, das ein Testergebnis (pass/ failed) hat [5] [6].

Die Testfälle werden in einer Abstraktionsmatrix dargestellt, die zur Erstellung einer analytischen Testfalldatenbank verwendet werden kann. Die Abstraktionsmatrix soll dabei an die Testfälle und deren Testschritte gekoppelt werden können, um eine Parametrisierung der Analyseskripte zu gewährleisten. Auch Metainformationen, wie Testfall ID, werden in der Abstraktionsmatrix gespeichert.

Sobald die Aktionen und Bewertungen definiert wurden, kann in die zyklische Abarbeitung der Messdaten gestartet werden. Dieser Prozess besteht aus vier Schritten (siehe Abbildung 1).

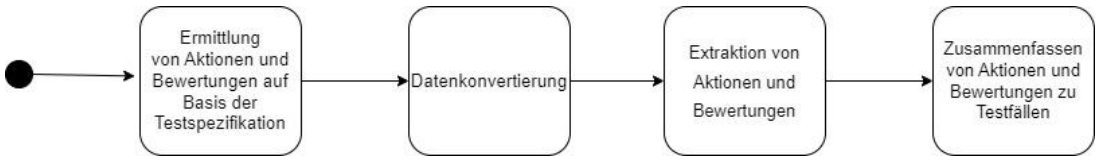


Abbildung 1: Konzept zur Abstraktion der Testfälle

3.2 Extraktion von Aktionen und Bewertungen

Anschließend werden die Aktionen und Bewertungsbedingungen auf die Auffälligkeiten angewandt. Eine Aktion ist dann aktiv, wenn die in Kapitel 3.1 ermittelten Bedingungen erfüllt sind. Das heißt konkret, wenn die Signalwerte des aktuell betrachteten Zeitpunkts den ermittelten Werten entsprechen. Der Zeitpunkt, bei dem die gefundene Aktion zum Ersten Mal aktiv ist, wird als Startzeitpunkt protokolliert. Sobald die Bedingung wieder nicht erfüllt ist, wird der Endzeitpunkt geschätzt und eine gefundenen Aktion-Instanz abgespeichert.

Genauso passiert dies bei den Bewertungen. Auch hier wird zunächst überprüft, ob die Aktivierungsbedingung erfüllt ist. Sobald diese Bedingung erfüllt ist, wird der Zustand der Signale überwacht und mit dem Bewertungskriterium verglichen. Wird das formulierte Bewertungskriterium innerhalb der Aktivierungszeit verletzt, wird die Bewertung als „failed“ markiert – ansonsten als „passed“ (vgl. Ansatz nach King et al. [7])

Zum Schluss erhält man eine Liste über gefundenen Aktionen und Bewertungen. Die Bewertungen können dabei unabhängig von der späteren Aggregation zu Testfällen zur Bewertung der Systemperformance bzw. -Reifegrad verwendet werden.

3.3 Zusammenfassen von Aktionen und Bewertungen zu Testfällen

Im nächsten Schritt werden die Aktionen und Bewertungen automatisch zu Testfällen zusammengebaut und für jeden Testfall ein Ergebnis ermittelt. Hierfür ist die zeitliche Abfolge von Aktivierung und Deaktivierung entscheidend. Die sequentielle Kombination von mehreren Aktionen untereinander beschreiben den Testfall. Mit Hilfe eines Suchalgorithmus werden passende Sequenzen an Aktionen in der Zeit gefunden. Hierzu wird die Liste an in der Messung gefundenen Aktionen für jeden Testfall separat untersucht. Anschließend werden die gültigen Bewertungen für jeden gefundene Testfall überprüft und ebenso wie die gefundenen Aktionen zum Testfall hinzugefügt. Das Gesamtergebnis des Testfalls bildet sich aus einer logischen Verknüpfung der Teilergebnisse der gefundenen Bewertungen (siehe beispielsweise Abbildung 2: Beispielhafte Abstraktion).

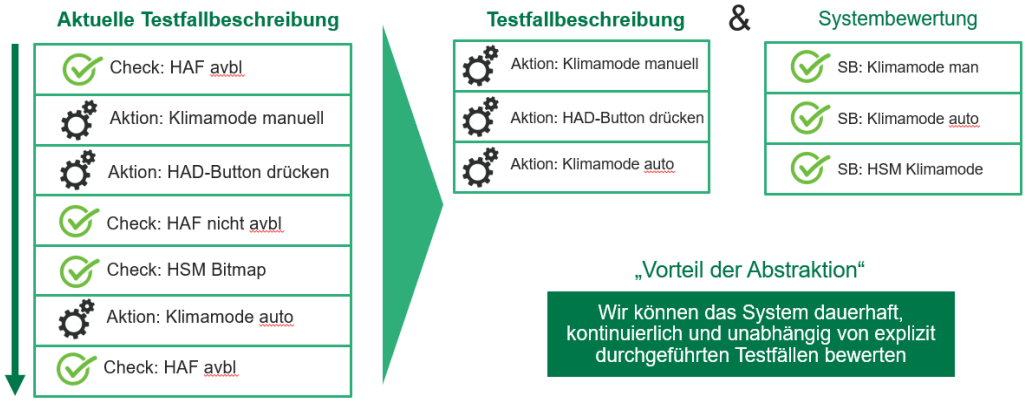


Abbildung 2: Beispielhafte Abstraktion in Testfallbeschreibung (Aktion) und Systembewertung

Ein Testfall kann somit mehrmals in einer Messung gefunden werden und wird zusammen mit den gefundenen Aktionen, Bewertungen und dem aggregierten Ergebnis dann in der in Kapitel 4 eingeführten Datenstruktur abgelegt.

3.4 Visualisieren der Teilergebnisse und manuelle Auswertung

Die Visualisierung der Teilergebnisse kommt nach dem 4. Schritt aus Abbildung 1. Sie ermöglicht es den Ingenieuren, schnell und effizient einen Überblick über die gefundenen Testfälle, Aktionen und Bewertungen zu erhalten. Das Ziel der Visualisierung ist es, die Ergebnisse so aufzubereiten, dass eine intuitive und übersichtliche Auswertung möglich wird.

Das zentrale Element der Visualisierung ist ein Dashboard, das die wesentlichen Informationen zu den durchgeführten Testfahrten und den dabei erkannten Testfällen aggregiert darstellt. Hierbei werden die Ergebnisse in Form von Diagrammen, Tabellen und interaktiven Grafiken präsentiert. Das Dashboard ist modular aufgebaut und erlaubt es, unterschiedliche Ansichten zu wählen, abhängig von der Rolle des Nutzers (z.B. Testingenieur, Entwickler, Management).

Die Ergebnisse können nach verschiedenen Kriterien gefiltert und sortiert werden, z.B. nach Testfahrt, Fahrzeugmodell, Datum der Erprobungsfahrt oder nach spezifischen Testfällen. Eine Zeitachse zeigt die zeitliche Abfolge der erkannten Aktionen und Bewertungen innerhalb einer Testfahrt, wodurch Auffälligkeiten und potenzielle Probleme leicht identifiziert werden können.

4 Strukturierung der Daten

Zur strukturierten Ablage der Ergebnisse wurde eine relationale Datenbank aufgebaut, welche in verschiedenen Tabellen die erkannten Aktionen, Bewertungen und Testfälle zu einer Messung speichert. Zusätzlich werden Meta-Informationen zu jeder Messung abgespeichert, um Informationen zur Baureihe, zur verwendeten Software, etc. auch im Nachgang zur Verfügung zu haben. Ziel der Struktur ist eine filter- und sortierbare Speicherung.

Der zentrale Bestandteil des Entitätsdiagramm ist die Liste der Testfahren (test_drives), hier werden pro Testfahrt Meta Informationen, wie Länge der Testfahrt, das Datum und die Dauer der Testfahrt, sowie eine Referenz auf ein entsprechendes Fahrzeug gespeichert werden.

Jeder Testfall besteht neben einer eindeutigen ID und einem Namen wiederrum aus mehreren Aktionen (actions) und Bewertungen (assessments). Wird nun eine Messung ausgewertet, werden Aktionen (extracted_actions) und Bewertungen (extracted_assessments) gefunden, die zu gefundenen bzw. extrahierten Testfällen gespeichert werden. Jeder gefundene Testfall (extracted_test_case) ist somit eindeutig mehreren Aktionen und/oder Bewertungen zugeordnet und gehört eindeutig zu einer Messung. Die Datenablage dient als Basis für die Visualisierung und Speicherung der Daten.

5 Evaluation

Zwischen den beiden Testinstanzen, die in dieser Methode gemeinsam ausgewertet werden können, besteht der gravierende Unterschied der Referenzsensorik, also einer zusätzlichen Informationsquelle zur Überprüfung der eigentlichen Fahrzeugsensoren. In den Daten aus Fahrten, die am HiL-Prüfstand durchgeführt wurden, und woher die Grundzüge dieser Methode stammen, ist diese Referenzsensorik über die bekannte Simulationsumgebung vorhanden. Dies erhöht die Möglichkeiten einer Aktivierung ungemein, da die Umwelt und der Zustand des Fahrzeuges zu jeder Zeit bekannt sind.

Bei den Daten aus realen Erprobungsfahrten ist keine Referenzsensorik verbaut, das heißt die Implementierung von Aktionen und Bewertung muss sich auf die vorhandenen Bus-Signale stützen. Zusätzliche Informationen, die der Testfahrer bereitstellen könnte, sind schwierig automatisiert zu verarbeiten und mit einiger Ungenauigkeit behaftet, weshalb auch darüber keine Lösung der fehlenden Referenzsensorik erreicht wird.

Aufgrund der, durch die oben beschriebene Problematik, reduzierten Möglichkeiten der Bestimmung der Aktivierung der einzelnen Aktionen kann die Methode bei realen Testfahrten nicht die gleiche Aussagekraft und den Umfang der Prüfungen im Vergleich zu Fahrten in simulierten Umgebungen liefern. Nichtsdestotrotz wurde ein wertvoller erster Schritt getätigt, der mit geringem Aufwand zusätzliche Erkenntnisse aus vorhandenen Fahrdaten liefern kann.

6 Ausblick

Es gibt verschiedene Möglichkeiten, wie die Methode in der Zukunft erweitert werden könnte. Zum einen wäre es von Vorteil, die Algorithmen der Methode direkt in das Cloud Portal zu integrieren, in dem die Messdaten von Realfahrten abgelegt werden. Dies würde die Anwendung erleichtern und die Methode einem breiteren Publikum zugänglich machen, als es derzeit bei der „offline“ Auswertung der Fall ist.

Eine Möglichkeit, das Problem der fehlenden Referenzsensorik anzugehen, wäre eine Auswertung von Videoaufzeichnungen der Sensor-Kameras. Hier könnte beispielsweise ein neuronales Netz zur Erkennung verschiedener Situationen eingesetzt werden. Implementierung und Auswertung wären aber sehr aufwändig, darüber hinaus würde man zu einem gewissen Maße, die Fahrfunktion „nachprogrammieren“, die ja ebenfalls über Sensoren das Umfeld klassifiziert und Situationen bewertet.

Zusätzlich könnten GPS-Positionen, die während der Fahrt aufgezeichnet werden, auf interessante oder kritische Straßenpunkte untersucht werden. Dadurch würde sich eine zusätzliche Informationsquelle ergeben, die möglicherweise mehr Aktivierungen oder Prüfungen ermöglicht (Abbildung 3).

Abbildung 3: Aktionen in Bezug zur Kartenposition

7 Literaturverzeichnis

[1] R. Pfeffer, P. Ukas und E. Sax, „Potential of virtual test environments for the development of highly automated driving functions using neural networks," *ATZ Fahrerassistenzsysteme,* 2018.

[2] E. Sax, *Tatort Test „Test is not the Last",* Stuttgart: Vector Testing Symposium, 2015.

[3] Waymo, „Wayme Autonomous Vehicle Disengagement Report 2017," Waymo, 2017.

[4] C. Wang und H. Winner, „Overcoming Challenges of Validation Automated Driving and Identification of Critical Scenarios," in *IEEE Intelligent Transportation Systems Conference, ITSC*, 2019.

[5] S. Otten, J. Bach, C. Wohlfahrt, C. King, J. Lier, H. Schmid, S. Schmerler und E. Sax, „Automated Assessment and Evaluationof Digital Test Drives," in *Advanced Microsystems for Automotive Applications 2017*, Karlsruhe, Springer Verlag, 2017.

[6] C. King, J. Bach, S. Otten und E. Sax, „Identifikation von Fahrszenarien während einer virtuellen Testfahrt," in *INFORMATIK 2017*, Chemnitz, 2017.

[7] C. King, L. Ries, C. Kober, C. Wohlfahrt und E. Sax, „Automated Function Assessment in Driving Scenarios," in *IEEE International Conference of Software Testing, Verification and Validation*, 2019.

Automated NCAP tests with ASAM OpenSCENARIO

Jakob Kaths, Benedikt Schott, Hannah Bernauer, Markus Pielmeier, André Pinnel

Solutions for Virtual Test Drives
Vector Informatik GmbH
Baierbrunner Str. 23
81379 München
jakob.kaths@vector.com

Abstract: Euro NCAP provides descriptions of collision scenarios to test active safety systems of vehicles such as AEB. To conduct these tests in simulation, the scenarios must be described in a machine-readable format. This application story sheds light on the capabilities of OpenSCENARIO XML and OpenDRIVE in that regard. As the Euro NCAP scenarios are widely used, Vector decided to contribute the implementation of these scenarios using the ASAM standards open source at https://github.com/vectorgrp/OSC-NCAP-scenarios.

1 Introduction

It is undisputed that the development and validation of complex ADAS functions require scenario-based testing as an essential component of the test strategy. Due to the high resource consumption of real-world test drives and the often late availability of technology carriers and prototypes in the development process, these test drives are increasingly conducted in a simulated manner. This process already begins in early development phases with model-based function design (MIL) continues through the implementation of functions as control unit code (SIL) and extends to testing with isolated but real target control units (HIL). It is desirable to reuse any test drive scenarios across all development phases and different execution environments as unchanged as possible.

By defining input formats to compliant simulators, OpenSCENARIO and OpenDRIVE form the basis for this reusability. According to the OpenSCENARIO XML specification “a scenario is a description of how the view of the world changes with time, […] this encompasses […] both world-fixed (static) elements such as the road layout and road furniture, world-changing (dynamic) elements such as weather and lighting, vehicles, objects, people, and traffic light states.” [ASA24]. While OpenDRIVE covers the static part of the scenario, OpenSCENARIO addresses the dynamic content.

The relevant scenario content can stem from various sources, including recorded test drives, accident databases, expert knowledge from function developers, or regulatory requirements. Euro NCAP, for example, provides descriptions of collision scenarios to test active safety systems of vehicles. Autonomous emergency braking (AEB) and forward collision warning are the most relevant of these systems. Although these tests are not mandatory, one may count them towards the regulatory requirements. The specified scenarios cover the pre-dominant accidents found in relevant databases [GER15] and include collisions between the vehicle under test and a variety of targets including passenger cars [EUR24a], motorcycles, bicycles, and pedestrians [EUR24b]. This and the fact that the ratings by Euro NCAP are of high importance for OEMs make these scenarios well-accepted and widely used, both on real test tracks and in simulation.

2 Motivation

Simulations of Euro NCAP scenarios are conducted by the OEMs and their suppliers. Furthermore, numerous simulators exist, many of them with proprietary scenario descriptions. Therefore, it must be suspected that many implementations of the scenarios in various formats have been created in the past, both by users of simulators and tool providers. This, undoubtedly, has many disadvantages: duplicated efforts, non-reusable scenarios in proprietary formats and potential misinterpretations of the scenario definitions to name just a few. To alleviate these issues, we decided to comprehensively evaluate the suitability of the open standards OpenDRIVE and OpenSCENARIO for modeling various scenarios [FUN24]. Specifically, the following aspects were examined:

- The use of OpenSCENARIO XML parameters to avoid proprietary pre-processing of simulation input data.
- The suitability of the OpenSCENARIO XML Synchronize Action for generating conflict situations under different initial conditions and parameters.
- The functional capabilities related to logical scenarios in OpenSCENARIO XML for parameter variations.
- The possibility to re-use the scenario descriptions and the post-processing of the simulation results from MIL over SIL to HIL.

After an initial implementation had proven OpenSCENARIO XML and OpenDRIVE capable of fulfilling these requirements, Vector decided to contribute its implementation of these scenarios open source on Github [VEC02].

The goal is to avoid duplicate work within the community and to foster discussions about the correct interpretation of the standards and scenario descriptions by Euro NCAP.

3 Implementation using OpenSCENARIO XML, OpenDRIVE and MDF

Euro NCAP provides test protocols free of charge that contain the descriptions of the tests including the scenario definitions. These are made available as textual descriptions and sketches, but also tables with parameters that shall be varied, like the speed of the vehicle under test. Furthermore, assessment protocols are available that describe how to calculate the ratings after the test conduct [EUR24c], [EUR24d]. The vehicle and environment simulator Vector DYNA4 [VEC01] natively supports both OpenSCENARIO XML and OpenDRIVE as simulation input formats. Hence, the implementation and simulation of the scenarios was possible without any intermediate conversion to proprietary input formats. The findings of the implementation can be summarized as follows:

1. NCAP static parts in OpenDRIVE
    - Simple straight roads and symmetrical intersections allow for easy creation of matching OpenDRIVE descriptions.
2. NCAP dynamic parts in OpenSCENARIO XML
    - The OpenSCENARIO Synchronize Action is used for the behavior descriptions in many of the scenarios: It synchronizes the target entity's arrival at a target position with the vehicle under test and was of immense value to model the desired collisions.
    - Together with this action, the extensive usage of parameters was key to fulfill the goal of avoiding proprietary pre-processing.
    - An apt choice of parameter definitions made it possible to translate the parameter tables from Euro NCAP directly into OpenSCENARIO XML variation files.
    - The initial set of developed and published scenarios has been made such that the largest possible set of features is already covered, especially regarding the movement of the vehicle under test relative to the target. Therefore, longitudinal, and perpendicular collisions with a straight moving vehicle under test are included as well as scenarios in which the vehicle under test makes a turn. OpenSCENARIO XML proved capable of all of these, with a minor issue as described in chapter 5.
3. NCAP assessment based on MDF results
    - ASAM MDF is used for the recorded data, when simulating the scenarios in DYNA4.
    - A Python-based reporting then automatically calculates the Euro NCAP ratings based on key performance indicators such as impact speed or collision avoidance.

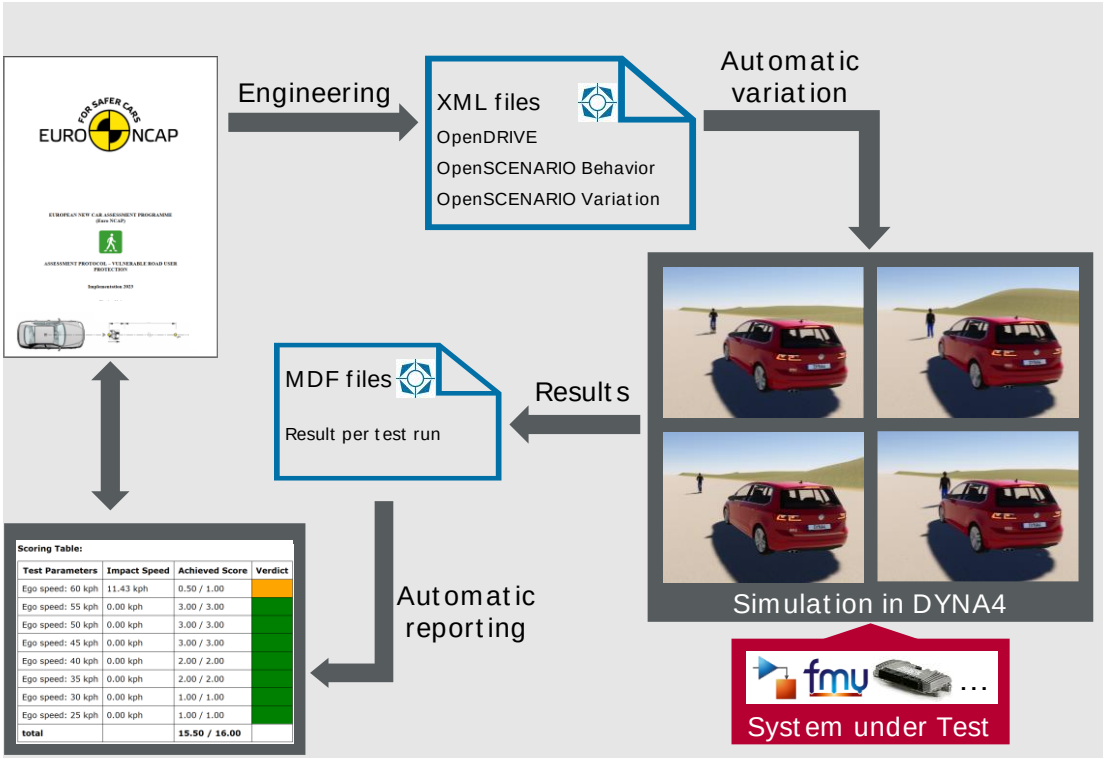


Figure 1: Workflow for creation, simulation, and reporting of Euro NCAP scenarios

The entire workflow from creation of the scenarios based on the Euro NCAP protocols up to the reporting is summarized in Figure 1.

4 Re-usability in different execution environments

As Figure 1 suggests, the system under test may be available in different stages throughout the function development and can range from a Simulink-based model or as an FMU (MIL) through control unit code (SIL) to real target control units (HIL). The interfacing to the system under test and the execution environment of the simulation change according to these stages. The DYNA4 models can be compiled for different environments such as Windows and Linux Executables, FMUs and various real-time hardware targets. Independent of the execution environment, the DYNA4 models directly ingest OpenSCENARIO XML and OpenDRIVE. In this way, a full re-usability of the scenario definitions is ensured. Only the parameter variation should be performed in an upstream step to avoid this calculation overhead when performing tests with real-time hardware and reduce the effort to handling the switching of parameter sets within the test automation. For example, Vector CANoe can be used for the test execution for SIL and HIL tests with DYNA4 models directly embedded. Test designs with Vector vTESTstudio then switch through the different pre-calculated parameter sets.

5 Future Work and Conclusion

As the implementation of the Euro NCAP scenarios in the OpenSCENARIO XML format is time-consuming, not all scenarios described in the test protocols by Euro NCAP have been implemented and published yet. The remaining scenarios will be added to the Github repository [VEC02] successively. As the currently implemented scenarios already cover most features, no further obstacles with regards to the standards are expected. However, during the implementation, a likely candidate for future standardization has been identified already: For scenarios in which the vehicle under test makes a turn at an intersection, Euro NCAP prescribes concatenated trajectories using straight, clothoid and arc elements. Such trajectories are supported by the latest OpenSCENARIO XML version 1.3.

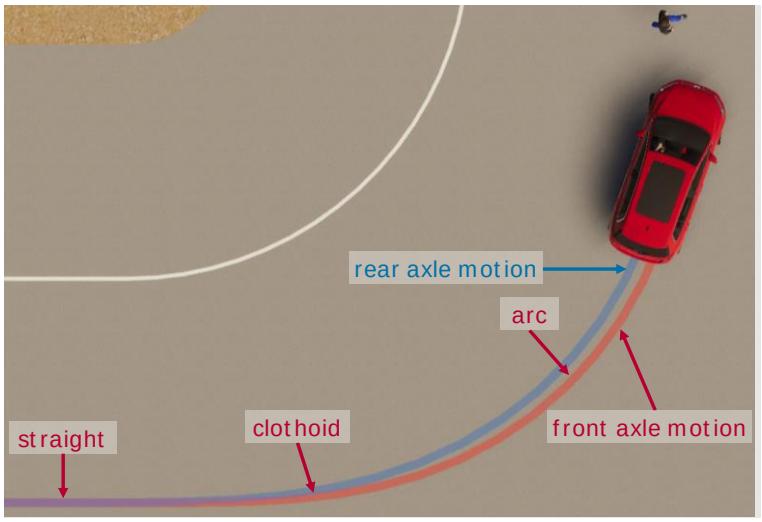


Figure 2: Front and rear axle motion when following a prescribed trajectory of a turning maneuver

OpenSCENARIO XML, however, uses the center of the rear axle as the control point for following trajectories, while Euro NCAP specifies the center of the front axle, which is a common reference coordinate system for vehicle dynamics simulation. Figure 2 highlights the difference of front and rear axle motion when following such trajectories. Until a feature supplement in the standard is possible, a standard-compliant but still proprietary controller property was introduced in DYNA4 to move the control point accordingly.

Another issue that has been identified does not affect the result of the scenario execution but would improve the style of the OpenSCENARIO XML files. As of now it is possible to store certain information in catalog files for improved reusability. This feature of OpenSCENARIO XML has been extensively used for the implementation. However, it is not possible to refer to certain values of the catalog entries. Therefore, certain values like the dimensions of the vehicle under test and the targets are duplicated among the catalogs and the actual scenarios, which deteriorates maintenance.

Altogether, the ASAM standards OpenSCENARIO XML and OpenDRIVE proved to be capable of modelling the scenarios described by Euro NCAP for the assessment of active safety systems. As desired when initiating the implementation of the scenarios, it was also possible to avoid any external pre-processing, even for the parameter variations. As a result of this and the architecture of DYNA4, it could also be proven that the standardized input formats are directly suitable for simulation execution in different environments for use-cases ranging from MIL over SIL to HIL tests. The same holds true for the post-processing of results as the simulations' output is independent of the execution environment.

6 References

[ASA24] ASAM e.V. (2024). ASAM OpenSCENARIO XML BS 1.3.0 Specification.

[EUR24a] Euro NCAP. (2024). Euro NCAP Test Protocol – AEB Car-to-Car systems, Implementation 2023, Version 4.3.1.

[EUR24b] Euro NCAP. (2024). Euro NCAP Test Protocol – AEB/LSS VRU systems, Implementation 2023, Version 4.5.1.

[EUR24c] Euro NCAP. (2024). Euro NCAP Assessment Protocol – Safety Assist Collision Avoidance, Implementation 2023, Version 10.4.1.

[EUR24d] Euro NCAP. (2024). Euro NCAP Assessment Protocol – Vulnerable Road User Protection, Implementation 2023, Version 11.4.

[FUN24] Funk, O. (2024). Simulative Untersuchung automatisierter Notbremsfunktionen, Masterarbeit. Hochschulföderation SüdWest.

[GER15] Gerstenberger, M. (2015). Dissertation - Unfallgeschehen an Knotenpunkten. Technische Universität München: Technische Universität München.

[VEC01] Vector Informatik GmbH. Euro NCAP: Test of AEB and FCW Functions with Virtual Test Drives. https://www.vector.com/ncap

[VEC02] Vector Informatik GmbH. OSC-NCAP-scenarios. https://github.com/vectorgrp/OSC-NCAP-scenarios

Virtueller Test in der praktischen Anwendung

M.Sc. Felix Strauß, Dipl. Ing. (FH) Alexander Merkel, Nic Eckstein

Electronics and Virtual Testing Solutions
Bertrandt AG
Birkensee 1
71139 Ehningen
Felix.Strauss@bertrandt.com
Alexander.Merkel@bertrandt.com
Nic.Eckstein@bertrandt.com

Abstract: Der Trend zum Software-Defined-Vehicle (SDV) erhöht zunehmend die Komplexität der Software im Fahrzeug. Dies erschwert auf der einen Seite die gewohnten linearen Absicherungs- und Integrationsprozesse. Auf der anderen Seite lässt sich die Komplexitätszunahme kaum noch durch den Einsatz von hardware-basierten Testplattformen beherrschen. Eine weitere Hardware-Skalierung erscheint bei wachsender Komplexität unrealistisch sowie unwirtschaftlich. Speziell der virtuelle Test mittels virtueller Steuergeräte (vECUs) bietet hier Abhilfe. Hervorzuheben ist, dass virtueller Test weit mehr als klassisches Software-in-the-Loop-Testing (SiL) bedeutet. Vielmehr kann über eine Integration von Basissoftware und ggf. einer zusätzlichen Chip-Emulation ein sehr realitätsnahes Abbild von Steuergeräten hergestellt werden. Eine ganzheitliche Absicherung von Fahrzeugsoftware bieten diverse Integrationsstufen virtueller Steuergeräte. Jede Stufe bietet sowohl Chancen als auch entsprechende Limitierungen. Diese müssen entsprechend bekannt und bewertet sein, um einen digitalen Absicherungsprozess aufbauen zu können. Das Hauptaugenmerk muss dabei auf einem holistischen Integrationsprozess liegen.

1 Motivation

In den letzten Jahren befindet sich die Automobilindustrie in einem tiefgreifenden Wandel. Die Megatrends Connected, Autonomous, Shared und Electric (CASE) transformieren das Fahrzeug hinsichtlich seiner Funktionen zu einem grundlegend neuen Produkt. Kundenanforderungen, die durch die Smartphone-Welt und autonome Fahrfunktionen geprägt sind, treiben die fortschreitende Digitalisierung und eine signifikante Erhöhung der Softwarekomplexität voran. Darüber hinaus erfordert der Übergang zur Elektromobilität Neuentwicklungen der soft- und hardwareseitigen Komponenten des Antriebsstrangs und bedingt neue Fahrzeugplattformen.

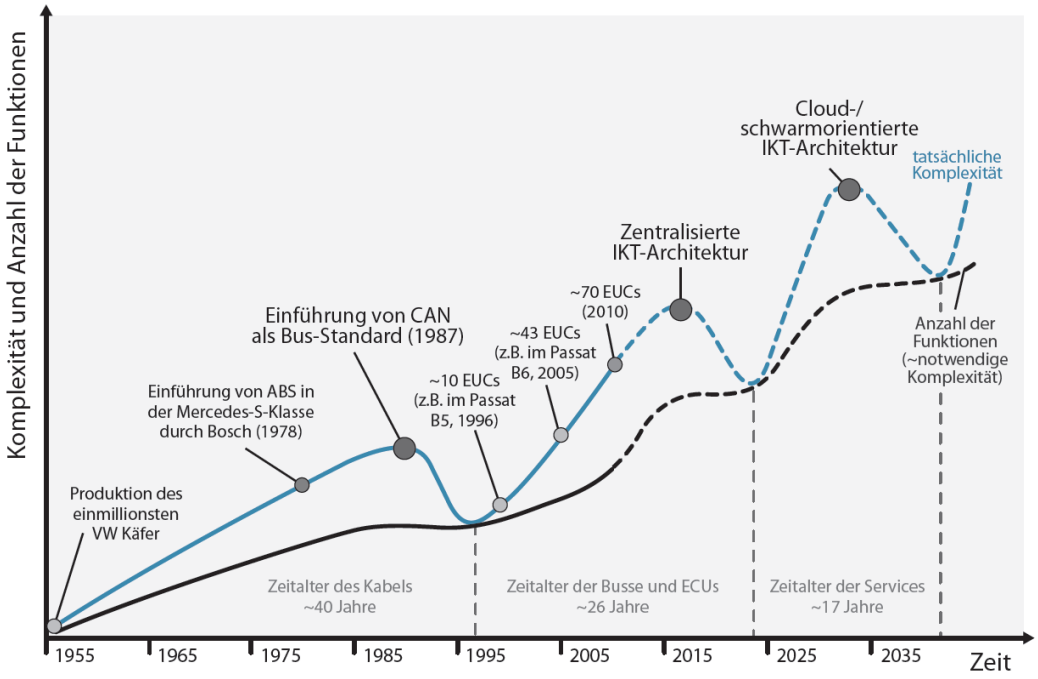


Abbildung 1: Komplexität und Anzahl Funktionen im Fahrzeug [FOR11]

Diese Trends führen zu einer Zunahme der Komplexität auf allen Ebenen (Abbildung 1): Die Steuergeräte werden leistungsfähiger und stärker vernetzt, die Softwarefunktionen umfangreicher. Die Anzahl der notwendigen Codezeilen nimmt exponentiell zu. Es ist zu beobachten, dass klassische Ansätze der E/E-Absicherung der notwendigen Geschwindigkeit und Menge nicht mehr Stand halten. Im Zuge dieser Entwicklungen manifestiert sich jedoch auch ein Umdenken in der E/E-Absicherung. Bertrandt leistet für zukünftige Absicherungsstrategien einen wertvollen Beitrag, indem der Fokus vor allem auf das Potential der virtuellen Absicherung gelenkt wird.

2 Stand der Technik

2.1 Virtualisierung

Unter Virtualisierung ist die Nachbildung eines Hard- oder Softwareobjekts durch Hilfe einer Abstraktionsschicht zu verstehen. In vielen Fällen spricht man auch von Hardwareabstraktion. Hiermit lassen sich virtuelle (d.h. nicht-physische) Geräte oder Dienste wie emulierte Hardware, Betriebssysteme oder Netzwerke erzeugen. Primäres Ziel der virtuellen Bereitstellung ist eine Unabhängigkeit von häufig sehr spezifischen und dadurch limitierten und teuren Zielgeräten zu erreichen. Vielmehr erfolgt die Bereitstellung auf weit verbreiteten Verbraucherendgräten, die in aller Regel aus einer x86-Prozessorarchitektur und einem Windows- oder Linux-Betriebssystem bestehen. Abbildung 2 zeigt diesen Prozess als Prinzip-Skizze.

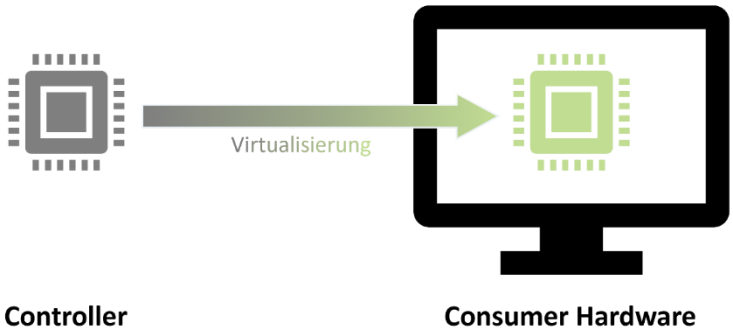


Abbildung 2: Prinzip-Skizze Virtualisierung

2.2 AUTOSAR Classic

Zur Herleitung der Steuergeräte-Virtualisierung, lohnt sich zunächst ein Blick in den AUTOSAR (AUTomotive Open System ARchitechture) Classic Standard. AUTOSAR ist eine Initiative verschiedener OEM und Zulieferer, die seit der Gründung im Jahr 2003 eine Referenzarchitektur für die Softwareentwicklung im Automotive-Bereich geschaffen hat [AUT22]. Diverse Ansätze der Abstraktion ermöglichen eine flexible Softwareentwicklung über Steuergerätegrenzen hinweg.

In erster Näherung definiert AUTOSAR eine Schichtarchitektur bestehend aus der *Applikationsschicht*, dem *Runtime Environment (RTE)* und der *Basissoftware (BSW)*. Die Applikationsschicht implementiert im Rahmen der Applikationssoftware (ASW) logische Fahrzeugzusammenhänge und Regelalgorithmen. Idealerweise ist sie hardwareunabhängig. Das *Runtime Environment*, eine Art Middleware, dient als Kommunikationsschicht zwischen einzelnen Softwarekomponenten (SWC) sowie zwischen *Applikationsschicht* und *Basissoftwareschicht*. Letztere stellt diverse Dienste zur Verfügung, die in drei Abstraktionsschichten unterteilt werden: *Service Layer*, *ECU Abstraction Layer* und *Microcontroller Abstraction Layer (MCAL)*.

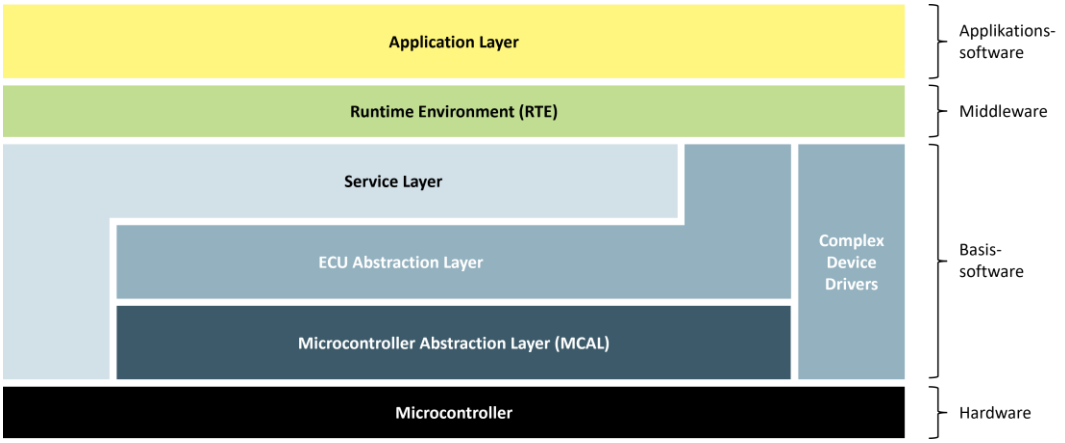


Abbildung 3: AUTOSAR Classic Architektur [AUT22]

Ein entscheidender Aspekt für die Steuergeräte-Virtualisierung ist die Trennung von möglichst hardwareunabhängigen Schichten und der darunterliegenden Hardware-Abstraktion, dem *MCAL*. Ein zweiter wichtiger Aspekt ist die strikte Abgrenzung von Funktionalitäten in Dienste und die hieraus resultierenden Schnittstellendefinitionen. Entlang dieser Schnittstellen können Funktionsblöcke gut abgegrenzt und logisch „geschnitten“ werden. Eine Ausnahme im Standard stellen die sogenannten *Complex Device Drivers (CDD)* dar. Diese ermöglichen die Vermischung der entkoppelten Schichten und stellen ein teilweise schwer zu überwindendes Hindernis für die Virtualisierung dar.

2.3 Klassifizierung virtueller Steuergeräte

Virtuelle Steuergeräte haben zum Ziel produktiven Code des Steuergeräteprojekts zu nutzen und diesen für eine, von der Zielhardware (Target) verschiedene, Plattform bereitzustellen. In aller Regel ist das Target des Steuergeräts ein Mikrocontroller und die Zielplattform des virtuellen Steuergeräts ein Windows oder Linux-System mit x86-Prozessorarchitektur. Ausgehend von AUTOSAR wurde folgende theoretische Klassifizierung von virtuellen Steuergeräten abgeleitet [PRO20] und kann als Orientierung gesehen werden:

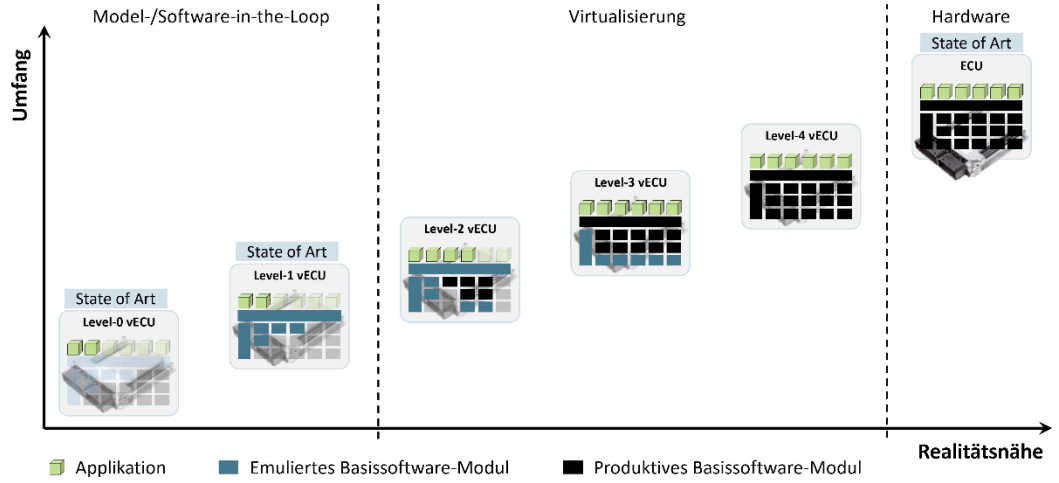


Abbildung 4: vECU-Level aus [PRO20]

Level-0 vECU

Level-0 ist vor allem im Modultest im Rahmen des Model-in-the-Loop (MiL) zu finden. Hierbei besteht die vECU aus einem oder mehreren Reglermodellen, welche zur Entwicklung der Algorithmen selbst genutzt werden, nicht jedoch aus produktivem Code. In der Softwareabsicherung spielen diese in aller Regel keine wesentliche Rolle und sollen hier vernachlässigt werden.

Level-1 vECU

Das Level-1, auch mit *High Cut* [VDA22] bezeichnet, wird durch einen logischen Schnitt entlang der RTE erreicht. Es handelt sich um eine Komposition aus einigen SWCs und der entsprechenden RTE. Einsatz findet hierbei nur produktiver Code. Im Rahmen dieser vECU werden zusätzlich sehr rudimentäre *System Services* emuliert, welche Speicherbereiche und eine Ablaufsteuerungen von Prozessen (Scheduling) bereitstellen. Diese vECU wird in der Regel für die funktionale Absicherung der spezifischen Systemfunktionen entworfen.

Level-2 vECU

Mit dieser vECU-Form wird eine erweiterte Emulation der Basissoftwareschicht integriert. Es gibt jedoch keine klare Definition für eine logische Schnittebene. Vielmehr orientiert sich die Emulation der BSW am Anwendungsfall. Einige Beispiele der Erweiterung zur Level-1 vECU sind: COM-Stack, Bus-Interfaces (CAN, Ethernet, etc.), non-volatile Memory oder Diagnosedienste. Die Level-2 vECU ermöglich bereits in sehr frühen Phasen Tests auf der Protokollebene.

Level-3 vECU

Das auch mit *Low Cut* bezeichnete Level-3 wird oberhalb der Hardwareschicht geschnitten und beinhaltet die gesamte Kommunikationsschicht der Basissoftware. Technisch gesehen setzt sich eine Level-3 vECU aus allen hardwareunabhängigen Anteilen der produktiven Software sowie einer Virtualisierung des MCAL und eines anteilig virtualisierten Betriebssystems zusammen. *CDDs* müssen häufig gesondert betrachtet und angepasst werden. Die Kommunikation dieser vECU erfolgt auf Grundlage der jeweiligen Busprotokolle. Die Level-3 vECU erweitert die Reichweite für die Absicherung deutlich bis in die Basissoftware hinein.

Level-4 vECU

Die *Binary Target vECU* ist binär identisch zum realen Steuergerät. Dies wird durch eine Emulation des Chipsatzes der Zielplattform (Target) auf einem klassischen x86-System erreicht. Diese stellt den Maschinenbefehlssatz für die Hardwaretreiber zur Verfügung. Erreicht wird das unter anderem mit sogenannten Kernel-based virtual machines (KVM). Bei der Level-4 vECU gibt es keinen Code-Unterschied zwischen virtuellem und realem Steuergerät.

3 Praktische Anwendungen

3.1 Potentiale und Limitierungen virtueller Steuergeräte

Jedes vECU-Level bedingt durch den logischen Schnitt unterschiedliche Potentiale und Limitierungen. Ausgehend von der technischen Kenntnis der unterschiedlichen Level können Potentiale und Limitierungen nun hergeleitet werden.

Einzelne Potentiale und Limitierungen sind in der folgenden Tabelle farblich nach dem Ampelsystem bewertet:

	Level-1	Level-2	Level-3	Level-4
Debugging durch Brake-Points / Anhalten der Simulation	🟢	🟢	🟢	🟢
Skalierung - Simulationszeiten (Echtzeitfaktor)	🟢	🟢	🟡	🟡
Skalierung - Parallelisierung	🟢	🟢	🟡	🟡
Zeitaufwand der Auslieferung	🟢	🟡	🟡	🟢
Funktionaler/logischer Schnitt von Steuergeräten	🟢	🟡	🟡	🟡
Basisfunktionen: Diagnose, Security und Busprotokolle	🔴	🟡	🟢	🟢
Determinismus / Realitätsnähe	🟡	🟡	🟡	🟢
Kosten	🟢	🟢	🟡	🟡
Kopplung mit Hardware-Setups	🔴	🟡	🟢	🟡
Fehlersimulation von Hardwarefehlern (SW-Rückfallebene)	🔴	🔴	🟢	🟡
Einsatz bei Treiberentwicklung und Compiler-Analyse	🔴	🔴	🔴	🟢
Flashen, Analyse von Speicher- und Prozessorauslastungen	🔴	🔴	🔴	🟢
Integration Non-Autosar/Legacy Code, CDDs	🔴	🔴	🔴	🟢

Tabelle 1: Potentiale und Limitierungen

Mit der Kenntnis der Potentiale und Limitierungen kann bewertet werden, welches vECU-Level für welchen Anwendungsfall genutzt werden kann oder definitiv ausgeschlossen werden sollte. Bei der Bewertung der einzelnen Level ist festzustellen, dass verschiedene Level sich sinnvoll ergänzen können. Im Rahmen einer Absicherungsstrategie muss bewertet werden, welche Anwendungsfälle einer großen Skalierung bedürfen und wo sich deswegen eine Virtualisierung lohnt. Dies kann beispielsweise bei einer Vielzahl von abzusichernden Varianten der Fall sein. Ebenso finden sich viele Testszenarien, bei denen die Absicherung an hardware-basierten Testplattformen eine Verschwendung der Ressource darstellt, da nur die reine Softwarefunktion getestet wird. Häufig ist eine gestaffelte Absicherung mit einer Level-1 vECU, einer Level-3 und einer realen ECU sehr zielführend. Je nach Softwareprojekt kann jedoch auch eine Level-4 vECU unabdingbar sein. Dies kann beispielsweise bei Mikroprozessorarchitekturen und AUTOSAR Adaptive Anteilen der Fall sein.

3.2 Einsatz virtueller Steuergeräte

Der Einsatz virtueller Steuergeräte entlang der Integrationsstufen im V-Modell lässt sich aus den Potentialen und Limitierungen ableiten und ist in Abbildung 5 prinzipiell dargestellt. Level-4 findet sowohl im Entwurf als auch im Systemtest Einsatz. Level-1 und Level-3 decken gemeinsam nahezu den gesamten rechten Teil des V's ab.

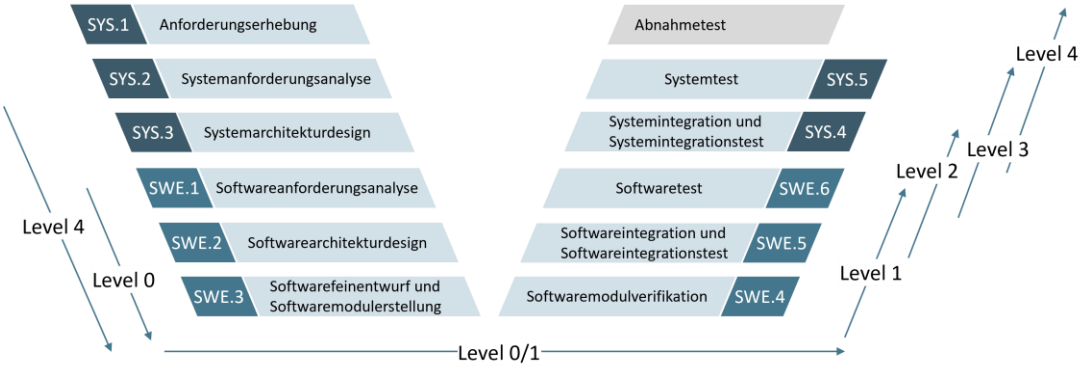


Abbildung 5: Einsatz virtueller Steuergeräte im V-Modell

3.3 Anwendungsbeispiele

Die zentrale Frage, die beim Einsatz von virtuellen Steuergeräten zu beantwortet ist, ist die folgende: Was ist der hauptsächliche Testfokus: *die funktionale Softwarewirkkette*, am Ende „nur" über den Hardwaretreiber an ein Kommunikationsmedium (Bus) angeschlossen oder doch der *Hardwaretreiber und die verbundene Physik selbst*? In aller Regel (außer in der Treiberentwicklung) ist ersteres die Antwort. Betrachtet man die Wirkkette innerhalb eines AUTOSAR-Steuergeräts (Abbildung 6), lassen sich solche Fälle mit virtuellen Steuergeräten ab Level-3 ideal absichern, da maximal die Treiberschicht virtualisiert wird.

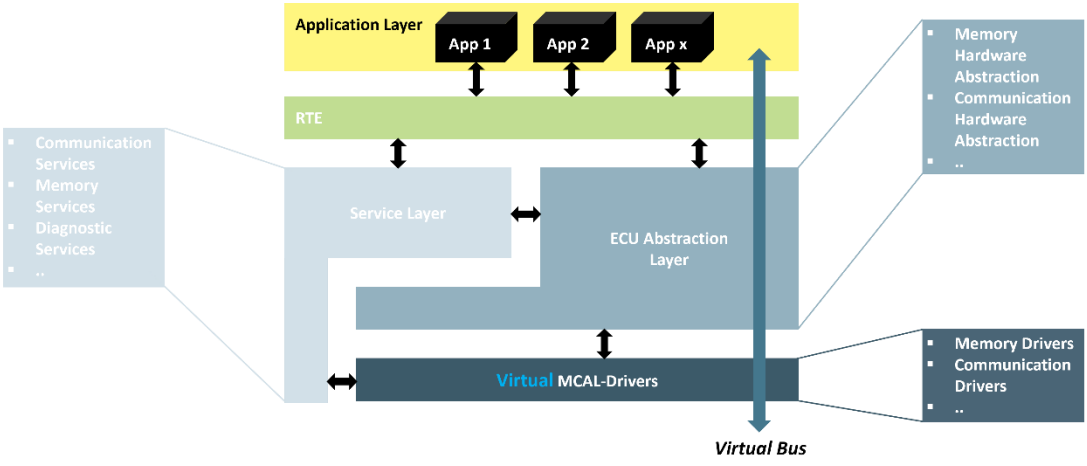


Abbildung 6: Prinzip-Skizze virtualisierter MCAL im Level-3 (CDD vernachlässigt)

Im Folgenden werden drei anschauliche Fälle beschrieben. Zur Vorbereitung wird das virtuelle Steuergerät jeweils in eine Simulation eingebettet, welche alle notwendigen Schnittstellen bedienen kann. Im Falle von Sensorschnittstellen wird der Regelkreis durch die Simulation der Regelstrecke geschlossen. Busschnittstellen werden mit virtuellen Bussen verbunden und mit einer Restbussimulation (Mock) ergänzt.

a) **Komponententest: Diagnose und Buskonformität**
Der erste Fall, siehe Abbildung 7a), beschreibt einen Komponententest von Basis- bzw. Querschnittsfunktionen wie Diagnose oder Buskonformität. Hierbei werden im ersten Fall automatisiert die Diagnose-Parameter (basierend auf dem ODX) getestet. Der Bustreiber kann hier nicht mit abgesichert werden, da dieser virtualisiert ist. Das ist jedoch nicht weiter schlimm, da der Treiber nur „Mittel zum Zweck“ ist und nicht Teil des eigentlichen *System-under-Tests*. Bei den Buskonformitätstests müssen Abstriche bei Buskurzschlüssen oder Ähnlichem gemacht werden, die Protokolltests sind jedoch ausnahmslos möglich (bspw. Service-Discovery).

b) **Fehlerspeichertest**
Beim Fehlerspeichertest werden häufig Fehlerzustände hervorgerufen und anschließend die Reaktion im Fehlerspeicher überprüft. Der relevanteste Anteil dieser Funktion findet nicht in RAM oder ROM statt, sondern in den Services der höherliegenden Schichten. Somit sind auch diese Tests mit virtuellen Steuergeräten durchführbar. Ganz besonders gut lassen sich die Simulationen von beliebigen Fehlerzuständen darzustellen, welche mit Realteilen oder im Fahrzeug deutlich schwieriger ist.

c) **Service Oriented Vehicle Diagnostics (SOVD)**
Service Oriented Vehicle Diagnostics ermöglichen Szenarien wie Remote-Diagnose. Hierzu wird ein Fahrzeug oder das entsprechende Connectivity-Steuergerät mit dem OEM-Backend verbunden. Solche Zugriffe erfordern beim Test einen sehr robusten kabellosen Zugriff in den Testumgebungen. Mit virtuellen Steuergeräten lassen sich solche robusten Zugriffe aus dem Backend sehr einfach umsetzen. Da es sich bei SOVD um ein reines Softwarefeature handelt ist es sehr gut virtuell testbar.

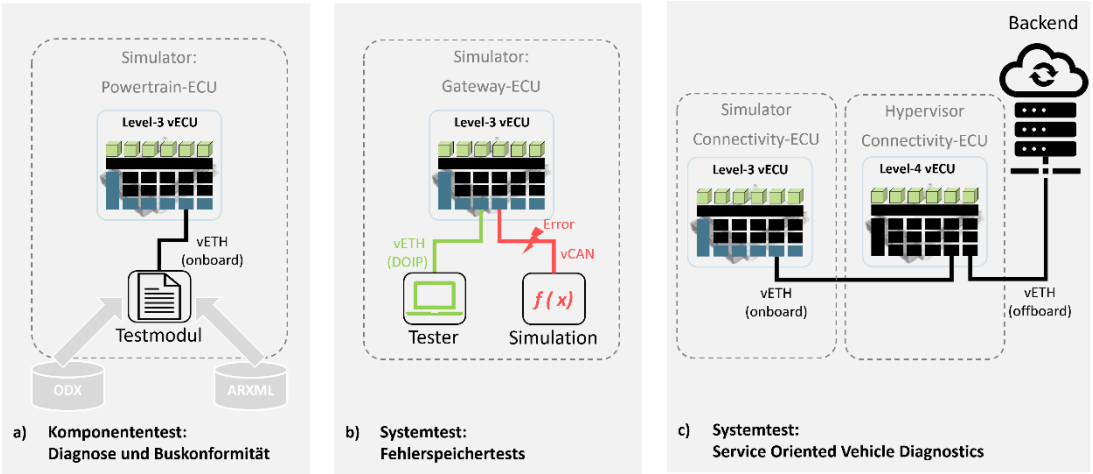


Abbildung 7: Prinzip-Skizze beispielhafter Anwendungen

4 Zusammenfassung und Ausblick

In diesem Beitrag wurde die Technologie virtueller Steuergeräte (vECUs) detailliert hergeleitet. Eine zentrale Rolle bei der Entwicklung dieser virtuellen Steuergeräte spielt der AUTOSAR-Standard. Verschiedene Stufen von vECUs bieten unterschiedliche Potenziale und Limitierungen. Anhand praxisnaher Beispiele wurde demonstriert, dass die virtuelle Absicherung ein erhebliches Potenzial über die Absicherung von Applikationsfunktionen hinaus bietet. Insbesondere bei der Absicherung von Basisfunktionen zeigen vECUs großes Potenzial. Mit vECUs ab Level-3 können viele Basisfunktionen frühzeitig und in hoher Qualität abgesichert werden. Dies trägt nicht nur zur Qualität der nachfolgenden Integrationsstufen bei, sondern kann auch die Entwicklung der Basissoftware beschleunigen. Entscheidend ist, dass die Absicherung der Funktionen bereits während des Entwurfs der Softwarearchitektur berücksichtigt wird.

Neben der vECU-Technologie ist die Implementierung eines durchgängigen und konsistenten Absicherungsprozesses über ein entsprechendes Testkonzept und Testmanagement von zentraler Bedeutung. Dieser Prozess sollte einem umfassenden Ansatz folgen. Es ist essenziell, bereits im Testkonzept zu entscheiden, welche Anforderungen mit virtuellen Steuergeräten getestet werden können und welche den Einsatz von Hardware erfordern. Um diese Potenziale zu nutzen, ist ein ganzheitlicher Entwicklungsansatz notwendig. Durch konsequente Verfolgung dieser Ansätze lassen sich nicht nur signifikante Kosteneinsparungen realisieren, sondern auch die Entwicklungsgeschwindigkeit erheblich steigern. Vor diesem Hintergrund kann das virtuelle Steuergerät zu einem entscheidenden Faktor in der zukünftigen Fahrzeugentwicklung werden.

5 Literaturverzeichnis

[AUT22] AUTOSAR. (kein Datum). *autosar.org*. Abgerufen am 18. 3 2023 von https://www.autosar.org/

[FOR11] ForTISS GmbH. (2011). *Mehr Software (im) Wagen: Informations- und Kommunikationstechnik (IKT) als Motor der Elektromobilität der Zukunft.* ForTISS GmbH. Abgerufen am 18. 3 2024 von https://mediatum.ub.tum.de/doc/1287138/270280.pdf

[PRO20] prostep IVIP. (2020). Requirements for the Standardization of Virtual Electronic Control Units (V-ECUs). *Smart Systems Engineering*.

[VDA22] Verband der Automobilindustrie e. V. (VDA). (2022). VDA 710 Software-in-the-Loop (SiL) Standardisierung (Version 07/2022). Verband der Automobilindustrie e. V. (VDA).

Virtuelle Evaluation von ADAS-Funktionen mit offenen Standards am Beispiel von Euro NCAP Szenarien

Karl Schreiner, Fabian Otter, Dennis Otte, Marek Bachmann und Frank Beutenmüller

TWT GmbH Science & Innovation, Industriestraße 6, 70565 Stuttgart, Germany
research@twt-gmbh.de

Zusammenfassung. Der Vortrag stellt die Integration der standardisierten Euro NCAP-Test Szenarien für den Kollisionsschutz von Fußgängern und Fahrradfahrern in Tronis® vor. Die Integration der standardisierten Euro NCAP-Test Szenarien zielt darauf ab, die Entwicklung automatisierter und vernetzter Fahrfunktionen zur Steigerung der Verkehrssicherheit zu verbessern und zu vereinfachen. Tronis® ist eine auf der Unreal-Engine basierende Plattform für virtuelles Prototyping und Absicherung von Fahrerassistenzsystemen und bietet die Möglichkeit, die Erfassung von Umgebungsdaten mittels Kamera sowie Radar- und Lidar-Technologien realitätsnah zu simulieren. In Kombination mit präziser Modellierung und Emulation realer Umgebungs-, Verhaltens- und Bewegungsdaten ist Tronis® in der Lage, kritische Unfallszenarien zwischen Fahrzeugen und Fußgängern oder Fahrradfahrern realitätsnah zu bewerten. Dies bietet den einzigartigen Vorteil, kritische Unfallszenarien zu evaluieren, die in der Realität aus ethischen Gründen nicht durchführbar sind. Weiterhin kann durch den Einsatz von Tronis® die Entwicklung entsprechender Systeme beschleunigt und die Notwendigkeit realer Prototypen reduziert werden. Tronis® entkoppelt das Testing von der Durchführung aufwändiger realer Fahrtests durch eine virtuelle Einspeisung der Umgebungsbilddaten an die Kameras und Sensoren der Fahrerassistenzsysteme.

1 Einleitung und Motivation

Die Automatisierung und virtuelle Simulation von Fahrerassistenzfunktionen in digitalen Simulationssoftware gewinnt immer mehr an Bedeutung für die Automobilindustrie. Im Fokus dieser Entwicklung steht die Notwendigkeit, Fahrerassistenzsysteme unter realistischen Bedingungen zu testen und zu validieren, um ihre Zuverlässigkeit und Effektivität zu gewährleisten. Der Aufwand für die Durchführung von realen Feldtests ist sehr hoch, da sie eine umfangreiche Planung, Organisation und oft auch den Aufbau spezieller Testumgebungen erfordern. Hinzu kommen die Kosten für Personal, Material und die Instandhaltung der Testeinrichtungen. Digitale Prüfgelände ermöglichen eine detailgetreue Nachbildung und Resimulation von realen Verkehrsszenarien, ohne

dass aufwändige physische Tests durchgeführt werden müssen. Diese virtuellen Umgebungen bieten eine sichere und effiziente Plattform für die Fahrsimulation, auf der die Assistenzsysteme unter verschiedensten Bedingungen getestet werden können. Die Integration solcher digitalen Gelände kann beispielsweise mit der Simulationssoftware Tronis® erfolgen [1].

1.1 Problemstellung

In der Entwicklung moderner Fahrerassistenzsysteme sind Systemtests am Gesamtverbund "Fahrzeug" im Zuge der Entwicklung unerlässlich. Dies betrifft sowohl entwicklungsbegleitende Fahrversuche als auch Abnahmetests für die Sicherstellung der abschließenden Serienreife.
Die Hochrüstung und Inbetriebnahme von Prototypen, die Durchführung umfangreicher Testkampagnen auf verschiedenen Strecken und unterschiedliche Bedingungen erfordern erhebliche finanzielle und personaltechnische Ressourcen. Eine weitere Herausforderung ist die begrenzte Reproduzierbarkeit der Ergebnisse. Unterschiedliche Wetterbedingungen, Fahrstile und Straßenverhältnisse können die Testergebnisse beeinflussen, was die Vergleichbarkeit erschwert. Zudem bestehen Risiken für die Sicherheit der Testfahrer und anderern Verkehrsteilnehmer, insbesondere bei Tests unter extremen Bedingungen oder bei der Erprobung neuer Technologien. Des Weiteren ist durch die fortschreitende Globalisierung und die zunehmende Dezentralisierung der Entwicklung und Produktion die lokale Verfügbarkeit von Prototypenfahrzeugen häufig nicht gegeben. Frühe Tests in der Vorentwicklung, während noch keine realen Fahrzeuge verfügbar sind, können mit virtueller Simulation zu extremen Kosteneinsparungen im späteren Verlauf führen, da Probleme in der Software früh erkannt werden (Shift-Left-Approach).

2 Virtuelle Simulation

Bei der Simulation mit einer Game-Engine, wie beispielsweise mit der Unreal Engine von Epic Games [2], ist zunächst eine Abwägung der Eigenschaften dieser Technologie hervorzuheben. Die Engine ist darauf optimiert, mindestens 30 Updates des Simulationszustands pro Sekunde sicherzustellen. Dieser Wert kann auch höher liegen, moderne Spiele streben 60 bis 120 Frames/s an. Dies schränkt sowohl die zeitliche Auflösung, als auch die Genauigkeit der physikalischen Berechnungen ein. Das System nimmt Ungenauigkeit in Kauf, um die gewünschte Framerate zu erreichen. Allerdings liefert die Software dafür eine Soft-Realtime Simulation und eine immersive Darstellung der simulierten Umgebung.
Diese Echtzeitfähigkeit, also die Fähigkeit, die geforderten Simulationsdaten zuverlässig zeitgerecht bereitstellen zu können, ermöglicht es, eine geschlossene Feedbackloop mit externer Soft- oder Hardware zu implementieren, häufig sogar auf dem selben Computer. Dabei kann die Engine den Output dieses Systems direkt z.B. zur Steuerung des virtuellen Fahrzeugs nutzen. Dies kann auch über eine Middleware wie ROS oder APOLLO geschehen.

Die Immersion wird relevant, wenn die Reaktionen oder das Verhalten von Menschen untersucht werden soll. Die psychologische Wirkung eines Experiments auf einen menschlichen Beobachter kann nur dann verwertbare Ergebnisse liefern, wenn die Illusion gut genug gelingt.
Dem entgegen stehen Simulationsumgebungen wie Ansys oder Comsol, die physikalische Korrektheit und Präzision zum Preis eines hohen Berechnungsaufwands versprechen. Sie benötigen für die Berechnung einer simulierten Sekunde i.A. ein Vielfaches dieser Zeit und können somit keine geschlossene Feedbackloops in Echtzeit betreiben. Eine überzeugende Immersion ist in der abstrakten Darstellung dieser Software nicht zu erwarten.

3 Szenariengenerierung

In der virtuellen Absicherung ist die Entwicklung von Szenarien ein wichtiger Prozess. Je nach Use-Case erfordert die Simulation eine detaillierte Modellierung der Umgebung, damit dynamische Elemente unter realitätsnahen Bedingungen getestet werden können. Eine wichtige Anforderung ist die Interoperabilität und die technische Eingliederung in bestehende Prozessketten. Im folgenden wird auf die Anwendung offener Standards für die Szenariengenerierung und die Erstellung von Verkehrsraummodelle eingegangen.

3.1 Szenarien aus Open Standards

Mit dem Einzug von trainierten Modellen auf Basis von Maschine Learning oder Neuronalen Netzwerken entsteht ein hoher Bedarf an Kartenmaterial, an dem die Modelle trainiert werden können. Dabei ist es wichtig, dass der Datensatz ausreichend Variation aufweist. Dies kann auch durch Parametrisierung erreicht werden. Die Association for Standardization of Automation and Measuring Systems (ASAM) [3] stellt für diesen Zweck die Standards OpenDRIVE, OpenSCENARIO und OpenCRG bereit. Auf dieser Basis können Szenarien und Trainingsdaten zwischen verschiedenen Umgebungssimulationen ausgetauscht werden und die Reproduzierbarkeit gesichert werden. Mit dem Open Simulation Interface (OSI) steht zudem ein Standard für die Kommunikation mit einer Co-Simulation bereit.
Diese Standards sind für ein Ökosystem für Umgebungssimulationen von hoher Bedeutung. Für eine erfolgreiche Kooperation von mehreren Simulationsteilnehmern ist eine gemeinsame Ground Truth unerlässlich. Zudem können Straßenzüge, die einmal in OpenDRIVE implementiert wurden, immer wieder für unterschiedliche Experimente genutzt werden. Sie sind leicht zu parametrisieren.
Die deklarative Umgebungsbeschreibung bestehender Dokumente ist allerdings i.A. nicht auf eine fotorealistische Darstellung dieser Umgebung ausgelegt. Häufig enthalten sie wenig oder keine Informationen über statische Objekte oder Gebäude. Die Angaben beschränken sich typischerweise auf Bounding Boxen. Dies stellt die Entwickler vor die Herausforderung, die fehlenden Elemente (möglichst

automatisiert) zu ergänzen. Allerdings erleichtert die deklarative Beschreibung auch die Parametrisierung der Karten.
Zunächst muss für die Simulation ein Mesh für das Straßennetzwerk erzeugt werden. Dies ist relativ einfach, da die Beschreibung durch die Standards hier sehr detailliert ist. Eine Herausforderung liegt darin, dass Geometrien häufig überdefiniert sind. So wird in OpenDRIVE die Oberfläche von Kreuzungen durch überlagerte Lanes beschrieben. Dies führt jedoch zu visuellen Artefakten und kann zu Problemen mit den virtuellen Sensoren führen.

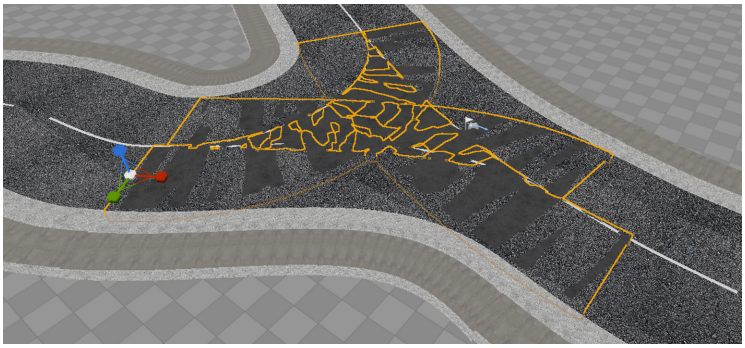


Abb. 1. Aus einer OpenDrive-Datei erstelltes Netz. Überlappende Straßen führen zu visuellen Artefakten auf der Oberfläche

Eine weitere Herausforderung liegt in der Gestaltung der Umgebung. Spezifisch Gebäude, Vegetation am Straßenrand oder verkehrsrelevante Objekte wie Bushaltestellen und deren Materialeigenschaften wie Reflexion benötigen eine detailliertere Beschreibung um dem angestrebten Detailgrad zu erreichen. Eine Stärke der Game-Engines ist die realistische Darstellung Reflexionen und Schattenwürfe zur Laufzeit.

3.2 Erstellung eines Verkehrsraummodells

Für eine ganzheitliche Darstellung der Szene in einer virtuellen Simulation benötigt die Umgebung oder der Akteur (zum Beispiel in Form von Sensoren) ein physikalisch parametrisiertes Verkehrsmodell, welches die Umgebung des Fahrzeugs und den Szenenkontext vollständig beschreibt. Weitere Daten sind Umweltinformationen oder Gebäude und Städtemöbel, die im Standard CityGML abgebildet werden, Katasterdaten und digitale Geländemodelle. CityGML ist ein Austauschformat und Datenmodell für 3D-Stadt- und Landschaftsmodelle und bietet eine wichtige Grundlage für Visualisierungen und Anwendungen im 3D-Kontext. Im Forschungsprojekt LevelUp werden Methoden zur automatisierten Erstellung und Integration von Level of Detail 3 (LOD3) Verkehrsraum Modellen entwickelt.
Die Prozesskette der Digitalisierung des Verkehrsraummodells besteht aus der

Datenerfassung von rohen Lidar Point Clouds, der Datenverarbeitung und Datenintegration. Bei der Erhebung von Rohdaten werden die Daten mithilfe von KI-assistierten Verfahren annotiert, welches für größere Datenmengen skaliert werden kann. Die Datenverarbeitung kombiniert eine Methode zur automatisierten Objekt-Rekonstruktion aus Fusionierten- oder Rohdaten und Methoden zur Objekterkennung. Im Bereich der Datenintegration liegt der Schwerpunkt auf ein automatisiertes Tool für die Integration verschiedener Datenquellen aus Kataster, Geländemodelle, Luftbilder, Gebäudemodelle, HD-Karten, Landnutzung und Straßenmodelle in ein ganzeinheitliches Verkehrsraummodell.

Abb. 2. Beispiel importiertes Verkehrsraummodell in Unreal Engine

Abbildung 2 zeigt den Import am Beispiel der Stadt München, Schwabing-Freimann. Die Daten werden aus verschiedenen Quellen im Rahmen der Vorprozessierung zusammengestellt und können durch weitere Informationen beliebig ergänzt werden. Die Bereitstellung der erforderlichen Daten wird durch eine unabhängige Datenquelle in Form eines QGIS Server als Backend realisiert [4]. Im Rahmen des Projektes entwickelt TWT Plugins für Unreal Engine, welches die Integration von Geodaten aus einem GeoInformationsSystem (GIS) ermöglicht. Die Hauptkomponente erlaubt eine Verbindung zu einem Server und ruft die Geodaten über standardisierte Web Services ab. Dies wurde exemplarisch bereits erfolgreich realisiert. Derzeit werden folgende Services durch das Plugins unterstützt: WMS (WebMapService), WCS (WebCoverageService) und WFS (WebFeatureService).

Das Plugin übernimmt die automatische Koordinatentransformation zwischen Geokoordinaten und dem Koordinatensystem aus Unreal Engine. Bei Simulationsstart werden Chunks von Terraindaten generiert und mit entsprechend selektiertem Inhalt gefüllt. Ziel von LevelUp ist Entwicklung eines LOD3-digitalisierten Verkehrsmodell welches als Digitaler Zwilling, Simulatorunabhängig oder als Datenerzeugung für externe Algorithmen verwendet werden kann. Durch diese hohe Detail-tiefe von hochauflösenden 3D-Objekten

kann die virtuelle Validierung reale Szenarien genau simulieren, bevor sie in realen Fahrzeugen eingesetzt werden.

4 Testing SiL HiL

In diesem Kapitel werden zunächst allgemeine aktuelle Verfahren sowie Richtlinien und Gesetze im Bereich des Testens von Fahrerassistenzsystemen erläutert. Anschließend wird auf konkrete Testszenarien, insbesondere die NCAP-Test-Szenarien, eingegangen und deren Integration in die Tronis-Simulationsumgebung dargestellt.

4.1 Physische Tests

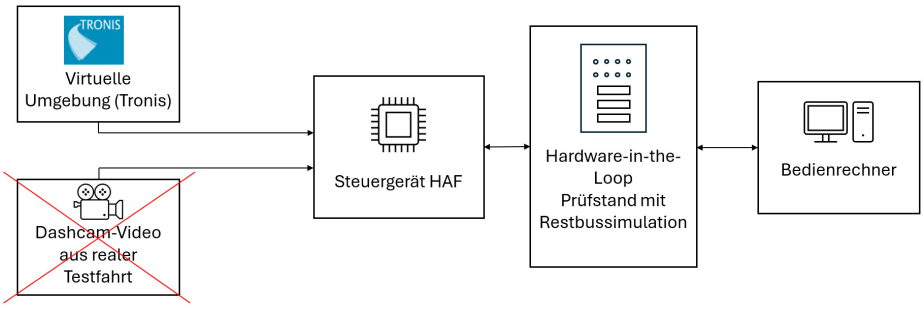


Abb. 3. Aufbau eines Videoprüfstands mit virtuellen Eingangsgrößen

Die EU-Verordnung 2019/2144 vom 27. November 2019 schreibt für neu typgenehmigte Personenkraftwagen seit Mitte 2022 und für alle neu zugelassenen PKWs seit Mitte 2024 vor, diese mit diversen Fahrerassistenzsystemen auszustatten[1] Dazu gehören Verkehrszeichenerkennung, Notbremsassistenten, Spurhalteassistenten und weitere Systeme. Die Verordnung spezifiziert im Detail die geforderten Eigenschaften dieser Sicherheitssysteme. Als Beispiel wird im folgenden die Verkehrszeichenerkennung (kurz: VZE) herangezogen. Sie muss in der Lage sein, zu jedem Zeitpunkt der Fahrt die aktuell gültige Geschwindigkeitsbegrenzung über Kameras in Verbindung mit Kartendaten bereitzustellen und diese Informationen sowohl dem Fahrer als auch dem Intelligenten Geschwindigkeitsassistenten zur Verfügung zu stellen.
Zur Absicherung der Funktionsweise ist es notwendig, tausende Kilometer in allen relevanten Märkten im realen Fahrbetrieb zurückzulegen. Nur somit konnte bisher eine zuverlässige Erkennungsrate gewährleistet werden. Unterstützend gab

[1] EU-Verordnung 2019/2144 vom 27. November 2019 (https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=celex%3A32019R2144)

es bisher die Möglichkeit, die reale Fahrt mit Dashcams aufzunehmen und das Videomaterial im späteren Verlauf als Eingabegröße zum Testen verschiedener Softwarestände und Applikationen zu nutzen (Video-Prüfstand).
Die virtuelle Absicherung mit Tronis erlaubt es nun, den Großteil der Fahrstrecke virtuell zurückzulegen. Über die Simulation können Verkehrszeichen aus allen relevanten Ländern und in verschiedensten Zuständen in die Software zur Verkehrszeichenerkennung eingespielt werden. Abbildung 3 zeigt schematisch den Aufbau eines Prüfstands für hochautomatisiertes Fahren (HAF) unter Einsatz virtueller Umgebungsdaten.
Eine weitere Anwendungsmöglichkeit ist die Integration externer realer Sensorik durch Hardware-in-the-Loop (HiL) Systeme. Dies ermöglicht es den Anwendern, das Verhalten von realer Sensorik unter verschiedenen Bedingungen zu testen. Dies bedeutet, es steht kein reales Fahrzeug und keine reale Umgebung zur Verfügung, sondern ein virtuelles Fahrzeug [5]. Das virtuelle Fahrzeug liefert Informationen wie Position, Orientierung und bietet eine Schnittstelle um die Ergebnisse aus der externen realen Hardware wiederzuverwenden.

4.2 NCAP

Das European New Car Assessment Programme (Euro NCAP) ist eine unabhängige Organisation, die 1997 in London gegründet wurde, um die Sicherheit von Neuwagen in Europa zu bewerten.
NCAP-Szenarien definieren standardisierte, reproduzierbare Tests, die reale Verkehrssituationen darstellen. Diese Szenarien umfassen typische Unfallkonstellationen wie das plötzliche Auftauchen von Fußgängern oder Fahrradfahrern, was die Leistungsfähigkeit von Kollisionsschutzsystemen unter praxisnahen Bedingungen prüft. Abbildung 4 zeigt ein NCAP Szenario in dem ein Fußgänger aus der Sichtverdeckung zwischen zwei Autos spontan auf die Straße tritt. Für Automobilhersteller sind diese Tests von hoher Relevanz, da die Ergebnisse sowohl die Sicherheitsbewertung ihrer Fahrzeuge als auch deren Marktakzeptanz beeinflussen.
Zur Evaluation und Entwicklung neuer und leistungsfähiger Kollisionsschutzsysteme wird es sinnvoll und teilweise unerlässlich sein, die NCAP-Tests in einer virtuellen Umgebung durchzuführen. Fahrradfahrer und Fußgänger werden in den klassischen physischen Tests durch Puppen repräsentiert, die jedoch nicht in der Lage sind, das komplexe und dynamische Verhalten realer Menschen nachzubilden. Abbildung 5 zeigt die Implementierung des "CPNC-50" NCAP Szenarios. Für moderne, KI-basierte Kollisionsschutzsysteme sind in solchen Szenarien Nuacen im Verhalten entscheidend, da sie die Differenzierung ermöglichen, ob ein Fußgänger die Straße betritt oder nicht. Da es ethisch nicht vertretbar ist, diese Tests mit realen Personen durchzuführen, bietet hier die virtuelle Umgebung von Tronis eine sichere und realitätsnahe Alternative. Mithilfe immersiver Augmented Reality und Virtual Reality lassen sich menschliche Verhaltensmuster präzise simulieren, ohne die Sicherheit der Probanden zu gefährden. Dies ermöglicht eine genauere und praxisnähere Bewertung der Kollisionsschutzsysteme.

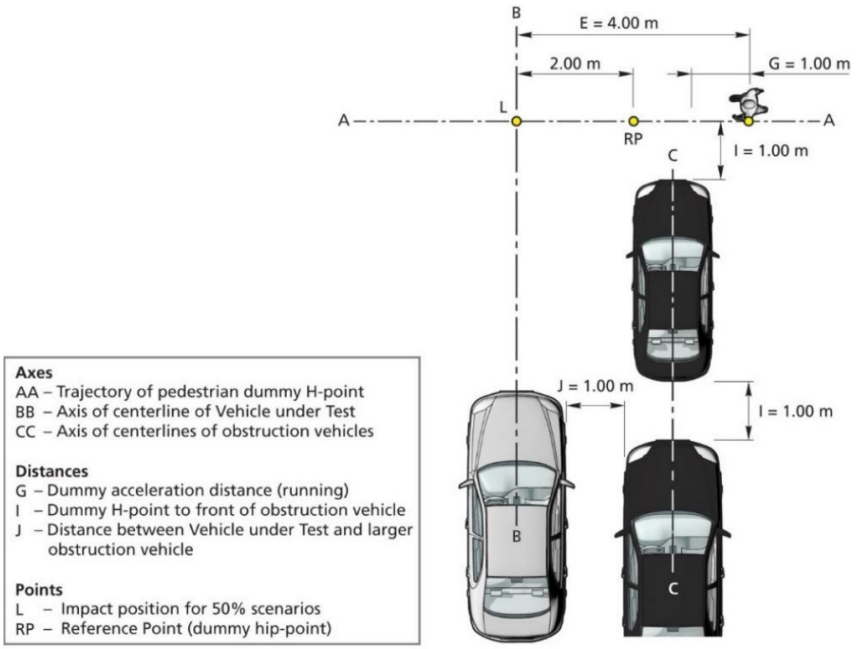


Abb. 4. Car-to-Pedestrian Nearside Child 50% (CPNC-50)"Kollisionsszenario aus dem Test Protocol for AEB VRU Systems"des Euro NCAP

Abb. 5. CPNC-50 modelliert in Tronis

5 Zusammenfassung

Die Integration der Euro NCAP-Test Szenarien in virtuelle Testumgebungen zeigt auf, wie virtuelle Absicherung die Entwicklung und das Testen von Fahrerassistenzsystemen revolutionieren kann. Durch die realitätsgetreue Simulation von Verkehrssituationen und die Möglichkeit, Tests, die den Menschen betreffen, sicher durchzuführen, entsteht ein wesentlicher Wettbewerbsvorteil. Traditionelle Testmethoden erfordern umfangreiche Ressourcen und sind oft zeitintensiv. Virtuelle Tests hingegen eröffnen den Zugriff auf einen großen Pool von parametrisierbaren Karten und Szenarien und ermöglichen eine schnellere und kosteneffizientere Durchführung, ohne dabei die Qualität der Ergebnisse zu beeinträchtigen. Darüber hinaus garantieren sie eine hohe Reproduzierbarkeit der Tests. Zukünftige Arbeiten sollten sich auf die Weiterentwicklung der Vehicle-to-Everything-Kommunikation (V2X) und die Integration weiterer offener Standards konzentrieren. Die Vernetzung von Fahrzeugen untereinander und mit der Infrastruktur bietet großes Potenzial, die Sicherheit und Effizienz automatisierter Fahrfunktionen weiter zu steigern. Virtuelle Testumgebungen in Kombination mit den Euro NCAP-Test Szenarien werden zukünftig ein unverzichtbares Werkzeug für die moderne Fahrzeugentwicklung darstellen.

Danksagung

Diese Arbeit wurde teilweise vom Bundesministerium für Bildung und Forschung (BMBF) aufgrund eines Beschlusses des Deutschen Bundestages im Rahmen des Projektes „Entwicklung einer Methode zur automatisierten Erstellung und Integration eines Level-of-Detail 3 (LOD3) Verkehrsraum Modells“ (LevelUP) gefördert.

Literatur

[1] Karl Kufieta und Michael Ditze. „A virtual environment for the development and validation of highly automated driving systems“. In: *Bargende M., Reuss HC., Wiedemann J. (eds) 17. Internationales Stuttgarter Symposium*. Vieweg: Springer Vieweg, Wiesbaden (2017), 2017, S. 1391–1401.

[2] *Unreal Engine Webseite*. Letzter Zugriff: 11. September 2024. 2024. URL: `https://www.unrealengine.com/de`.

[3] ASAM. *Standards - ASAM*. Letzter Zugriff: 11. September 2024. 2024. URL: `https://www.asam.net/standards/`.

[4] *QGIS project. QGIS Server manual*. Letzter Zugriff: 11. September 2024. 2024. URL: `https://docs.qgis.org/3.22/de/docs/server_manual/introduction.html`.

[5] Felix Kistler u. a. „Vehicle endurance testing through automated test driving“. In: *20. Internationales Stuttgarter Symposium*. Hrsg. von Michael Bargende, Hans-Christian Reuss und Andreas Wagner. Wiesbaden: Springer Fachmedien Wiesbaden, 2020, S. 389–401. ISBN: 978-3-658-29943-9.

Validation and Verification of Lidar System: AI-Generated Point Cloud and Object Sensor Model

Basem Abusaif
basem.abusaif@valeo.com

Ahmed Adham
ahmed.adham@valeo.com

Ahmed Elragaby
ahmed.elragaby@valeo.com

Ahmed Yousif
ahmed.yousif@valeo.com

Abstract: Modeling and verification of LiDAR sensor data and algorithms are crucial for robust product development. While previous research has explored LiDAR sensor modeling and phenomenological modeling, there has been limited focus on the fusion of point cloud generation with phenomenological modeling and their verification against real sensor data. This paper addresses this gap by proposing two models: a generative AI model for LiDAR point cloud generation and a phenomenological model for algorithmic tasks like object detection, tracking, and lane detection. Both models are rigorously validated against real sensor data. Our approach facilitates comprehensive end-to-end and module-specific testing, improving validation of perception algorithms. Verification results demonstrate high accuracy, with the phenomenological model achieving 95% similarity to real sensor data and the generative AI model 96%, ensuring robustness and reliability.

1 Introduction

LiDAR (Light Detection and Ranging) technology is crucial in fields such as autonomous vehicles, robotics, and environmental monitoring for creating precise 3D maps, making it vital to ensure the reliability and performance of LiDAR systems across diverse real-world scenarios. However, validating these systems through real-world testing is often expensive and limited in scope. To address these challenges, simulation-based validation offers a promising solution by allowing researchers to test LiDAR systems extensively under simulated real-world conditions, including environmental factors and sensor characteristics, without the high costs associated with field trials. In this paper, we introduce a novel approach to LiDAR validation using advanced simulation models that integrate phenomenological and generative AI to accurately replicate the spatial, intensity, and noise characteristics of LiDAR data. This method provides a scalable, cost-effective way to rigorously test and optimize LiDAR systems, facilitating their broader deployment across various industries.

Contribution: 2024 AutoTest Technical Conference | 16 - 17 October 2024 | Stuttgart

2 Related Work

[YE23, SRB21, PMD19] explored the development of a LiDAR phenomenological sensor model that integrates ground truth data from a simulation platform to generate detection lists, closely mimicking the performance of actual LiDAR systems. This method allows the simulation model to produce detection lists faster than real-time, facilitating quicker validation cycles and better integration with hardware-in-the-loop and software-in-the-loop testing methods. In contrast, [Nic22] investigated generating LiDAR point clouds using two diffusion models conditioned on generated images. [Ach18] focused on training point cloud autoencoders and fitting generative priors, while [LH21] developed two-stage models with a diffusion model for individual points in the second stage and a latent flow model or latent GAN in the first stage. [EAE+19] uses a Unet architecture to generate point cloud echo pulse width and mimic the behavior of the noise profile of radial distance.

Our approach starts with ground truth bounding boxes as input, which serve as an object list for simulating perception outputs. Unlike traditional methods, we do not rely on computationally expensive ray tracing to mimic the behavior of an actual LiDAR sensor. Instead, our method leverages key performance indicators, LiDAR-specific phenomena, and point clouds representing radial distances. The model is trained to generate point cloud intensities that closely resemble real sensor data, learning from both point cloud artifacts (such as radial distance, intensity, and point-level annotations) and ground truth bounding boxes. This gives us a competitive advantage by significantly reducing computational overhead while maintaining high accuracy in sensor simulation.

3 Methodology

3.1 Overview

Validating and verifying LiDAR systems, including components like perception, point cloud generation, and LiDAR-specific phenomena, requires extensive and precise datasets. [VL09] Achieving system maturity demands rigorous testing to ensure reliable performance across diverse conditions. Due to the complexity of LiDAR technology and its integration into autonomous systems, real-world data alone is often inadequate because of the vast range of scenarios required for thorough testing. [KKK11]

To tackle these challenges, simulated data have become crucial for validating LiDAR systems. They offer a controlled environment for testing, allowing early detection of issues, speeding up development, and reducing research and development costs by minimizing extensive real-world testing. [Com22]

LiDAR simulation models are designed to address specific validation needs, with each model tailored to its purpose. For evaluating the accuracy and reliability of LiDAR perception systems, high-fidelity point cloud models are essential for replicating real-world scenarios and thoroughly testing perception algorithms [JHS22]. In hardware-in-the-loop (HiL) setups, real-time performance is critical, and OSI (Open Simulation Interface) point

cloud models provide real-time data that mimics actual LiDAR sensor behavior, ensuring seamless interaction with other vehicle systems [HPK22]. For object-level fusion systems, phenomenological sensor models generate detection lists that simulate LiDAR outputs, facilitating the development and validation of fusion algorithms to ensure accurate object detection and tracking in complex scenarios [SRB21, RR19, PMD19].

Data input interfaces for simulation models vary by framework, but the open simulation interface (OSI) ensures compatibility across platforms. Complying with ISO 23150 standards for simulation scenarios [fS23], OSI offers a standardized framework that improves interoperability among different simulation environments.

Simulating a LiDAR sensor is complex due to the need to replicate its intricate principles. The model must accurately represent laser pulse characteristics, signal processing, receiver properties, and environmental effects. High-fidelity simulations require significant computational resources, typically involving parallel processing with multiple GPUs. [MBW+23]

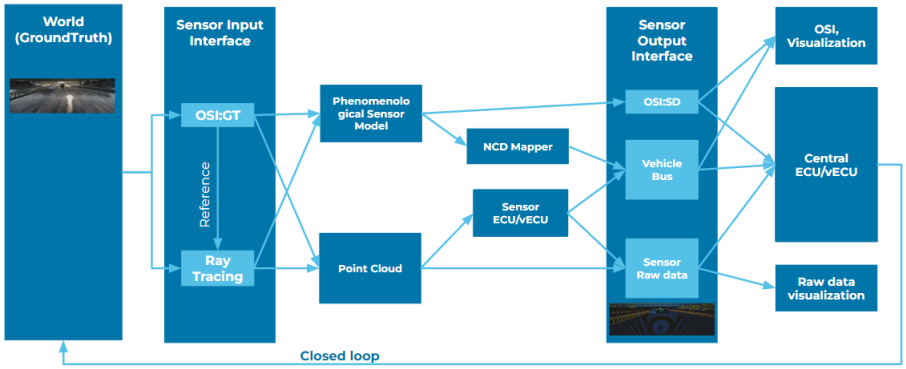


Figure 1: Simulation Pipeline Architecture

3.2 Phenomenological Sensor Model

The purpose of this LiDAR sensor model is to accurately simulate the sensor's response to a given scene. The input to the model includes ground truth data, comprising objects and lanes, along with their positions, velocities, and accelerations. The model processes this data through several stages: managing coordinate systems, filtering the field of view, handling occlusion of objects and lanes, tracking, and, finally, controlling key performance indicators (KPIs). The KPI controller is specifically designed to replicate the behavior of a real LiDAR sensor, ensuring that the model's output closely matches the statistical characteristics of the actual device. Figure 2 illustrates the main architecture of our model, and each of these stages will be discussed in detail [YE23].

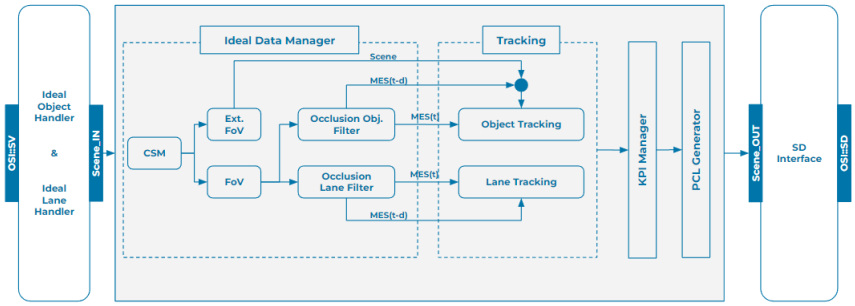


Figure 2: Phenomenological Sensor Model Component Diagram

3.2.1 Coordinate System Manager

As illustrated, the goal is to identify all objects, lanes, and visual cues relative to the LiDAR sensor's position. This step involves taking the input from the scene, which may be recorded relative to the vehicle or a map's reference frame, and performing the necessary transformations to determine their positions with respect to the LiDAR sensor. Figure 3 illustrates the different types of reference frames involved in this process.

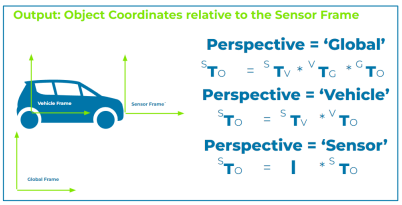


Figure 3: Coordinate System Manager

3.2.2 Field of View Filtration

At this stage, we have transformed all objects and lanes to be relative to the LiDAR sensor's position. With this information, along with the sensor's basic configuration—such as the maximum radial distance and the horizontal and vertical minimum and maximum angles—we can determine visibility. If any part of an object or lane falls within these defined regions, it is considered visible by the LiDAR, as we are still operating under ideal conditions. Figure 4 illustrates how we define the vertical and horizontal limits of the LiDAR sensor.

Figure 4: Field of view filtration

3.2.3 Occlusion

This module is designed to filter out objects and lanes that, although technically within the LiDAR's field of view, remain undetected due to occlusion. Occlusion occurs when one object blocks another, preventing the LiDAR sensor from recognizing the obscured object. To accurately simulate this, we apply a series of nonlinear transformations to the objects and lanes, projecting them into a canonical space. This process helps identify which objects are occluded. Figure 5 illustrates the impact of occlusion on objects and lanes.

Figure 5: Objects and Lanes Occlusion

3.2.4 Tracking

This stage finalizes the ideal sensor model output, making it ready for noise injection. Our goal here is to replicate the tracking behavior found in a real sensor. For instance, if an object is visible but then moves out of the field of view or becomes occluded in subsequent frames, the sensor should still be able to predict its location for several frames. We achieve this by using Kalman filters, which predict the new location and match it with the true and ideal positions defined in the scene, while maintaining the correct object ID.

3.2.5 KPI Manager

The concept behind the KPI manager is to create a high-fidelity model that accurately mimics the behavior of a real sensor. To achieve this, we have collected extensive data, including the positions, orientations, velocities, and accelerations of objects and lanes. Additionally, we considered other factors, such as the type of detected object or lane. By comparing data from the real sensor with ground truth data, we were able to identify the sensor's error or noise in each parameter, such as velocity and position in the X and Y axes.

Using this information, we developed a discrete optimal controller to introduce noise into the ideal sensor model, replicating the deviations observed in the real sensor relative to the ground truth. This ensures that the distribution of each parameter matches that of the actual sensor. Figure 6 illustrates the purpose of the KPI manager, showing that the aim of this module is to generate deviations similar to those observed between the real sensor and the ground truth data.

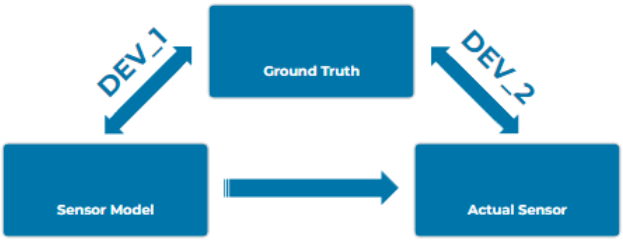


Figure 6: KPI Manager

3.2.6 PCL Generator Model

The PCL model fills the 3D bounding boxes of dynamic and static objects with point cloud data for each object within the LiDAR's field of view. This process involves segmenting the point cloud and assigning relevant points to their respective bounding boxes. These bounding boxes encapsulate the objects, with the points representing the surface geometry and features of each object within the LiDAR's field of view.

3.3 AI-based Intensity Model

AI is changing the automotive industry by making vehicles smarter and safer. One major use is in handling LiDAR data, which helps vehicles see and recognize their surroundings in 3D for tasks like detecting objects and avoiding collisions. By using the UNet architecture, originally for image processing, vehicles can now better understand these 3D images, leading to improved safety and efficiency in self-driving systems.

3.3.1 Polar Grid Map

The Polar Grid Map (PGM) is a 3D tensor-based representation of a full LiDAR scan, that encodes a point cloud of scan data. In this representation, each channel of the PGM is a 2D grid map where each row corresponds to a horizontal layer of the LiDAR data. Using the sensor as the reference point, each scan point is defined by its distance from the sensor, along with its azimuth and inclination angles. The rows and columns of the PGM correspond to the inclination and azimuth angles of the scan points, while the cell values store information about the specific points. In the first channel of the PGM, the cell value indicates the scan point's distance, and in the second channel, it means the scan point's class. This representation can be expanded by adding additional channels to store more information,although In our study we use 3 channels represented in ray tracing radial distance, object class ,and object material. An example of PGM representation in Figure 7.

Figure 7: Polar Grid Map

3.3.2 Intensity Model Architecture

In Figure 8 shows intensity model architecture primarily consists of multiple stages, starting with a simulation environment input that generates data for further processing. This input is then mapped into a Polar Grid Map (PGM), which organizes the LiDAR data into a structured format. Following this, a preprocessing step refines the data, preparing it for the core of the architecture a UNet-based deep neural network (DNN). The UNet DNN processes the data to predict the desired output with enhanced accuracy. After the DNN, a post-processing step is applied to further refine the results, leading to the final expected output. This structured approach ensures that the architecture efficiently handles the input data, leading to accurate and reliable predictions. Additionally, our integration with the Phenomenological sensor model will be facilitated through the PCL generator model that map the object level data into ray tracing point cloud, ensuring seamless data flow and enhanced object detection capabilities.

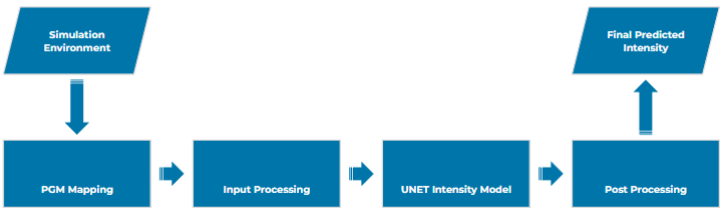


Figure 8: Intensity Model Architecture

3.3.3 Intensity DNN

Our intensity prediction model is built around a UNet-based deep neural network (DNN), which is well-suited for processing the structured data provided by the Polar Grid Map (PGM). The UNet architecture is ideal for this task due to its unique structure, which effectively captures both detailed and contextual information through its encoder-decoder design. The architecture consists of two main components: the contracting path (encoder) and the expansive path (decoder).

The contracting path captures the contextual features of the input data by downsampling it through a series of convolutional layers, each followed by max-pooling. This process extracts increasingly abstract features while reducing the spatial dimensions of the data.

The expansive path then works to restore the original resolution by upsampling the feature maps using transposed convolutions. To preserve spatial information and improve accuracy, skip connections are utilized between corresponding layers of the encoder and decoder, ensuring that the network retains both high-level context and fine-grained details.

In this model, the UNet processes the PGM input to predict the intensity values of each scan point. Accurate intensity prediction is crucial for tasks such as environmental perception and object detection, which rely on precise interpretation of LiDAR data.

The model is trained using a modified Mean Squared Error (MSE) loss function, enhanced with a regularization term to prevent overfitting. The MSE loss function is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} w_j^2$$

In this formula, y_i represents the true intensity values, $\hat{y}_i$ represents the predicted intensity values, n is the number of scan points, w_j denotes the model weights, and λ is a regularization parameter that controls the extent of weight penalization. The addition of the regularization term helps to reduce the model's complexity, encouraging it to generalize better to unseen data by discouraging overly complex models that may fit the training data too closely.

By minimizing this loss function, the model not only learns to make accurate predictions but also maintains robustness against overfitting, ensuring reliable performance in real-world scenarios. This combination of UNet architecture and a regularized MSE loss function allows the model to deliver precise intensity predictions critical for advanced automotive applications. The effectiveness of the model is demonstrated in Figures 9 and 10, where the predicted intensities closely align with the ground truth, showcasing the model's accuracy. The comparison between the two figures highlights the model's ability to replicate the true intensity values with minimal error.

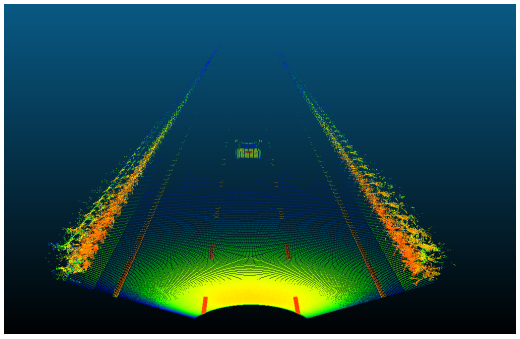


Figure 9: Ground Truth Intensities

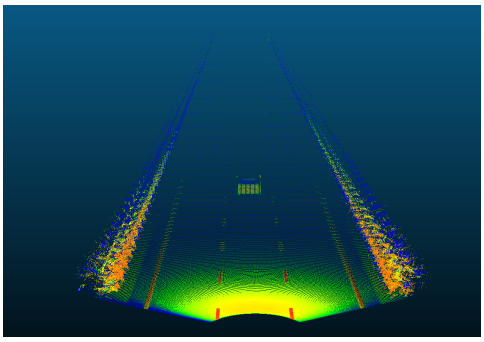


Figure 10: Predicted Intensities

4 Sensor Model Qualification

To ensure the developed model accurately simulates the real sensor, a sensor model qualification process is implemented. This involves comparing real recorded data from the

vehicle's system with simulated data. [LS]

The qualification process, shown in Figure 11, involves recording real-world data and creating a corresponding digital twin scenario in the simulation environment. These scenarios are rigorously compared using key performance indicators (KPIs) and bug-to-bug testing in Hardware-in-the-Loop (HiL) or Software-in-the-Loop (SiL) setups. This method ensures accurate replication of the sensor's functionality, including any defects or bugs.

Point cloud intensity qualification evaluates the quality and accuracy of intensity values in a 3D point cloud, representing the reflectivity of surfaces captured by LiDAR sensors. This process is crucial for ensuring that the intensity data accurately reflects real-world conditions, aiding in tasks like object detection and environmental mapping. By validating the intensity values by calculate the deviation between the Point cloud intensity sensor model and the real sensor , we can improve the reliability of point cloud data for downstream applications.

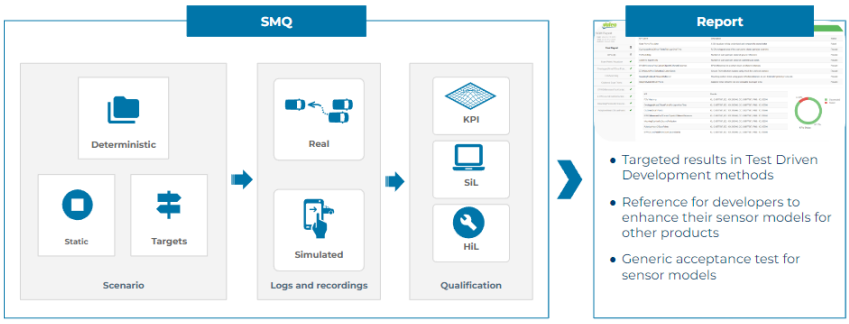


Figure 11: Sensor Model Qualification Components

5 Conclusion

In conclusion, this study highlights the importance of combining AI-generated point clouds with phenomenological sensor models to accurately validate and verify LiDAR systems. These models simulate real-world sensor performance, ensuring reliability. The AI model effectively generates point clouds that closely resemble those from actual LiDAR sensors, while the phenomenological model accurately mimics object detection and tracking. Together, these models allow for comprehensive testing of the LiDAR system and its components, enabling early detection of issues during development. These methods provide a cost-effective alternative to extensive real-world trials by offering detailed simulations across various conditions, ensuring the system's readiness for diverse scenarios. As technology advances, these approaches will be key in developing reliable autonomous systems across different industries.

References

[Ach18] P. et al. Achlioptas. Learning Representations for 3D Point Clouds, 2018.

[Com22] McKinsey & Company. Digital Twins: The Art of the Possible. https://shorturl.at/wAxZV, 2022. Accessed: 2024-08-20.

[EAE+19] K. Elmadawi, M. Abdelrazek, M. Elsobky, H. M. Eraqi, and M. Zahran. End-to-End Sensor Modeling for LiDAR Point Cloud. In *IEEE Int. Transportation Systems Conf.*, pages 1619–1624, Auckland, New Zealand, 2019.

[fS23] International Organization for Standardization. ISO 23239-1:2023 - Road Vehicles. https://www.iso.org/standard/83293.html, 2023. Accessed: August 21, 2024.

[HPK22] A. Haider, M. Pigniczki, and M. Köhler. High-Fidelity Automotive LiDAR Sensor Model. *Sensors*, 22(7556), 2022.

[JHS22] W. Jansen, N. Huebel, and J. Steckel. Physical LiDAR Simulation in Real-Time Engine. In *IEEE Sensors 2022*, 2022.

[KKK11] B. Khaleghi, A. Khamis, and F. Karray. Multisensor data fusion: A review. *Informat. Fusion*, 14(1), 2011.

[LH21] S. Luo and W. Hu. Diffusion Models for 3D Point Cloud Generation, 2021.

[LS] S. Ljungberg and F. Schalling. Benchmarking of LiDAR Sensor for Automotive Perception. Unpublished.

[MBW+23] Sivabalan Manivasagam, Ioan Andrei Bârsan, Jingkang Wang, Ze Yang, and Raquel Urtasun. Towards Zero Domain Gap: A Comprehensive Study of Realistic LiDAR Simulation for Autonomy Testing. In *ICCV*, 2023.

[Nic22] A. et al. Nichol. Point-E: Generating 3D Point Clouds, 2022.

[PMD19] Thomas Ponn, Fabian Muller, and Frank Diermeyer. Systematic Analysis of the Sensor Coverage of Automated Vehicles Using Phenomenological Sensor Models. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, June 2019.

[RR19] A. Rövid and V. Remeli. Raw Sensor Fusion in 3D Object Detection. In *IEEE 17th World Symposium on Applied Machine Intelligence and Informatics*, January 2019.

[SRB21] J. Schmitt, C. Robel, and B. Bäker. Measurement-Based LiDAR Sensor Model. In *21st Int. Stuttgarter Symposium*, pages 424–435, 2021.

[VL09] J. Van Laere. Challenges for IF Performance Evaluation. In *12th Int. Conf. on Information Fusion*, July 2009.

[YE23] A. Yousif and M. Elsobky. LiDAR Phenomenological Sensor Model. In *Mobility 4.0*, number 2023-01-1902, 2023.

Scenario Generation for Testing Automated Driving Systems

Lukas Birkemeyer, Ina Schaefer

{lukas.birkemeyer, ina.schaefer}@kit.edu

Abstract: The SOTIF-standard (ISO 21448) establishes scenario-based testing as state-of-the-art for verifying and validating Advanced Driver Assistance Systems (ADAS) and Automated Driving Systems (ADS). However, the SOTIF standard lacks details for selecting scenarios used as test cases. Consequently, missing details of SOTIF hinder its practical application. In this paper, we analyze existing scenario generation techniques and discuss whether they generate SOTIF-compliant scenario suites. Subsequently, we leverage variability modeling techniques to address two essential challenges that remain open: How to model an overall scenario space and how to cover it practically? We elaborate on sampling strategies and coverage criteria that are relevant for generating SOTIF-compliant scenario suites. Finally, we assess and compare generated scenario suites using mutation testing indicating the ability of scenario suites to detect potential failures.

1 Introduction

Highly automated Advanced Driver Assistance Systems (ADASs) are widely spread in modern vehicles, aiming to increase comfort, safety, and sustainability in road traffic. Increasing the autonomy of ADAS up to Automated Driving Systems (ADSs) potentially improves both aspects even further. However, increasing autonomy simultaneously increases the testing requirements. Testing strategies that are sufficient for testing ADAS become insufficient for verifying and validating ADS. For example, Wachenfeld and Winner [WW16] state that distance-based testing is practically not suitable to verify and validate highly automated ADAS/ADS due to the extremely large number of possible test kilometers required. Instead, the current Safety of the Intended Functionality (SOTIF) standard (ISO 21448) [iso22] establishes *scenario-based testing* as state-of-the-art to test highly automated ADAS/ADS. Scenario-based testing includes the environment of the vehicle in the testing process to evaluate the interaction of the System Under Test (SUT) with the environment. A *scenario* is a test case in scenario-based testing describing the environment of a vehicle, including the road infrastructure, static/dynamic actors, and weather conditions. In practice, to perform scenario-based testing, a *scenario suite*, a set of scenarios, is required. The SOTIF standard [iso22] has two requirements for those scenarios suites: (Req-1) The scenario suite needs to *sufficiently* cover all possible scenarios that potentially occur in the operational design domain of the ADAS/ADS. (Req-2) Scenarios need to minimize the set of hazardous scenarios. A scenario is *hazardous* if it contains triggering conditions turning functional insufficiencies of the SUT into hazardous behavior potentially causing harm [iso22].

Although the SOTIF standard describes requirements for scenario suites, it lacks details on how to generate them. Modeling and testing all possible real-world scenarios is practically not feasible due to the extremely large number of possible scenarios [iso22]. Thus, regarding (Req-1), it is unclear how to define a coverage criterion to *sufficiently* cover the space of possible scenarios. Regarding (Req-2), it is unclear how to determine whether a scenario contributes to minimizing the set of hazardous scenarios. Moreover, the SOTIF standard lacks details for assessing scenarios. Thus, for test engineers, it is not possible to evaluate whether a scenario suite is *good* for testing ADAS/ADS. Without defining those missing details, concrete methodologies to generate scenarios suites, or metrics to assess generated scenarios, it is not possible to apply the SOTIF standard practically.

In this paper, we pursue the overarching research goal: Generation of SOTIF-compliant scenario suites to practically verify and validate ADAS/ADS. We discuss existing scenario generation methods wrt. the SOTIF standard and identify current research gaps towards SOTIF-compliant scenario generation. We introduce a combinatorial scenario generation concept capable of generating scenario suites according to a coverage criterion (Req-1). Subsequently, we contribute two improvements towards closing existing research gaps. The improvements extend combinatorial scenario generation for generating scenario suites that additionally comply with (Req-2). We assess generated scenario suites by their ability to detect potential failures in an SUT and evaluate the proposed improvements for combinatorial scenario generation using a large-scale automotive case study.

2 State of the Art: Scenario Generation

To determine whether existing scenario generation approaches can already generate SOTIF-compliant scenario suites, we performed an Systematic Literature Review (SLR) [BKS23] according to the guidelines of Kitchenham [Kit04]. We defined a search string that we applied to search engines of scientific databases and defined objective inclusion/exclusion to collect relevant literature proposing scenario generation approaches for testing ADAS/ADS. In total, the following findings are based on 150 selected articles. We observe that existing scenario generation approaches are systematic, i.e., not random. Prime examples for systematic scenario generation approaches that can automatically generate large-scale scenario suites are data-driven, optimization-based, or combinatorial approaches.[1]

To analyze existing scenario generation approaches for SOTIF-compliance, we determined which entities, potentially occurring in a scenario, are covered (Req-1). Moreover, we determined whether existing scenario generation approaches consider knowledge of the SUT. The SUT-knowledge is relevant for generating scenario suites that comply with (Req-2); holistically considering the scenario space independently of the SUT, in turn, is relevant for generating scenario suites that comply with (Req-1). Additionally, we determined whether the behavior of the SUT for scenario generation has to be *known* or *unknown* and whether a generated scenario is *hazardous* or *not hazardous*, as the SOTIF standard [iso22] requires to minimize the set of (*unknown*) *hazardous* scenarios (Req-2).

Although existing scenario generation approaches can in principle address all *types* of

[1] Techniques to generate scenarios are ordered according to their share in current literature.

scenario entities,[2] they do not cover each possible entity potentially occuring in the real world. Optimization-based approaches contribute to minimizing the set of *hazardous* scenarios (Req-2) since they consider the knowledge of the SUT to exploit the space of possible scenarios. These approaches assume that critical scenarios will likely trigger hazardous behavior of the SUT. Combinatorial scenario generation approaches, in turn, are beneficial to generate scenario suites covering the scenario space (Req-1) since they holistically explore the scenario space independently of the SUT. We conclude that none of the existing scenario generation approaches generates SOTIF-compliant scenario suites. Existing scenario generation approaches either address (Req-1) by exploring the possible scenario space or (Req-2) by exploiting the possible scenario space. Based on our findings, we identify two research gaps that remain open: (1) combining existing scenario generation techniques to address both requirements of the SOTIF standard, and (2) balancing exploration vs. exploitation.

3 Combinatorial Scenario Generation

In this paper, we combine scenario generation approaches addressing both SOTIF-requirements to close existing research gaps hindering the practical application of the SOTIF standard. As a basis, we describe a combinatorial scenario generation approach complying with (Req-1). Moreover, we describe a method for assessing scenario suites in accordance with SOTIF.

Scenario Generation: In [BPV+22], we propose a combinatorial scenario generation approach leveraging variability modeling techniques for generating scenario suites. Variability modeling techniques use *feature models* to capture the space of possible configurations [ABKS13]. In simulation tools used to test ADAS/ADS, a *feature* of the feature model describes an atomic scenario entity. Thus, the feature model represents the space of all possible scenarios that might be simulated using the simulation tool. Selecting a set of features from the feature model results in a scenario in the simulation tool. We apply established sampling strategies (namely uniform random sampling and feature interaction coverage sampling) to derive suites of scenarios from the scenario space feature model. Uniform random sampling selects scenarios randomly, while feature interaction coverage sampling covers all interactions of features systematically, i.e., the interaction of a specific number of atomic scenario entities.

Scenario Assessment: In [BPV+22], we propose mutation testing to assess generated scenario suites. Mutation testing is an established method in software engineering to assess test cases according to their ability to detect potential failures in an SUT [JH10]. In mutation testing, artificial errors are seeded in the SUT to represent faults potentially occurring in the real world. A *mutation score* indicates the share of killed *mutants* (i.e., number of detected faulty SUT). To decide whether a mutant is killed or alive, mutation testing refers to a *kill criterion*. In accordance with the SOTIF standard, we develop a kill criterion explicitly focusing on safety-critical incidents [BPV+22].

Our results in [BPV+22] demonstrate that feature models are suitable for representing scenario spaces. Moreover, feature interaction coverage sampling can reduce the number

[2] We define *types* of scenario entities, according to the classification proposed by Bagschik et al. [BMM18]

of test scenarios up to 96% while maintaining their fault detection capabilities in terms of a high mutation score. We conclude that variability modeling techniques are beneficial for generating scenario suites that sufficiently cover a scenario space (Req-1).

4 SOTIF-compliant Scenario Generation

We propose two concepts to improve the combinatorial scenario generation approach towards SOTIF-compliant scenario generation by additionally addressing (Req-2).

Selective Sampling: The first improvement addresses the coverage criterion. Since the SOTIF standard lacks details of a concrete coverage criterion, in [BPV+22], we applied feature interaction coverage sampling using a combinatorial interaction coverage criterion. However, Kaltenecker et al. [KGS+19] state that feature interaction coverage sampling does not consider the number of features included in a configuration. In the scenario generation context, the number of features correlates to the complexity of a scenario. A crowded scenario is more complex than a scenario without road traffic. Thus, a scenarios' complexity is relevant for defining *sufficient* coverage.

To realize a complexity-aware coverage criterion, in [BPRS24], we adapt and modify the distance-based sampling approach provided by Kaltenecker et al. [KGS+19]. We introduce *selective sampling* considering the complexity distribution of an overall scenarios space. The intention is to generate a scenario suite with the same complexity distribution as the overall scenario space. Additionally, similar to distance-based sampling [KGS+19], we combine selective sampling with feature coverage criterion by prioritizing scenarios containing atomic scenario entities (i.e., features) that are not covered in the current scenario suite [BPRS24].

Our results in [BPRS24] indicate that selective sampling maintains the testing quality of generated scenario suites in terms of mutation scores compared to feature interaction coverage and distance-based sampling. Additionally, selective sampling generates scenario suites that cover the complexity distribution of an overall scenario space.

Parameter Sampling: The second improvement expands combinatorial scenario generation regarding (Req-2) by combining combinatorial scenario generation with parameter sampling [BFGS23]. We observe that scenarios contain discrete and continuous parameters. In combinatorial scenario generation [BPV+22], we discretize continuous parameters using equivalence classes based on expert knowledge assuming all parameter values of an equivalence class trigger equivalent behavior of the SUT. However, in [BFGS23], we demonstrate that equivalence classes are not sufficient.

Instead, in [BFGS23], we propose splitting the scenario sampling process into two consecutive steps: First, we select *discrete* scenario entities using feature interaction coverage sampling, and second, we select concrete parameter values of *continuous* parameters using parameter sampling. To realize parameter sampling, different sampling techniques such as random, equidistant, or optimization-based sampling can be applied. A feedback mechanism can improve parameter sampling by producing critical scenarios potentially minimizing the set of (unknown) hazardous scenarios. Depending on the definition of the parameter ranges and sampling strategy, test engineers can balance exploration vs. exploita-

tion of the scenario space. For example, subdividing the full possible parameter range into small subparameter ranges enables a strong coverage criterion.

Our results in [BFGS23] demonstrate that random parameter sampling outperforms expert-defined equivalence classes. Concerning the requirements of the SOTIF standard, parameter sampling enables to optimize continuous scenario parameters for minimizing the set of (unknown) hazardous scenarios (Req-2) while sufficiently covering a scenario space wrt. discrete scenario entities (Req-1). Thus, parameter sampling extends the combinatorial scenario generation approach towards SOTIF-compliant scenario generation. The feedback mechanism and the definition of subparameter ranges enable testers to balance exploration (Req-1) vs. exploitation (Req-2) while generating scenario suites.

5 Evaluation

In this section, we evaluate combinatorial scenario generation and its improvements [Bir25]. The fundamental experiment setup is as follows: We generate scenario suites with different strategies and assess generated scenario suites using the mutation score.

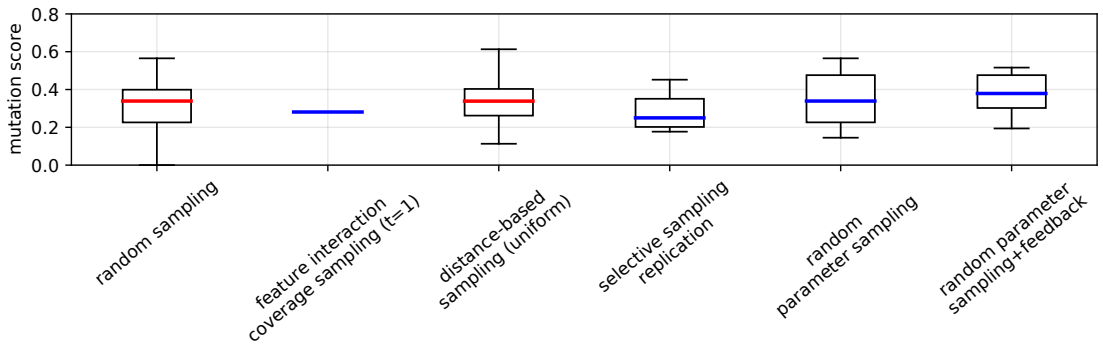
Subject System: In this evaluation, we use an Adaptive Cruise Control (ACC) case study. An ACC is an ADAS controlling the longitudinal velocity of a vehicle considering the traffic ahead. We implement a large-scale scenario space feature model covering highway scenarios relevant to test ACC functionality. To discretize continuous scenario parameters such as velocity, rain rate, or temperature, we define three equivalent classes each. The resulting feature model consists of 197 features representing a space of $152 \cdot 10^{12}$ concrete scenarios.[3]

Sampling Strategies: We generate scenario suites with feature interaction coverage sampling (namely the YASA algorithm [KTS+20]) using feature-wise (t=1) and pair-wise (t=2) coverage. We also generate scenario suites using selective sampling replicating the complexity distribution of the overall scenario space. To apply parameter sampling, we consider the continuous range of possible scenario parameter values. We generate scenario suites with purely random parameter sampling and in combination with the feedback mechanism. The feedback mechanism resamples scenario parameters until a scenario contains road traffic triggering ACC functionality. As baselines, we generate scenario suites using uniform random sampling and distance-based sampling [KGS+19]. Since most scenario sampling strategies are non-deterministic, we average over ten versions of generated scenario suites each. Inspired by the two feature interaction coverage criteria feature-wise and pair-wise, we generate scenario suites of 7 and 80 scenarios with all considered approaches.

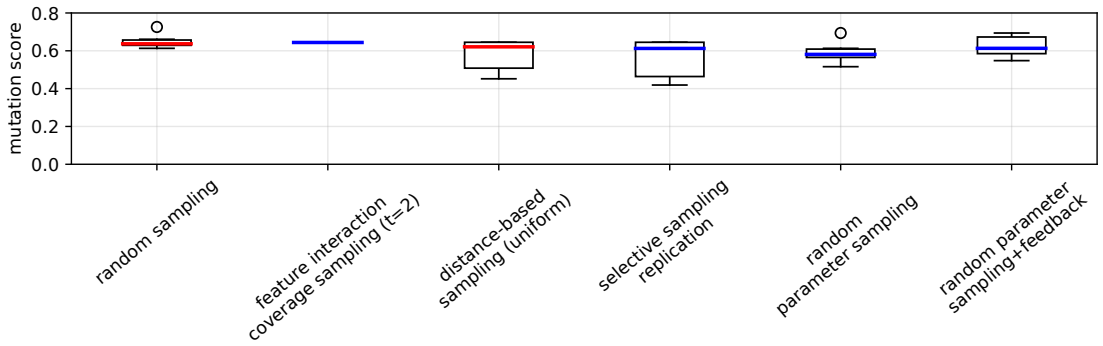
Scenario Assessment: We generate a set of 62 unique mutants of the ACC case study that implement exactly one artificial error using the SIMULTATE mutation framework [PRWN16]. We use the same set of mutants to assess each scenario suite. As a domain-specific kill criterion, we focus on safety-critical incidents [BPV+22]

Results and Findings: In Figure 1, we present the mutation scores we determine for scenario suites generated using several sampling strategies: (a) for the smaller scenario

[3] Counted with SharpSAT: `https://github.com/marcthurley/sharpSAT`



(a) Small scenario suites containing 7 scenarios



(b) Large scenario suites containing 80 scenarios

Figure 1: Mutation scores for scenario suites generated using different sampling strategies. The higher the mutation score, the better is the scenario suite in detecting potential failures.

suites of 7 scenarios; (b) for the larger scenario suites of 80 scenarios. The higher the median mutation score, the *better* is the scenario suite in detecting potential failures. A mutation score of 1 means that the scenario suite is capable of killing all mutants, while a mutation score of 0 means none of the potential failures is detected. A mutation score of 0.5 means that the scenario suite kills 50% of all mutants, which is as good as chance. For *feature interaction coverage sampling*, we observe that scenario suites generated using pair-wise coverage lead to *better* scenario suites than feature-wise coverage. Indeed, random sampling leads to similar median mutation scores, but the mutation scores scatter more, which impacts the reliability of a single test run. For *complexity-aware sampling*, the results reveal that distance-based sampling and selective sampling lead to similar median mutation scores compared to feature interaction coverage sampling. Standard distance-based sampling slightly improves the median mutation score for the small scenario suites. These results indicate that considering the complexity of scenarios is relevant for generating scenario suites. The additional benefit of selective sampling is an additional complexity-aware coverage criterion, which is in accordance with (Req-1). For *parameter sampling*, our results indicate that random parameter sampling improves scenario suites for small scenario suites compared to feature interaction coverage sampling and selective sampling. For large scenario suites, the median mutation scores are similar. Adding a feedback mechanism to parameter sampling further improves the generated scenario suite.

6 Conclusion

In this paper, we consider the research challenge of generating SOTIF-compliant scenario suites. To close the research gaps highlighted in [BKS23], we combine and evaluate existing concepts for generating scenario suites regarding the SOTIF standard. The combined scenario generation concept is based on a combinatorial approach proposed in [BPV+22]. Two improvements extend the combinatorial scenario generation approach towards SOTIF-compliance: 1) We improve the coverage criterion (Req-1) for sufficiently covering a scenario space using selective sampling [BPRS24] taking into account scenario complexity. 2) We combine combinatorial scenario sampling with parameter sampling [BFGS23] for generating (unknown) hazardous scenarios (Req-2) and enabling balancing exploration (Req-1) vs. exploitation (Req-2). Using a mutation score as a metric, we evaluated and compared the proposed concepts based on a large-scale ACC case study [Bir25]. Our results indicate that higher feature interaction coverages lead to *better* scenario suites. Moreover, a complexity-aware coverage criterion generates scenario suites leading to similar median mutation scores, but realizes an additional coverage criterion that is relevant wrt. (Req-1) of the SOTIF standard. Parameter sampling that uses a feedback mechanism improves the median mutation scores while generating SOTIF compliant scenario suites by addressing both requirements (Req-1) and (Req-2).

We conclude: Combining existing scenario generation approaches leads to scenario suites compliant to both requirements of SOTIF. Thus, the proposed improvements close the research gap addressed in [BKS23] towards SOTIF compliant scenario generation. In the future, we plan to determine the impact of balancing (Req-1) vs. (Req-2), i.e., focusing on covering the scenario space vs. focusing on scenarios that minimize the set of hazardous scenarios. Moreover, we aim to transfer the proposed concepts to further ADAS/ADS case studies to determine their generality. Lastly, we plan to identify the space of scenarios relevant for specific (types of) SUT to further minimize the number of test scenarios required to verify and validate ADAS/ADS.

7 Acknoledgment

This paper project is partially funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – 501798263.

References

[ABKS13] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. Feature-oriented software product lines. 2013.

[BFGS23] Lukas Birkemeyer, Julian Fuchs, Alessio Gambi, and Ina Schaefer. SOTIF-Compliant Scenario Generation Using Semi-Concrete Scenarios and Parameter Sampling. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023.

[Bir25] Lukas Birkemeyer. *Scenario Generation for Testing Advanced Driver Assistance Systems and Automated Driving Systems*. PhD thesis, Technische Universität Braunschweig, 2025. (forthcoming).

[BKS23] Lukas Birkemeyer, Christian King, and Ina Schaefer. Is Scenario Generation Ready for SOTIF? A Systematic Literature Review. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023.

[BMM18] Gerrit Bagschik, Till Menzel, and Markus Maurer. Ontology based scene creation for the development of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1813–1820. IEEE, 2018.

[BPRS24] Lukas Birkemeyer, Tobias Pett, Tobias Runge, and Ina Schaefer. Selective Sampling: A Complexity-Aware Sampling Strategy for Combinatorial Scenario-based Testing. 2024. (under review).

[BPV+22] Lukas Birkemeyer, Tobias Pett, Andreas Vogelsang, Christoph Seidl, and Ina Schaefer. Feature-Interaction Sampling for Scenario-based Testing of Advanced Driver Assistance Systems. In *Proceedings of the 16th International Working Conference on Variability Modelling of Software-Intensive Systems*, pages 1–10, 2022.

[iso22] ISO 21448:2022 Road vehicles — Safety of the intended functionality, 2022.

[JH10] Yue Jia and Mark Harman. An analysis and survey of the development of mutation testing. *IEEE transactions on software engineering*, 37(5):649–678, 2010.

[KGS+19] Christian Kaltenecker, Alexander Grebhahn, Norbert Siegmund, Jianmei Guo, and Sven Apel. Distance-based sampling of software configuration spaces. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1084–1094. IEEE, 2019.

[Kit04] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

[KTS+20] Sebastian Krieter, Thomas Thüm, Sandro Schulze, Gunter Saake, and Thomas Leich. YASA: yet another sampling algorithm. In *Proceedings of the 14th International Working Conference on Variability Modelling of Software-Intensive Systems*, pages 1–10, 2020.

[PRWN16] Ingo Pill, Ivan Rubil, Franz Wotawa, and Mihai Nica. Simultate: A toolset for fault injection and mutation testing of simulink models. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 168–173. IEEE, 2016.

[WW16] Walther Wachenfeld and Hermann Winner. The release of autonomous vehicles. *Autonomous Driving: Technical, Legal and Social Aspects*, pages 425–449, 2016.

Advances in multi-abstraction distributed vECU co-simulation technology for automotive software testing

Markus Wedler, Alexandru Fiodorov, Marcel Heistermann, Maximilian Fricke, Shubham Paul

Synopsys Inc.
Ritter Str. 23
52072 Aachen
wedler@synopsys.com

Abstract: This paper introduces the Synopsys electronics-digital-twin-fabric (eDT-fabric), a new backplane technology for distributed co-simulation for virtual electronic control-units (vECUs). This eDT-fabric technology addresses the need of OEMs to integrate vECUs and plant-models from various vendors into full-car simulations. These electronics digital twins are modeled at various abstraction levels with very different requirements regarding their execution platform. The eDT-fabric described in this paper therefore provides true multi-host, multi-host-platform, multi-host-OS and multi-abstraction co-simulation capabilities. Its advanced synchronization and data exchange mechanisms enable maximum performance while keeping the overall simulation deterministic.

1 Introduction

OEMs aim for virtualization of the entire car and its environment for full system validation of their embedded system software stacks prior to deployment on hardware-in-the loop testing platforms and subsequent field testing. Depending on the individual validation targets, Virtual ECUs modelled at different levels of abstraction are used in these co-simulation environments. It is thus desired to seamlessly switch between the abstraction levels and this needs to be supported by a distributed co-simulation platform.

The introduction of Adaptive AUTOSAR as a POSIX compliant embedded OS further increases the heterogeneity of the simulation models. Availability of certain virtual models and tools in the automotive software testing eco-system may be limited to certain OS-platforms. Specific hardware requirements may also be imposed by scenario simulators that often call for advanced GPUs for rendering of environment images.

In this context we introduce our next generation parallel co-simulation backplane technology - the so-called Synopsys eDT-fabric for distributed data-exchange and synchronization between virtual ECUs and environment simulations. This is the crucial component of our platform for full system test & development of embedded vehicle software. It offers true multi-abstraction, -host, -host-architecture, -host-OS support. Differences in simulator performance are embraced and additional communication overhead is minimized. Our synchronization methods ensure deterministic simulation and maximize parallel execution.

2 Motivation

The automotive industry is facing a lot of new challenges. The world-wide shortage of chips has compressed the traditional supply chain of OEMs, Tier 1s and Tier 2s etc. New entrants into the market come up with radical new approaches designing their own SoCs. In this manner they have the entire value chain from hardware to software in their own hand. This enables system-wide optimizations and innovations from silicon to software.

At the same time, we see a major overhaul of vehicle E/E architectures. The classical scheme of one ECU per function does not scale sufficiently to support new HAD/ADAS features. The electrification of the powertrain drastically changes the software needed in this domain as well. Moreover, car owners require infotainment systems that deliver similar or even better UI experiences as they are used to in their consumer electronics. All this leads to new zonal architectures that leverage powerful central compute clusters to do the heavy lifting in computing for several zonal controllers that coordinate specific regions in the car.

In total we are talking about millions of lines of code being executed. In the end the OEM is responsible for all parts of the software working together flawlessly. The fact that the code stems from various vendors today and often is only delivered in binary causes many integration issues. We envision that over time completely new software ecosystems will evolve. Standardized interfaces, e.g., AUTOSAR Classic for real time applications and AUTOSAR Adaptive for the POSIX based applications, are promising first step in that direction.

Software updates over-the-air will become widely available to roll out new Software features and enable fixing issues in the field that slipped despite all efforts.

However, this increases the attack surface for malicious access through the over-the-air update infrastructure. Combined with the vision for self-driving cars that are always connected through 5G networks. Thus, a strong focus on safety and security aspects of these interfaces is mandatory.

New entrants into the automotive markets challenge the traditional segregated development processes and thereby accelerate their time to market.

3 Electronics Digital Twins and their abstraction levels

Virtualization is a key technology to massively scale the software testing efficiency. The traditional physical testing strategy cannot keep pace with the above-mentioned trends and has become the bottleneck in development. The trend in the industry is clearly to transition towards virtualization, i.e., simulation-based testing [1].

This has numerous benefits. It enables early software bring-up even before physical prototypes of the tested subsystems become available. State-of-the-art virtual ECU technology provides deterministic simulations and a very high degree of debug visibility into the systems under test. This shortens debugging cycles and boosts the overall testing productivity. Electronics digital twins (eDTs) support agile development methodologies based on continuous integration/continuous delivery of the embedded software under development. The software nature of these virtual environments also enables a tighter collaboration through the supply chain of the OEMs [2]. Exchanging software in binary or source can be done frequently in increments, whereas Hardware needs to be physically manufactured and shipped with the attached costs. Thus, a high number of iterations that characterizes true agile development processes is no longer a showstopper for the collaboration.

Standardized APIs like the Functional Mockup Interface (FMI) for co-simulation and model exchange [2] for host-compiled vECUs or the SystemC/TLM standard [3] for modeling virtual prototypes of the underlying hardware that can run unmodified binary code, provide another important foundation for such collaborative development approaches.

For a more precise categorization of the various abstraction levels at which virtual ECUs can be modeled, we refer to the whitepaper [4]. In this paper, the authors introduce the abstraction levels visualized in Figure 1.

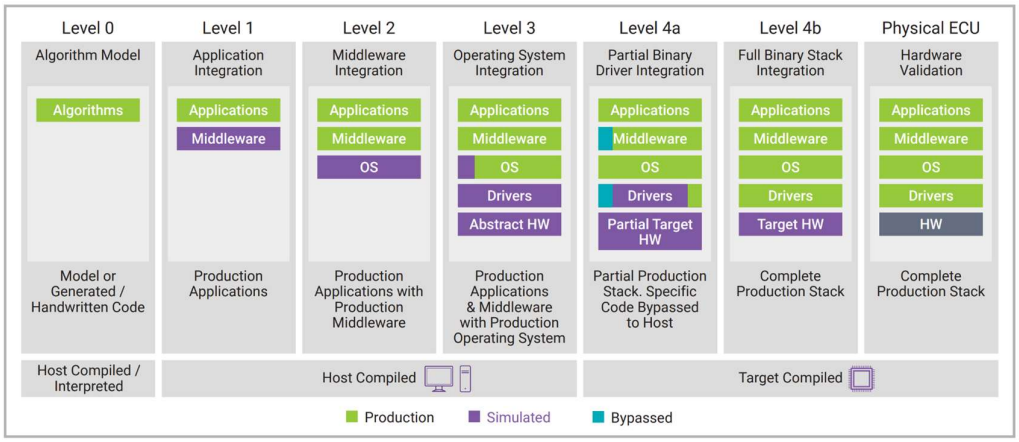


Figure 1: Virtual SW/HW abstractions for broadest use cases.

The categorization reaches from algorithmic models, created by model-based design tools like Matlab Simulink and tested in so called model-in-the-loop setups on the left side of the figure, to a complete physical prototype of an ECU running on a hardware prototyping system. The spectrum between these two cases is filled with vECU models that take into account ever more details of the underlying middleware, OS, hardware drivers and target hardware behavior. The figure shows which portions of the software stacks are taken from the real production code and which portions are simulated at the respective levels.

The Figure enhances the categorization of [5] that was mainly focused on AUTOSAR Classic use-cases to also reflect POSIX based developments around AUTOSAR Adaptive. For this purpose, the level 4 dealing with target compiled software is split into the levels 4a and 4b. At level 4a VirtIO driver substitution and host-extension technologies may be involved to simulate or bypass parts of the native driver behavior and thus reduce the scope of the necessary hardware models. By contrast, vECUs at level 4b can run the full binary production code like the target hardware.

4 Integration of vECUs

One important benefit of electronics digital twins is that we can leverage vECUs and their early availability for early system integration. Instead of waiting for a big bang integration when all ECUs are available, partial systems needed for specific driving scenarios can be setup up and tested. As an example, let us consider an adaptive cruise control and navigation tracking scenario. The ECUs involved in this scenario are integrated into the setup illustrated in Figure 2. It consists of 4 virtual ECUs for the navigation system, the central compute cluster, a zonal domain controller, and a control ECU, modeled at different abstraction levels. In addition, a third-party environment simulator and a playback node representing the Human Machine Interface (HMI) are added to the setup. Visualization, debug, and tracing capabilities are enabled for validation and root cause analysis in case of unexpected system behavior.

The need for integration of simulators and simulation models from various tool vendors with their specific hardware and platform requirements further increases the heterogeneity of the co-simulation. For example, the environment simulator may require a dedicated graphics card to render images. Other models may need an NPU, e.g., in the sensor data processing. Some tools may only be available on either Windows or Linux. Hypervisor based simulations may take benefit of specific host-architectures like aarch64.

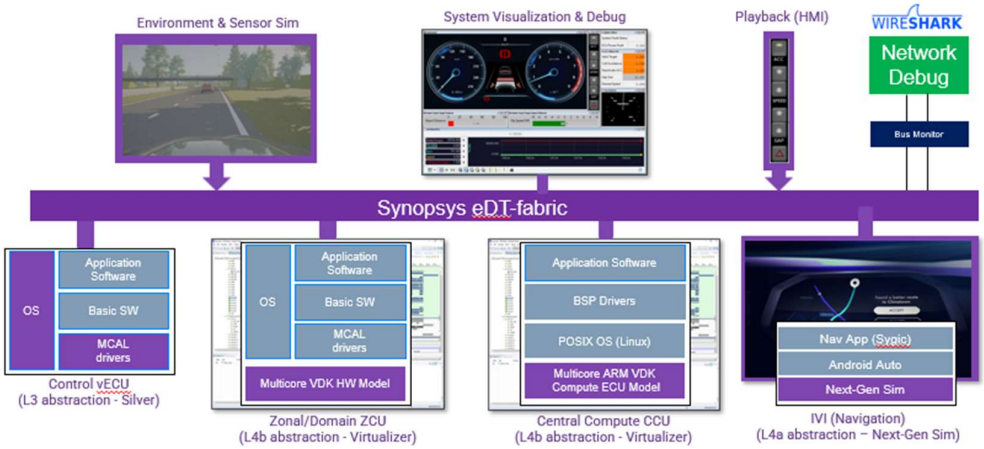


Figure 2: Early system integration for complex function validation.

5 Implications for the eDT-fabric and its features

In this section we will focus on our technology for the challenges described in the previous two sections. We introduce our new co-simulation backplane technology called electronics-digital-twin-fabric (eDT-fabric). This eDT-fabric has the following characteristics:

- **Multi-abstraction**: It allows for seamless integration of vECUs at level 1 to 4. It facilitates idealized Signal based communication as well as packet/frame-based communication for the CAN, Ethernet, and LIN. Further automotive field busses are on the roadmap.
- **Multi-host**: It provides a true distributed simulation on multiple execution hosts. It is cloud ready including container support. It does not rely on a central data-broker to avoid congestion and is scalable to the vehicle level.
- **Multi-Platform**: The eDT-fabric supports Linux and Windows OSes. Thus, it aids with restrictions regarding model availability, licensing, and dependence on platform specific APIs.
- **Multi-Architecture**: The core of the eDT-fabric is available on both Intel X86/X64 and Arm aarch64 host architectures. Simulators with dedicated hardware requirements, e.g., GPUs, NPUs or other hardware accelerators may run on dedicated machines that satisfy those requirements. Reserving the costly hardware for those participants in a co-simulation, that really need it may result in overall improved resource utilization.

The eDT-fabric technology complements the Synopsys co-simulation portfolio which consists of

- the Virtualizer Systems Interface (VSI 2.0) [6] - for co-simulation of level-4 ECUs with each other and with plant models, and
- Silver with its Synopsys Virtual Bus (SVB) - the co-simulation engine for vECUs at levels 3 and below.

Both VSI 2.0 and SVB rely on shared-memory-based communication and thus are very well suited for fast single host co-simulation of tightly coupled systems on a single host.

By contrast, the eDT-fabric has been designed with the distribution of the co-simulation on multiple hosts in mind from day 1. In general, we are targeting larger step sizes and assume that the vECUs communicate with each other through automotive networks or their abstraction in form of idealized communication at the software-signal level.

For more tightly coupled subsystems VSI 2.0 with its frontend Multi-Node and/or Silver can be used to co-simulate as part of an overall eDT-fabric co-simulation. The above example also illustrates that the eDT-fabric supports existing standards like the FMI standard for co-simulation. We provide a generic FMU that can be customized and imported into existing FMI importers, e.g., environment simulators or plant-model simulators to connect them to the eDT-fabric.

In the remainder of this Section, we will describe some of the key-features of the eDT-fabric.

5.1 High performance

When shifting to distributed co-simulation network latencies and bandwidth between simulation hosts need to be considered. Data exchange strategies that may be optimal for single-process and/or single-host co-simulations based on shared memory turn out to be suboptimal in the distributed case, as they assume that data can be exchanged between participants infinitely fast by swapping a bit in the shared memory.

The eDT-fabric therefore comes with a new loosely coupled synchronization and data exchange mechanism specifically tailored to support mixed abstraction. It exploits the fact that simulation speeds greatly vary between less abstract simulations at levels < 3 and detailed level-4 simulations. Even between different level-4 simulations performance may vary a lot from step to step. The above-mentioned algorithm exploits such load imbalances to reduce the overhead of the network communication for synchronization and data exchange.

Our full parallel launch mechanism that enables the user to kick-off an entire co-simulation from a central command line interface (CLI) cockpit substantially leads to the fastest possible startup time. Through respective configurations the authors of a co-simulation setup have full control over whether they want to run certain participants in headless mode, or whether they want their UI popup on the respective host to do some debugging.

5.2 Microservice-based architecture

The microservice-based architecture of the eDT-fabric includes a distributed life-cycle management for simulation and tool processes. Execution hosts for the eDT-fabric will run this service. Through the life-cycle management service an instance of the CLI cockpit may start, stop, and monitor preconfigured simulators which take part in a co-simulation.

To establish the communication matrix between the co-simulation participants, e.g., the vECUs and Plant models, a dedicated orchestration service is spawned by one of the involved lifecycle managers. The orchestration service is only involved in the co-simulation at the start and at the end, and whenever participants join or leave the co-simulation. For example, a user may be interested in tracing data in a specific interval of time. The tool used for the tracing would register itself through the orchestration service.

Note, that synchronization and data exchange between the simulators is not routed through the orchestration service, to avoid that its network interface becomes a central bottleneck in larger setups.

Simulations taking part in an eDT-fabric co-simulation also provide services to their peers, to the orchestrator, and to their lifetime manager. The participant service receives incoming synchronization and simulation data from other simulators. The orchestrator service enables the orchestrator to inform the participant about changes in the communication matrix. The lifetime manager can send commands, e.g., to gracefully stop the simulation.

The CLI-cockpit interacts with these services, e.g., for status and error retrieval, and to send commands to the participants.

5.3 Deployment

The eDT-fabric comes with a documented open API. This API is used by all integrations for Synopsys vECUs and external plant-models and tools into the eDT-fabric. We also provide a generic functional mockup unit FMU that supports the FMI standard for co-simulation. This FMU can be used to integrate third-party FMI importers into the eDT-fabric.

We support virtual networks for automatic protocols and idealized signal communication between vECUs. The API also provides a custom message type that allows to exchange binary raw data between co-simulation participants.

Compositions for distributed co-simulation are human readable and easy to manipulate through scripts or UIs. The eDT-fabric provides scripting support for their generation.

5.4 Advanced features

This section completes the feature overview of the eDT-fabric and outlines three of the advanced features that enhance performance and/or support fast debug turnaround times.

5.4.1 Record and Replay

The infrastructure of the eDT-fabric comes with a built-in option for recording the traffic from and towards each individual integrated simulator. This enables two different use-models for replay of traffic in debugging scenarios outlined in the sequel.

Fully isolated mode: For in-depth debugging, the user of the eDT-fabric can run a specific simulator standalone and use the input stimuli to that simulator from a recording taken in an earlier run of the complete setup. This enables extremely fast turnaround times for determining the root cause of an issue. Obviously, this requires narrowing down the root cause of a functional issue to a specific participant in the co-simulation. Here is where the second use-model for replay comes in handy.

Replace specific nodes by recording: Individual co-simulation participants (e.g., vECUs or plant-models) may be replaced by a dedicated replay node. This executable, that comes as part of the eDT-fabric installation, reads in a recording of the API calls of the participant, and exactly reproduces its behavior for the rest of the co-simulation. This can be used during debugging to remove, e.g., certain slow participants after they have been ruled as the source of an unintended behavior. As debugging typically requires multiple debug runs this saves time for the developer and lets him focus on those parts of the system, that are relevant for the current problem.

A second motivation for introducing replay nodes deals with the availability of models. At early stages of the integration the software for some of the ECUs may not be available yet. However, some of their behaviors may be crucial to be able to integrate and test already some of the other components. The eDT-fabric supports this by a recording format that is easy to generate and human readable and thus allows for generating a recording file without running the eDT-fabric. In this manner the replay component becomes a mock for the missing vECUs in early stages of the overall integration project.

5.4.2 “Zero”-overhead Tracing

Co-simulation participants in the eDT-fabric can be configured as monitors. In this mode, a participant will only read data from the eDT-fabric. With a good caching strategy and with vECUs in the picture that do real work, the rest of the system will hardly ever have to wait for these tracing nodes that do nothing but storing away traced data. Thus, the overhead for tracing participants is usually negligible.

5.4.3 Performance Profiling

Optimizing complex systems for performance, requires insight into where the bottlenecks of a system are. The eDT-fabric has a feature to collect profiling data for the participants of a co-simulation. It comes with a mechanism for time synchronization across machines and with visualizations for the following characteristics:

- Work and wait time diagrams to analyze bottlenecks is the co-simulation. On a per-step basis these diagrams show the duration that each simulator spent simulating and the duration that the simulators spent waiting for inputs from other simulators.
- Cumulative work and wait time diagrams.
- Real-time factor including and excluding wait times for comparison of achieved performance with theoretical limit.
- Concurrency visualization including cross-step concurrency for participants that only communicate indirectly or unidirectional.
- Step-Size variations for each of the simulators.

6 Case Study

As a demonstrator, we applied the eDT-fabric to simulate an adaptive cruise-control application with collision avoidance in a mixed abstraction setup. The overall co-simulation setup is illustrated by Figure 3. It consists of level 4b vECUs for the central compute cluster und the zonal/domain controller and a control vECU at level 3. The environment simulator IPG CarMaker provides external stimuli and the Silver cockpit is used for visualization and tracing.

The overall setup is mapped onto three execution hosts. A developer laptop with dedicated GPU runs CarMaker, a windows terminal server runs the Silver control ECU and the Virtualizer VDK for the zonal controller. Finally, the Arm based central compute cluster is mapped onto a hypervisor-based simulation running on an Arm aarch64 host.

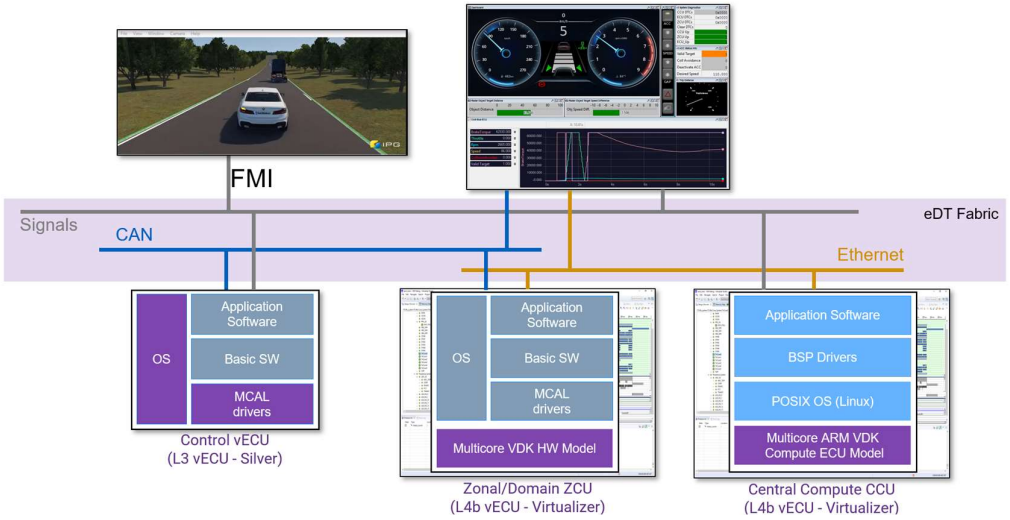


Figure 3: eDT ACC + collision avoidance mixed abstraction.

The central compute and the zonal controller are connected through Ethernet, the control ECU and the zonal controller through a CAN bus, and the environment stimuli are exchanged as software Signals.

The visualization and tracing cockpit visualizes the data and can be used to connect tools like Wireshark to inspect the network traffic.

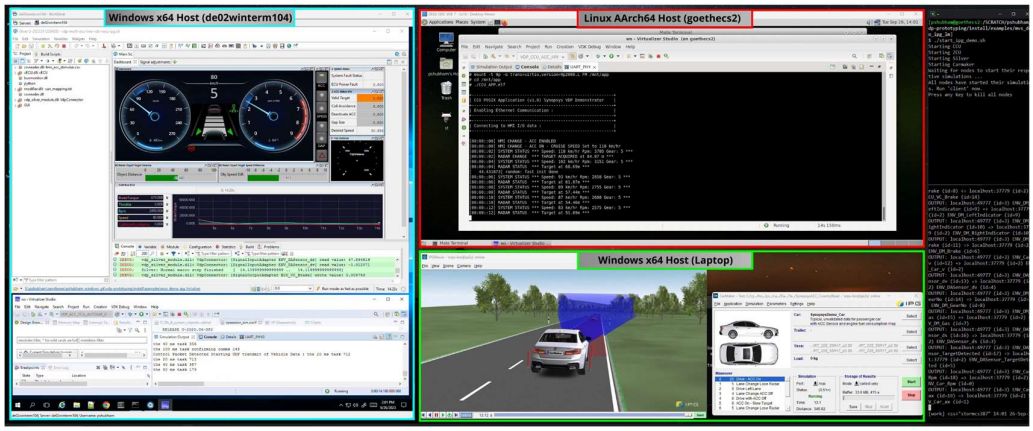


Figure 4: Screenshot of the running co-simulation

7 Conclusion

In this paper we introduced our new eDT-fabric for distributed multi-host, multi-abstraction, multi-host-platform, and multi-host-architecture co-simulation. It provides a rich feature set backed by a novel and performant distributed data exchange and synchronization algorithm. Its micro-service-based architecture makes it cloud ready and enables easy integration in existing customer cloud distribution solutions. Advanced features for analysis, profiling and debugging let our customers focus on their goal to develop innovative high quality embedded software that runs tomorrow's cars.

8 References

[1] T. DeSchutter, Ed., Better software. Faster! best practices in virtual prototyping, Synopsys,, 2014.

[2] I. Feldner et al., "The automotive virtual prototyping platform," Virtual Platform Working Group of the Arbeitskreis Automotive of the EDACentrum., 2019. [Online]. Available: https://www.edacentrum.de/en/whitepaper-automotive-virtual-prototyping-platform.

[3] FMI, MODELISAR Consortium, "Functional Mock-up Interface (FMI) specification," 10 05 2022. [Online]. Available: https://fmi-standard.org/.

[4] "IEEE Standard for Standard SystemC Language Reference Manual," *IEEE Std 1666-2011 (Revision of IEEE Std 1666-2005),* pp. 1-638, 2012.

[5] F. Thoen et al., "Whitepaper: Accelerating Development of Software-Defined Vehicles with Virtual ECUs," 2024. [Online]. Available: https://www.synopsys.com/verification/resources/whitepapers/virtual-ecu-wp.html.

[6] Prostep IVIP, "Whitepaper Virtual Electronic Control Units," 03 2020. [Online]. Available: https://www.prostep.org/fileadmin/downloads/WhitePaper_V-ECU_2020_05_04-EN.pdf. [Accessed 22 05 2023].

[7] Synopsys Inc., *Synopsys Virtualizer Co-Simulation with Third-Party Simulator manual,* Moutain View: Synopsys Inc., 2023.

Enhancing Automotive Testing Efficiency with AI-Driven Monitoring and Assessment Solutions

Geevarghese George, Georgios Koutroulis, Gerhard Schagerl, Milan Zivadinovic

{geevarghese.george, georgios.koutroulis, gerhard.schagerl, milan.zivadinovic}@avl.com

Abstract: Due to the ever increasing worldwide competition between premium automotive manufacturers and the stringent regulation standards, novel methods from the field of AI are required to substantially decrease developments costs without sacrificing the end quality of the product. However, it is challenging, first, to identify the right use case within the automotive product development and testing phases, and second, to choose the appropriate AI algorithms that will address the problem. In this article, three innovative solutions are discussed that aims at enhancing the testing efficiency, and ultimately decrease development times and costs.

First, an anomaly detection framework is presented for testing of newly developed internal combustion engines (ICE). Next, we present a data-driven methodology to perform End-of-Line (EoL) quality check to detect abnormalities in a semi-automated manner. Finally, a transfer learning approach for accurately estimating the State of Health (SoH) of e-vehicles is discussed. We demonstrate the feasibility and benefits of utilizing such statistical approaches for handling large amounts of telemetry data. The proposed solutions have the potential to automate and accelerate the testing process while minimizing the risk of faulty units.

1 Introduction

The demand for and transition to electric vehicles (EVs) has introduced new challenges in the automotive industry, with leading manufacturers reporting a loss of $20-30k per vehicle sold relative to the sales volume [Ltd]. Moreover, various economic factors in the post-pandemic era have driven up costs throughout the product lifecycle, even as efforts to further reduce greenhouse gas emissions and improve safety are intensifying. To this end, performing any kind of vehicle testing during the development phase, ranging from measuring emissions to validating the durability of the engine or an electric component is of paramount importance.

Such testing procedures are complex and need to be carefully designed, executed and ultimately monitored. Testbeds provide a controlled environment where various components, systems, and the entire vehicle can be rigorously tested and validated. These testing procedures typically generate large amounts of data which may be further analysed by modern data-driven techniques to extract valuable insights. Automated and efficient processing of this data supplies fast and accurate feedback to the development teams, enabling them to

make informed decisions and making effective decisions in the upstream developmental process. In this paper, we focus on developing methods to address these challenges in the different phases of vehicle development and manufacturing.

2 Methodology

In the following sections, we describe three data-driven solutions, focusing on their methodology and impact on the automotive industry during the product development and testing phases. In Section 2.1 we describe the application of a graph neural network-based anomaly detection framework for automotive engine testbeds. In Section 2.2 we discuss a statistically derived methodology for classification of EoL testing data of engines. In Section 2.3 we present the methodology for the accurate estimation of battery SoH leveraging transfer learning.

2.1 Anomaly detection in automotive engine test-beds

In the context of vehicle development, anomaly detection [Agg] can be used to identify issues in the development process, such as faulty components or suboptimal configurations. The endurance test is a typical validation and verification (V&V) cycle conducted on a testbed to verify the reliability of an internal combustion engine (ICE). This performed by running the engine under a specific load and speed profile for a predetermined period of time. Data are sampled at a relatively high frequency (1Hz) by sensors placed on the engine and the testbed to generate multivariate time series (MTS) data, which may include channels such as torque, rotational speed, temperature, and pressure.

2.1.1 Testbed data description

The dataset consists of multiple test runs with varying test scenarios and operating conditions for a unit under test (UUT). It includes 864 measured channels (sensor readings), approximately an hour long, spread across 575 measurement files. Channels are manually categorized into three classes: *output channels* (customer-defined), *important channels* (as identified by AVL domain experts), and *unimportant channels*. Channels irrelevant to the UUT, not present in all files, constant, erroneous, and redundant channels were removed based on expert review. The raw dataset does not contain any labeled anomalies. However, the engineers maintain a testbed diary which contains the information about the test runs and the anomalies detected during the test runs. This information is used as a ground truth to validate the detected anomalies by the model.

Characteristic patterns in the testbed signals, including level shifts at the beginning and high fluctuations at the end, were observed. The model should learn these patterns as normal behavior and detect anomalies such as sudden noisy spikes or drops in the signals. The anomaly detection framework is applied to such high frequency MTS data in an unsupervised [HTF] manner to extract channel interdependencies and forecast UUT failures.

2.1.2 Anomaly detection framework

The anomaly detection model that we employed is based on the recent developments in the domain of geometric deep learning. More specifically, we used the Graph Deviation Network (GDN) [DH] which is a graph-based model that uses Graph Attention Networks (GATs) [VCC+] to forecast the future values of a MTS.

The main benefit of a graph-based MTS forecasting model is that the model is able to learn non-linear, temporal dependencies along with structural- relationships between different channels. In order to achieve this, LASSO feature selection [HTF] is applied to the *important* and *output* channels of the MTS. The selected features are used to create an adjacency matrix, which is then transformed into embedding vectors to learn the graph structure. The GATs model takes these embedding vectors and self-lags [Box] of the channels, where a masked attention operation [KW] is performed over the edges of the graph.

The model performs a rolling forecast with a horizon of one timestep. The forecasting error is measured by common regression metrics such as mean squared error (MSE) and coefficient of determination (R^2). For quantifying a value as an anomaly, a model specific metric, known as the Graph Deviation Score (GDS) [DH] was also used. If the forecasted value is outside a threshold δ, set to the maximum GDS value per file, the model flags the value as an anomaly. The full workflow is depicted in Fig A.2.

2.2 Classification for End-Of-Line testing of engines

While the previous section focused on anomaly detection methodologies for testbed data during endurance testing of ICE engines, the following section describes a series of cold test (non-operating engine) measurements conducted sequentially as part of the EoL quality check. A typical EoL test sequence involves comprehensive evaluations, including initial inspections, cold start tests, step-wise RPM changes under varying load conditions, and emissions assessments. Each test sequence focuses on specific engine parts or behaviors and is associated with a specific set of channels.

The current practice during analysis of the recorded data is to use global extrema values (minimum and maximum) for each channel to detect possible faults. For this, a short window is selected and comparison is made with the predefined bands to check if the extrema value lies within the allowed range. If the value falls outside the predefined bands, a fault is reported. However, this methodology has drawbacks as the thresholds are set empirically and may not be optimal for all cases.

2.2.1 EoL data description

Each sensor ideally produces a time series with same length and frequency attributes during the measurement. We selected the single largest group of engine type, tested with the same execution sequence as the source data for the proposed EoL solution. Even though the data from such a test sequence is of high precision, the time series can have noise and time-misalignment. In those cases, the original time series may also need filtering and

time scale modification to align the time series.

2.2.2 EoL fault detection framework

Instead of predefined boundaries within time windows, we utilize a mathematical model that captures normal behavior of a time series. We employ a widely known sampling technique in statistics called *bootstrapping*, to construct a reference curve. To perform this, we select random samples of time series from a population of recorded time series. A mean signal is constructed by point-wise averaging of the selected sample. This process is repeated several times (hundreds to thousands) to create a new population of averaged time series'. The new population, according to Central Limit Theorem [Wil], will approach a normal distribution for a large sample size. The mean of this new population is the best estimate of the mean of the initial population of recorded time series. Based on the new population, the reference curve is constructed as the mean of the new population[1]. For each data point of the reference curve, a point-wise standard deviation (σ) can also be measured. The confidence intervals can typically vary for several standard deviations. Since the reference curve data points follow the Normal distribution, approximately 99.7% of the observations should fall within $\pm 3\sigma$ range. By relaxing the range to $\pm 4\sigma$, we should practically cover all *normal* observations. During thee classification phase, any data point outside this range can therefore be characterized as an *outlier*, using any statistical hypothesis testing such as a z-test.

2.3 Estimation of battery State-of-Health of electric vehicles

The current market for accurate SoH estimation is on the rise, with equal demand for cloud-based SoH estimators and predictive maintenance solutions. In this section, we demonstrate a data-driven machine-learning based approach as an alternative to the traditional physical state observer-based SoH estimation. Typically before the production of battery packs starts, in the design phase, the characteristics of cells are measured in a controlled laboratory environment. On the otherhand, recorded SoH data from the field is known to be noisy and inaccurate. To combat this issue with fleet data, a solution based on transfer learning [PY] was developed at AVL for accurate SoH estimation.

2.3.1 SoH data description

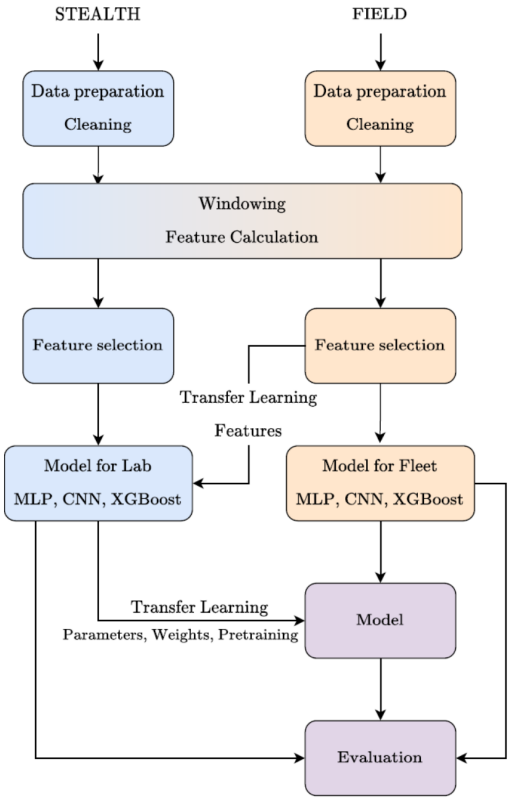
The data used for the SoH estimation usecase was collected from two different sources: laboratory data (STEALTH) and fleet data (FIELD). We collected data from two sources: laboratory (STEALTH) and fleet (FIELD). For the laboratory data, we cycled 8 lithium-ion cells on two testbeds. The relevant channels were current, voltage, and temperature. The data included aging cycles with periodic capacity measurements, tested over 100-200 days at a sampling rate of 4Hz. For the fleet data, we gathered telemetry data from 208 vehicles operated between 2019 and 2022. The relevant channels included current,

[1] Please note that such a reference curve can only be constructed for channels that show similar behavior on set of historical reference engines.

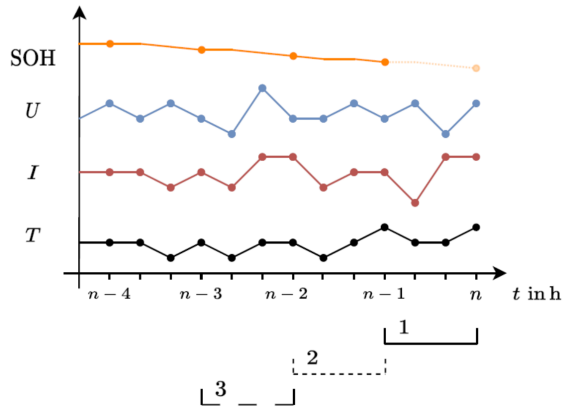
voltage, temperature, and SOH from the onboard BMS. The data was measured at a lower sampling rate of $1/30$Hz compared to STEALTH.

2.3.2 Framework for SoH Estimation

The proposed framework for SoH estimation leveraging transfer learning is depicted in Fig. 1a. The first step is to separately perform data cleaning, plausibilization and standardization for the two datasets. The preparation stage also includes the calculation of SoH from capacity, followed by adding a resting (indicating charge, drive or storage mode) channel. Several common statistical, energy-based, SoC change related and heatmap features were calculated additionally.



(a) Proposed framework for SoH estimation using transfer learning.



(b) Representative figure demonstrating the window used as feature and the last point of SoH as target.

Figure 1: A data driven SoH estimation framework and training methodology.

We use a window-based approach with 1-hour and 120-hour windows to compute input features. The last SoH value in the window is used as the target value (Fig. 1b). We additionally performed feature selection using LASSO coefficients [HTF] and recursive backward elimination.

Three classes of machine learning algorithms, namely multilayer perceptron (MLP), convolutional neural network (CNN), and extreme gradient boosted trees (XGBoost) were compared for SoH estimation. By making sure that the features were generalizable across the two datasets (Fig. 1a), we were able to train the models using transfer learning. Trans-

fer learning involves using knowledge from one domain to improve performance in another. In this case, the models were first trained on the STEALTH dataset and those model weights were fine-tuned for the FIELD dataset. This is appropriate since the area of application and modelling tasks for the two datasets are be the same. Please note that for XGBoost, there are several ways to perform transfer learning, such as transferring hyperparameters, continued training, and continued update, all of which were experimented with.

3 Results and Discussion

As described in Sec 2.1, the GDN model aims to learn the multivariate dependencies of the input channels during training. The graph structure that the model learned is shown in Fig A.1. The validation of the graphs was performed by domain experts, who described the graphs as excellent visual indicators of UUT behavior and a valuable tool for the detection of meta-trends. The generated graph had a total of 57 nodes (channels) of which 53 had edges (relationships) that reflected the real interdependencies to the UUT.

Another result from our study is that GDN excels in meta-trend analysis, providing clear and low-noise trend detection (Fig 2). It outperforms benchmark models[2] by identifying not only known trends but also discovering additional anomalous sequences related to erroneous behavior in the Unit Under Test (UUT). In terms of the step-ahead forecasting task, GDN performs satisfactorily, with mean squared error (MSE) of 0.0052 ($\sigma = 0.0064$) and R^2 of 0.7727 ($\sigma = 0.0052$), however it is not as performant as benchmark models.

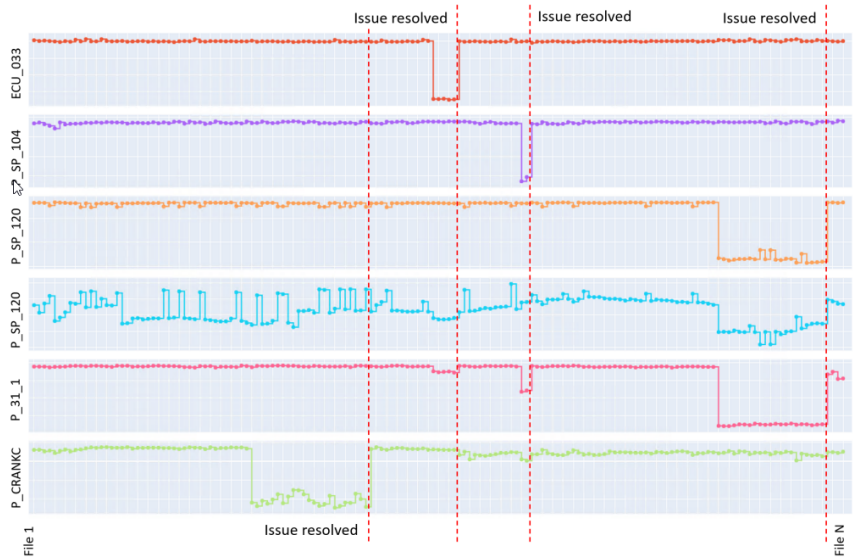
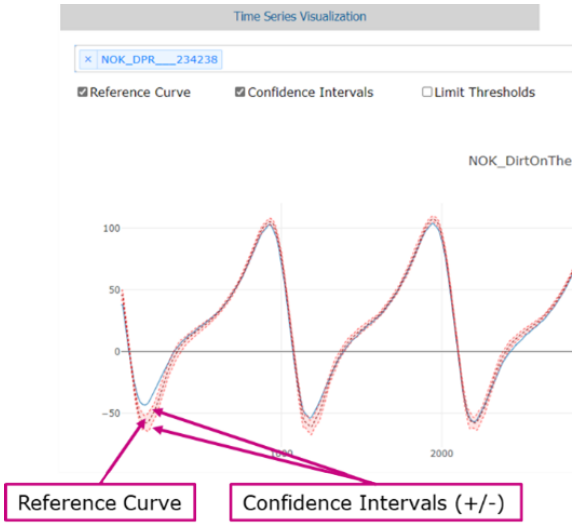


Figure 2: A section of six representative channels for unseen test files. The vertical axes show the R^2 metric per channel and the horizontal axis indicates the respective test files.

In Sec 2.2, an enhanced methodology for classification of EoL testing data of engines was

[2]MLP and LSTM were benchmarked on the same task.

presented. The main requirement for the OEM is to correctly classify possible erroneous engines during cold test in order to direct them to the second phase of a hot test. The presented methodology has helped OEMs to classify engine faults accurately and to reduce the number of false positive and false negative detections. A dashboard integrated with the AVL Data Analytics™ platform allows for interactive, real-time visualization for the operator along with the possibility to automate the process. A snapshot of the dashboard indicating the reference curve, adjustable confidence intervals and observed signal (Torque) is shown in Fig. 3a.



(a) Dashboard indicating the reference curve, confidence intervals and observed signal using the proposed methodology. The confidence intervals at the point level can be adjusted by the operator based on the boundary conditions.

(b) Summary statistics for each severity band regarding the ratio of points found in each severity band (confidence interval). Additionally, the middle point in the sunburst figure represents the overall fault severity measured by mean L_1 and L_2 norm.

Figure 3: Artifacts from the point-wise severity analysis of EoL testing data.

The fault severity is determined by the distance between the observed signal and the reference curve. This classification allows for a more granular distinction of the measured data points based on confidence intervals. This information can be visualized as part of the time series, or more conveniently as a sunburst chart as shown in Fig. 3b. Finally, the reference curve is dynamically updated to guarantee drift event amortization.

In Sec. 2.3, we described the methodology for the accurate estimation of battery SoH leveraging transfer learning. The important features for estimation included min, max, median SoC, mean current, temperature-based, and heatmap features. Additionally, energy throughput and resting time were relevant. Heatmap features were only useful for the FIELD dataset as the STEALTH dataset does not reach extreme temperatures. Lower SoH values and higher energy throughput were observed in the STEALTH data due to long age cycle tests.

With transfer learning, significant improvements were observed in comparison to training from scratch. For example, MLP showed a 35% reduction in the loss from start, and XGBoost showed a 36% reduction. The SoH estimation made with transfer learning of the XGBoost model is shown in Fig. 4. Pretraining on the STEALTH dataset helped the

models converge faster and achieve better performance. XGBoost, with the transfer of hyperparameters strategy, resulted in the best performance, but other strategies mentioned in Sec. 2.3.2 were also competitive.

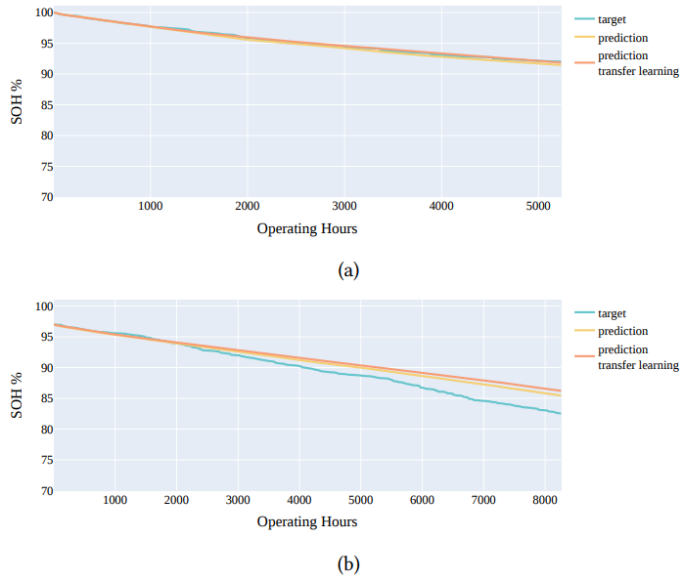


Figure 4: SoH estimation using XGBoost: The two plots show the (a) best, and (b) median prediction in terms of loss from start in the test set, for the model trained from scratch (yellow) and the model with transfer learning (orange) from STEALTH.

4 Conclusion

The automotive industry is undergoing a significant transformation with the advent of electric vehicles and the increasing demand for sustainable transportation solutions. Leveraging the vast amount of data generated during the vehicle lifecycle is crucial to minimize costs and improve the efficiency of the industry as a whole. In the near future, EU's data and AI strategy through the Digtial Europe Programme for shared data spaces would ease the barrier to data sharing across sectors and industries, including the automotive domain, to foster innovation and improve the quality of products and services.

In this context, we have explored how AVL can continue to support OEMs in addressing challenges related to testbed validation, end-of-line testing and state-of-health estimation for preventive and proactive action. Automation and digitalization tools such as AVL's Data Analytics™ is integrating these methodologies to provide customers with real-time feedback from various phases of the vehicle lifecycle.

The methodology we have developed for anomaly detection provides a novel approach for detecting meta-trends in time series data. Recent work carried out at AVL in the context of EV fleet monitoring has shown that GDNs are performant in detecting point anomalies [TMC+] as well. The technology presented for EoL testing has already been offered to OEMs to help them circumvent downtrends in engine production quality. The integration of the proposed methodology into existing workflows has the potential to significantly

impact the testing phase by improving the quality of detections, reducing the time spent to analyze faults, and consequenty reducing the cost for engine testing. We also demonstrated that estimating fleet-wide SoH using transfer learning of models trained on laboratory data is a viable approach. Integrating such models into the BMS or on a cloud-based environment for real-time estimation and predictive maintenance is the logical next step.

5 Acknowledgements

This work was partly supported by the SM4RTENANCE (European Deployment of Smart Manufacturing Asset 4.0 MultilateRal DaTa Sharing SpacEs for an AutoNomous Operation of CollAborative MainteNance and Circular SErvices) consortium, which received funding from the Digital Europe Programme of the European Union Under Grant Agreement No 101123423.

References

[Agg] Charu C. Aggarwal. *Outlier Analysis*. Springer International Publishing.

[Box] George Box. *Box and Jenkins: Time Series Analysis, Forecasting and Control*, pages 161–215. Palgrave Macmillan UK.

[DH] Ailin Deng and Bryan Hooi. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series.

[HTF] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer.

[KW] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks.

[Ltd] The Editorial Team at Plunkett Research Ltd. The Global Automotive Industry: Key Challenges and Opportunities.

[PY] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. 22(10):1345–1359.

[TMC+] Thomas Alexander Kristan, Milan Živadinović, Christian Rupert Rehrl, Roman Kern, and Alwin Tuschkan. AI-Based Digital Twin - Anomaly Detection and Diagnostics for HV Battery Behavior and Performance. 2024 JSAE Annual Congress (Spring).

[VCC+] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks.

[Wil] Rand R. Wilcox. *Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy*. Springer.

A Supplementary Figures

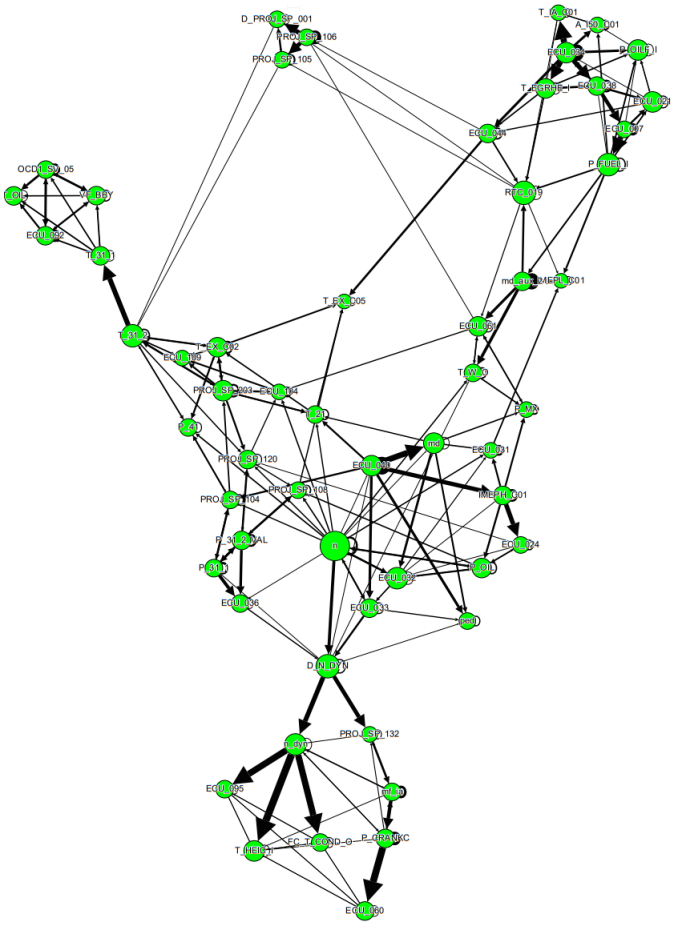


Figure A.1: The learned graph structure by the Graph Deviation Network (GDN). Each node is a channel and edge corresponds to relationships between the channels.

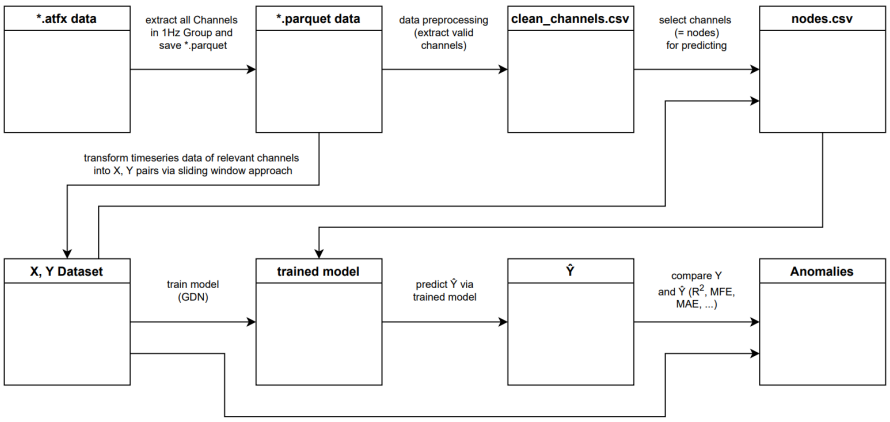


Figure A.2: Workflow for the anomaly detection task in automotive engine test-beds.

Moving towards explainable AI by utilizing a holistic lifecycle assurance framework

Lukas Klose M.Sc., Dr.-Ing. Lutz Kelch

Keysight Technologies Deutschland GmbH
Automotive & Energy Solutions
Mittelweg 7
D-38106 Braunschweig
lukas.klose@keysight.com
lutz.kelch@keysight.com

Abstract: As a result of the exponential increase in complexity of rule-based systems processing a large amount of data, the move to AI and data driven processes is inevitable.

This process requires a shift of testing and assurance paradigms.

Implementing AI models in automotive systems requires a novel approach of testing and documenting the process of decision making. Depending on the level of safety, different levels of interpretability and test coverage, as well as documentation are required. Fueled by the proposed Acts on AI, the need for explanations of the system is imminent.

Keysight Technologies, a market leader in test and measurement, proposes a novel holistic framework combining tests covering the complete lifecycle of an AI system, from initial conception to the last day of deployment, standardizing test executions and providing insights into the decision-making process of the Model under Test (MuT). While aligned with requirements stated by governmental institutions, the proposed solution enables AI Engineers as well as Domain Experts to join forces and improve their models based on insights and recommendations offered by the framework.

Separating the lifecycle into five steps (Problem understanding and Data Analysis, Feature Engineering, Model Training, Deep Model Evaluation, Model Inference) allows integration of the framework into different software engineering processes, like V-Model or Waterfall, necessary for safety critical systems, and platforms.

1 Introduction

The recent achievements in Artificial Intelligence (AI), especially in the realm of Machine Learning (ML) will accelerate the already ongoing inclusion of AI driven Advanced Driver Assistance Systems (ADAS) and fully Automated Driving (AD) systems.

With the adoption of AI into high-risk applications, a novel approach for testing those components must be implemented to ensure safe and secure operation, as well as strengthen the confidence of users in the reliability of the system.

Other than classic rule-based systems or decision trees, Neural Networks (NN) are black boxes by design. During the training process, a NN learns relations between the input data and the desired output by adjusting the weights between the different neurons. Depending on the design of the network (influenced by the complexity of the task which shall be solved using NNs), this number of weights can vary between several hundred and up to billions of parameters. The greater the number of parameters, the more difficult it is to interpret them and draw conclusions about the model behavior. However, the understanding of the inner working of a NN is crucial when relying on the correctness of its decision with human life at stake, especially in but not limited to corner case scenarios.

Therefore, the goal is to achieve interpretability and explainability of NNs to ensure their desired behavior even in critical situations. Moving towards general explainability, one will have to answer the following questions:

1. Is the available data relevant and sufficient to train the network?
2. At which point is the network fully and optimally trained?
3. How does the network deal with corner cases and scenarios not included in the training data?
4. And after deployment: Is the network still fitted to the current domain or did the domain change, rendering the current network unable to handle its task?

In the following sections, we will present a framework for answering those questions mentioned above and helping to speed up the development of high-quality AI solutions while cutting costs by providing support for engineers during the entire lifecycle of the ML model.

2 Problems of current explainability methods

Explainability and interpretability are necessary to achieve a deep understanding of the behavior, reliability and limitations of AI systems. During operation in ADAS/AD systems, the AI models will have to deal with unknown, e.g. not trained, scenarios. Especially in this domain, corner cases and extreme situations will be safety critical. Like in classical testing, it is not possible to evaluate every possible situation. Therefore, one must understand how and why the AI model decides the way it does. Utilizing local explainability methods can help to determine how different features influenced the decision of the AI model and to understand the cause of failures in training and deployment.

However, explaining faults after the fact is not sufficient. The goal must be to understand the AI system during the training and evaluation process and learn its limitations. This requires an understanding of the quality of the network, the training state and adaptation to the training data, as well as thorough tests on high quality test data unknown to the network.

Currently, no general method for achieving complete explainability and interpretability of common Neural Networks exists. If explainability is necessary, one must utilize inherently explainable AI models like Decision Trees.

If those models are not feasible to use, a combination of different interpretations, evaluations and test metrics can be utilized to get better insight into the behavior of the ML model.

Each metric can provide information for a specific part of the process of training a NN, e.g. the quality and distribution of data used for training and testing, the certainty of the network or the robustness to adversarial attacks, etc.

The downside of this approach is that it doesn't combine the results of different metrics to improve the understanding of the model's behavior, as well as the differences in implementation of the available metrics, forcing the developers to constantly write wrappers to enable the use with their specific model and or data format.

3 xpl[AI]ned: Moving towards explainability

To help customers create reliable AI faster, Keysight proposes a novel framework for testing, validating and assuring AI models alongside the complete lifecycle, while offering insights into the model along with documentation and enabling AI Experts and Domain Experts to join forces more effectively.

Fundamentally, we divide the AI lifecycle into five different stages:

1. Problem Understanding and Data Analysis
2. Feature Engineering
3. Model Training

4. Model Evaluation
5. Model Inference

As can be seen in Figure 1, each stage is directly connected with the previous and the next stage, creating a circular lifecycle model, where, after a model inference, a new iteration starts. With each iteration of the lifecycle, the quality of the model should improve, mitigating shortcomings detected in the inference stage. Due to the nature of ML development, changes in the model architecture or data structure might be necessary, as the model's behavior differs from expected values.

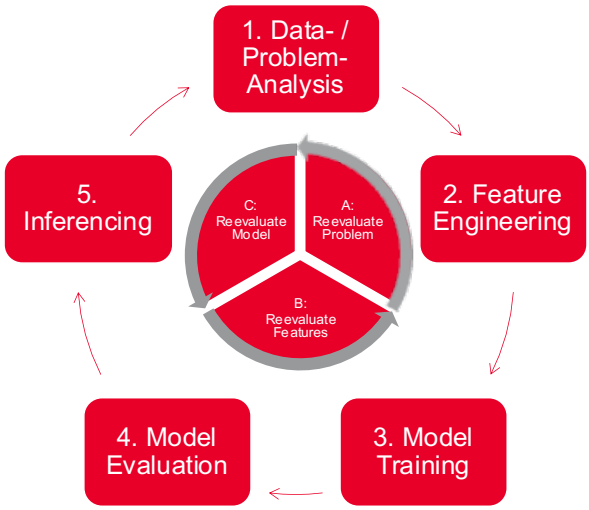


Figure 1: Proposed connected Lifecycle Model

Keysight's solution provides a framework for developing ML models within this lifecycle, allowing seamless changes between stages in both directions. The framework, as the process itself, is divided into five different test categories, aligned with the stage names. Each test category, by default, has several reference tests implemented by Keysight, which can be used out of the box. Each test will automatically generate a report and can be cross-referenced with other metrics. For example, when analyzing the pixel-wise accuracy for specific objects within a semantic segmentation model, the accuracy might be correlated to the overall number of pixels of this class within the dataset and the percentage of pixels belonging to this class on this specific location.

The AI model solving complex driving tasks will still need to interact with classic system and software development. Data busses, sensor / actor interactions and data processing are just a few examples of components which need to be developed alongside the AI component. This poses the challenge of aligning the two different approaches. With the proposed methodology it is possible to align the iterative process of ML development to the more static approach of the V-Model,

maintaining the well-established process for software development while enabling a test-driven AI development from the beginning, as can be seen in Figure 2.

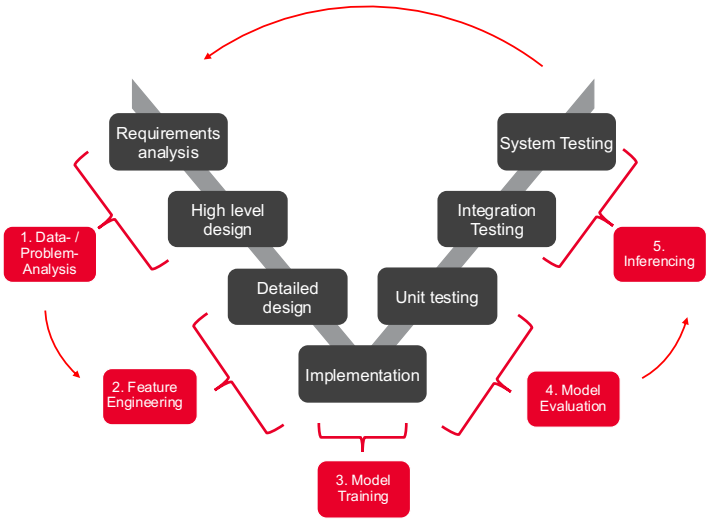


Figure 2: Alignment of Lifecycle Model to V-Model

While a general goal is to achieve total explainability and interpretability of those black box models, with the current state of technology only a partial explainability, e.g. a feature's influence on the outcome of a specific prediction, is possible. However, the field of AI research is moving at an extraordinary speed, so that one key feature of the platform is the easy extendibility by custom plugins. This allows on the one hand to quickly implement new tests and make them available, on the other hand it enables custom plugins for specific use cases, which might not be generally applicable to Deep Learning (DL) models but are required by stakeholders. The easy to implement plugin solution allows for an addition of custom metrics into the software, further accelerating the development lifecycle and eliminating the need for another solution.

4 Foundation for recommendations

By combining results from different tests from multiple stages, Keysight's solution can derive recommendations for improvement. These recommendations can be based on norms and guidelines, governmental requirements, or known good data (golden datasets).

Depending on use case and domain, there will be varying regulatory requirements, like the EU AI Act, defining specific minimum quality standards. These may be extended by industry standards, like DIN SPEC 13266, ISO 26262, ISO/PAS 21448 and the upcoming ISO/PAS 8800.

Keysight continuously monitors changes and extensions to those norms, guidelines and governmental requirements and classifies the results of tests accordingly. Based on this classification, recommendations are generated on how the current stage could be optimized. In addition, a comprehensive report is automatically generated for each test execution, containing all information about the model, used datasets, test configurations and results. This report serves two purposes; it can be used to keep track of different experiments within the organization and inform additional stakeholders like Risk Officers. As the documentation is also a proof of the ongoing endeavor to provide the model with the highest possible quality, it can also be used in case of an accident to work with the prosecution to prove that there was no negligence in the development process of the AI system. Secondly, those reports also contain recommendations for improvement, pointing towards problematic areas and causes, e.g. an uneven data distribution or low information content of the features used for training.

On the way towards explainability the combination of different metrics can enhance the transparency of the AI system, hinting towards limitations and helping to locate faults, not only in the training but also in the data used for training.

By utilizing a “golden dataset” created by Domain Experts as a foundation, the data selection and model training can be automatically checked against these datapoints, extrapolating to the problem areas which need attention. This golden dataset can also be a representation of the Operational Design Domain.

By defining the domain boundaries to match with the boundaries of the golden dataset, the framework can use test results from the golden dataset as a reference for checking the quality of the model.

While the task of creating a golden dataset may be challenging, its usage within xpl[AI]ned enables Domain Experts without a deep knowledge of AI to better apply their domain specific knowledge. This helps to further improve the model by creating a common basis for discussion, understood by AI and Domain Experts, making the black box more transparent. Combining the results of the metrics, a maturity score of the model can be calculated, serving as foundation for deciding whether the model is ready for deployment.

5 Case Study: Automatic analysis of Pedestrian detection

For most ADAS/AD functions, the camera is an important sensor. Other road users, drivable and non-drivable areas, traffic signs, obstacles and other information can be extracted from images. However, processing of the images is necessary to extract and use this information. One possible way of processing is pixelwise semantic segmentation, where each pixel of a given image is assigned to one predefined class, e.g. pedestrian, motor vehicle, drivable road, vegetation, etc.

For this use case, a custom dataset for training a classifier for semantic segmentation was recorded in the CARLA simulator, consisting of 8000 data samples, split into train (6k samples), test (1k samples) and validation (1k samples).

The first analysis for Semantic Segmentation data executed with the xpl[AI]ned framework already shows a significant imbalance in distribution of the different classes, e.g. 2.57 million “car” pixels vs. 0.55 million “pedestrian” pixels. While this is a common problem in semantic segmentation tasks and can be mitigated for example by using an adjusted loss function for unbalanced distributions during training, e.g. focal loss, the clustering of the different classes in certain areas of the image poses a problem, as can be seen in Figure 3.

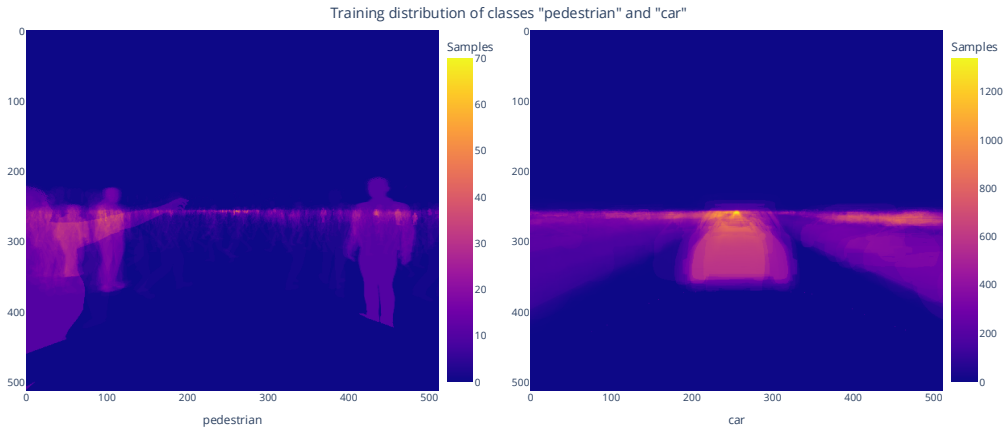


Figure 3: Distribution of classes in the training data

Training a simple segmentation network (approx. 18 million parameters) using a focal loss function, results in an overall accuracy score of 0.952. Despite that, analyzing precision (0.035) and recall (0.0002) for the class “pedestrian” shows that the overall score of the model is influenced by the imbalance in the data distribution, leading to misclassifications of “pedestrians”, as can be seen in Figure 4. However, the question, how exactly the decisions for the class “pedestrian” are influenced, remains.

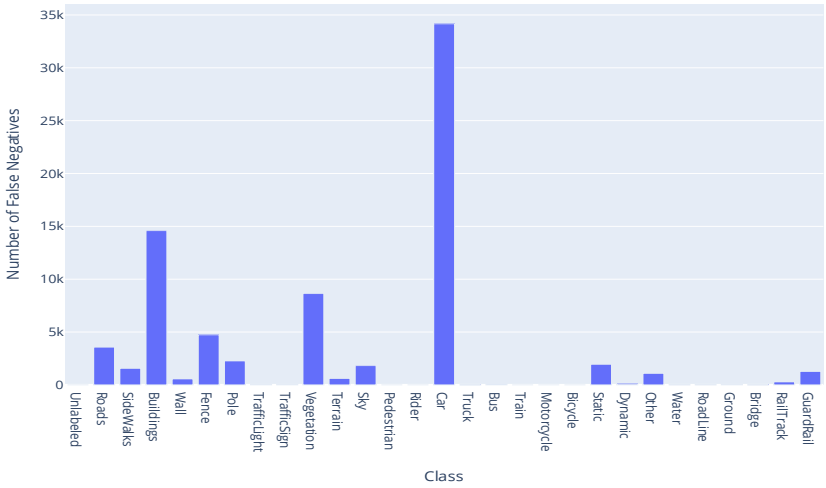


Figure 4: Class wise amount of "False Negatives" for class “pedestrian”

Xpl[AI]ned can automatically analyze and connect results of several metrics. Doing this for the confidence scores resulting from the probabilistic output layer of the network shows that the confidence for “pedestrians” is very low while the most likely class detected by the network for those pixels is “cars”. Comparing this information to the aforementioned data distribution, one can see that pedestrians are predominantly in locations with high proportions of cars or buildings.

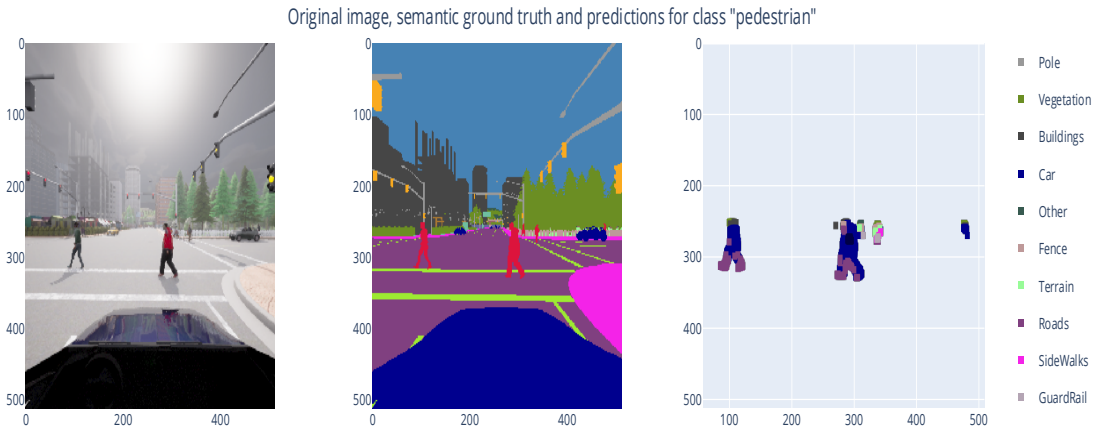


Figure 5: Misclassification of class "pedestrian"

This leads to the assumption that pedestrians based on the low share and location are misclassified as “cars”. To further confirm this assumption, the framework automatically selects datapoints which are outliers to the data distribution, as well as samples from the data distribution and analyzes the decisions of the model. As can be seen in Figure 5, the model classifies the pedestrian on the road as car while pedestrians in areas with low proportions of cars are not recognized as at all, but are being integrated into the background, thus becoming “invisible”.

Using this automatically gained information, necessary adjustments can be made to improve the network. By automatically analyzing the shortcomings and problems of the network, xpl[AI]ned can help to build your safety argumentation and improve the quality, reliability and trustworthiness of the MuT.

6 Conclusion and Summary

Being able to test AI systems during the complete lifecycle using a comprehensive framework can accelerate the development of safe ADAS/AD functions. While challenging, the understanding of the decision process in AI model is crucial. To achieve this understanding, a combination of different metrics from different stages of the AI lifecycle can help to understand boundaries and shortcomings of the model. With xpl[AI]ned, Keysight offers a comprehensive and easy to use solution for automating the test and validation during the AI development lifecycle, helping engineers to make AI driven ADAS/AD components safe, while saving time and effort.

Modulares Power-HIL-Konzept: Flexibles Testen und Validieren bei voller Leistung

Dr.-Ing. Manuel Fischer

dSPACE GmbH
RTDS-EED
Konrad-Zuse-Straße 98
71034 Böblingen
mafischer@dspace.de

Abstract: Das Validieren von Komponenten mit Power-HIL-Prüfständen hält mehr und mehr Einzug in bestehende Entwicklungsprozesse – nicht nur im Automobilsektor. Insbesondere die enorme Flexibilität, die es ermöglicht, das elektrische Umgebungs-verhalten des Prüflings per Software zu verändern, befördert Power-HIL-Prüfstände in den Status einer nennenswerten Alternative zu Prüfständen mit rotierenden Motoren. Der Beitrag stellt die wesentlichen Hauptkomponenten eines hochdynamischen Power-HIL-Prüfstands vor und leitet Anforderungen ab, welchen das leistungselektronische Stellglied genügen muss. Die Realisierung des Konzepts in Hardware wird aufgezeigt und abschließend mit Messungen belegt.

1 Einleitung

Die Elektromobilität ist einer der Megatrends in der Automobilbranche. Der Antriebsstrang eines elektrisch angetriebenen Autos besteht in der Regel aus einer Hochvoltbatterie, einem Traktionswechselrichter (WR) und einer elektrischen Maschine. Während des gesamten Entwicklungsprozesses spielen Testen und Validieren eine entscheidende Rolle, um neben Funktionsfähigkeit, Zuverlässigkeit und Sicherheit auch die Effizienz des Gesamtsystems nachzuweisen. Diese Hardware-Tests müssen in Übereinstimmung mit gängigen Normen, zum Beispiel der ISO 21498, durchgeführt werden.

Zu diesem Zweck wird üblicherweise ein Maschinenprüfstand verwendet (Abb. 1). Die untersuchte Maschine ist mit einer Lastmaschine gekuppelt, um unterschiedliche mechanische Betriebspunkte einzustellen. Der Prüflingswechselrichter (Device Under Test, DUT) wandelt die Gleichspannung der speisenden Batterie in die gewünschten Wechselspannungsformen um, um die untersuchte Maschine anzutreiben. Da diese Maschinenprüfstände unflexibel gegenüber variablen Maschinen- oder Batterieparametern sind, kann das DUT alternativ auch mit Hilfe eines Power-Hardware-in-the-Loop (PHIL)-Emulators getestet werden.

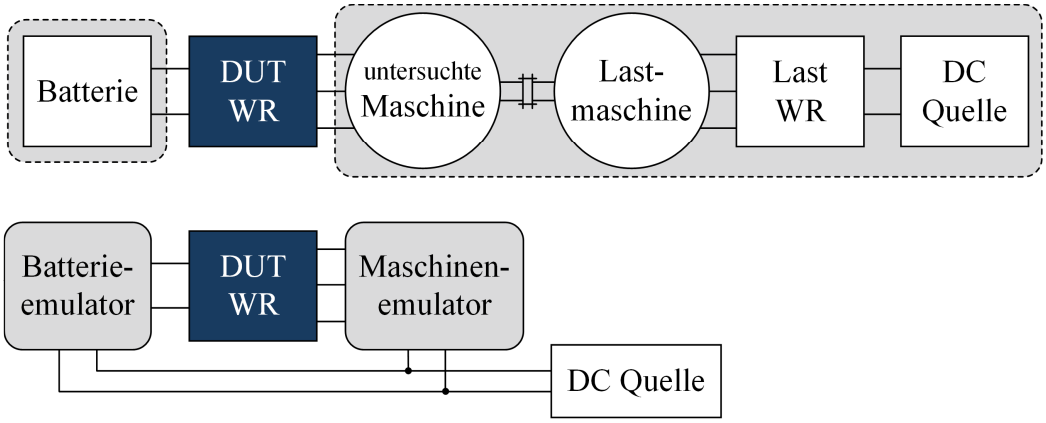


Abbildung 1: Schematische Darstellung eines herkömmlichen Maschinenprüfstands (oben) im Vergleich zu einem PHIL-Prüfstand (unten)

Dazu werden die Eingangs- und Ausgangsklemmen des DUT mit einem leistungselektronischen Stellglied (LESG) verbunden, welches dieselbe elektrische Charakteristik aufweist wie die betrachtete Batterie und die Maschine, siehe Abb. 1.

Der PHIL-Prüfstand besteht aus drei Hauptkomponenten:

- einem Batterie- und einem Maschinenmodell, die das elektrische Verhalten der Batterie und der Maschine in Abhängigkeit vom Schaltverhalten des DUT in Echtzeit berechnen,
- einer Regeleinheit, welche das LESG so ansteuert, dass die gleichen Lastströme wie im realen Prüfstand fließen,
- dem LESG selbst, das die gewünschten Lastströme einstellt.

In diesem Beitrag wird ein modularer Ansatz verfolgt, bei dem das LESG und die Stromregeleinheit jeweils auf einem Lastmodul vereint sind [MEY17, FIS22]. Jedes Lastmodul kann somit bidirektional als hochdynamische Stromquelle und -senke verwendet werden und genau eine Funktion elektrisch nachbilden, zum Beispiel einen Phasenanschluss einer Maschine oder einen einzelnen Batteriepol. Durch den modularen Ansatz ist man in der Anzahl emulierter Funktionen frei und nicht beispielsweise auf eine bestimmte Phasenanzahl der Maschine beschränkt. Der maximale Ausgangsstrom und damit die elektrische Maximalleistung einer Funktion kann über die Anzahl parallel geschalteter Lastmodule skaliert und je nach Testfall durch Veränderung der Modulanzahl variiert werden. Das elektrische Verhalten wird durch das Echtzeitmodell vorgegeben.

2 Modulares Power-HIL-Konzept

Abb. 2 zeigt die schematische Darstellung des gesamten PHIL-Systems. Das DUT ist an jedem Leistungsanschluss elektrisch an oben beschriebene Lastmodule angeschlossen, welche aus einem gemeinsamen Zwischenkreis V_{ZK} gespeist werden.

Die Lastmodule an den DC-Klemmen des DUT bilden das Verhalten der Batterie nach. Nach der Messung der Eingangsströme i_{Bat+} und i_{Bat-} berechnet das Batteriemodell den aktuellen Zustand und insbesondere die Ausgangsspannung $v_{Bat,soll}$ der emulierten Batterie in Echtzeit. Dieser Sollwert der Ausgangsspannung wird einem überlagerten Spannungsregler übergeben, der wiederum die Ströme $i_{Bat,soll}$ ermittelt, welche von den Lastmodulen gestellt werden müssen, damit die Spannung v_{Bat} über dem Kondensator C_{Bat} der gewünschten Batterieausgangsspannung $v_{Bat,soll}$ folgt.

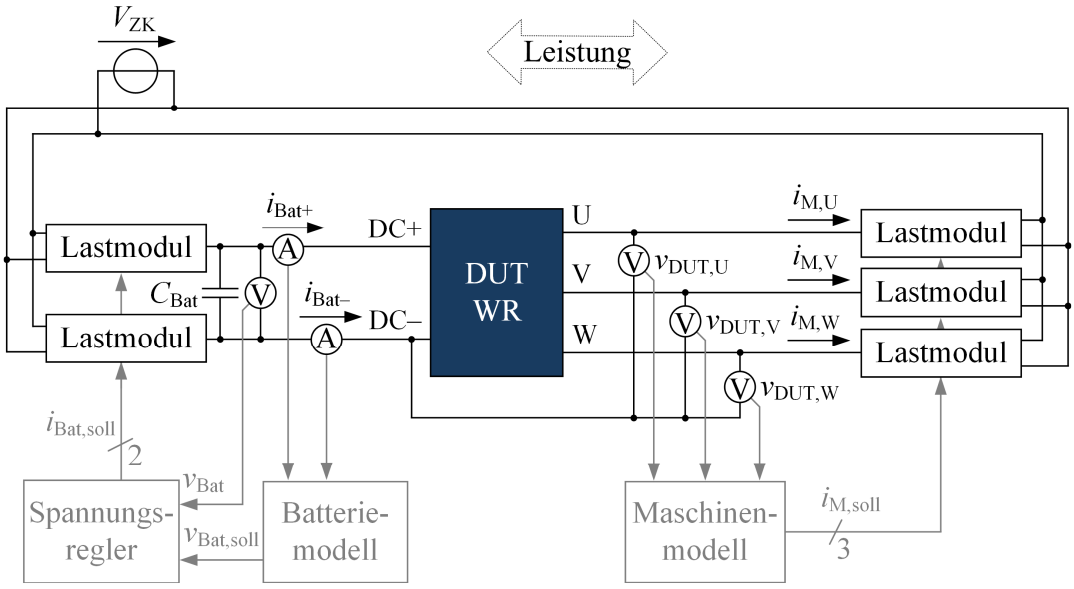


Abbildung 2: Übersichtsdiagramm des gesamten Power-HIL-Systems

Die Lastmodule an den AC-Klemmen des DUT emulieren das Verhalten der untersuchten Maschine. Nach der Messung der drei Ausgangsspannungen $v_{DUT,x}$ (mit x = U,V,W) berechnet das Maschinenmodell den aktuellen Zustand und insbesondere die Eingangsströme $i_{M,soll,x}$ der untersuchten Maschine und gibt diese als Sollwerte an die zugehörigen Lastmodule weiter. Darüber hinaus können zusätzliche Zustandsinformationen wie Kommunikation oder Lagesensoremulation auch über Signalschnittstellen ausgegeben werden.

Im motorischen Betrieb der virtuellen Maschine speisen die Lastmodule der Batterieemulation die eingangsseitige Leistung des DUT, die Lastmodule der Motoremulation speisen die Ausgangsleistung des Wechselrichters in den gemeinsamen Zwischenkreis zurück (und umgekehrt im generatorischen Betrieb). Folglich ergibt sich ein zirkulierender Leistungsfluss über den gemeinsamen Zwischenkreis, so dass die Gleichspannungsquelle V_{ZK} nur die Verlustleistung des Systems nachführen muss. Die hochdynamische Stromregelung der Lastmodule verhindert das Auftreten von ungewünschten Kreisströmen.

3 Hochvoltlastmodule

Die PHIL-Anwendung stellt mehrere Anforderungen an die Lastmodule: Spannungsfestigkeit für Batteriespannungen über 1000 V, hohe Ausgangsströme mit Bandbreiten von einigen kHz, Bidirektionalität hinsichtlich des Leistungsflusses, das Einbringen von möglichst wenig zusätzlichem Oberschwingungsgehalt und die Modularität, mehrere Module parallel betreiben und die Systemleistung je nach Anwendungsfall skalieren zu können. Zudem sollen die elektrischen Maximalwerte ohne Derating (Leistungshyperbel) einstellbar sein.

Bereits untersuchte Topologien sind Wechselrichter mit parallel geschalteten und versetzt angesteuerten Halbbrückenzweigen [NEM16], Wechselrichter mit Ausgangsfilter [FIS19], Linearverstärker [AMI19] oder speziell entwickelte Wechselrichterkonzepte wie Vierpunktwechselrichter mit veränderlichen Ausgangspotenzialen [FIS23].

Eine Topologie, die oben genannte Anforderungen vollständig erfüllt, ist der mehrzweigige NPC-Wechselrichter, siehe Abb. 3. Die zusätzlichen einstellbaren Ausgangsspannungen im Vergleich zu einem konventionellen Zweipunktwechselrichter verringern die harmonische Verzerrung der Ströme. Außerdem sperren in jedem Schaltzustand mindestens zwei Schalter die Zwischenkreisspannung V_{ZK}, wodurch jeder Schalter nur die Spannungsfestigkeit der halben Zwischenkreisspannung aufweisen muss. Die Schalter sind als schnell schaltende Siliziumkarbid-MOSFETs ausgeführt. Pro Modul sind drei NPC-Halbbrückenzweige parallel geschaltet. Dadurch kann der Ausgangsstrom dreimal so hoch sein wie der maximale Strom eines einzelnen Zweigs. Die drei Zweige sind jeweils paarweise über drei magnetisch gekoppelte Koppeldrosseln verknüpft, welche einer asymmetrischen Stromverteilung auf die drei Zweige entgegenwirken. Die Ausgangsspannung v_{Mod} kann am gemeinsamen Sammelpunkt je nach Schaltzustand der drei Zweige sieben diskrete Spannungszustände annehmen. Nach dem Sammelpunkt ist eine Längsdrossel L_C geschaltet, die als Regelstrecke zwischen dem DUT und den Lastmodulen wirkt.

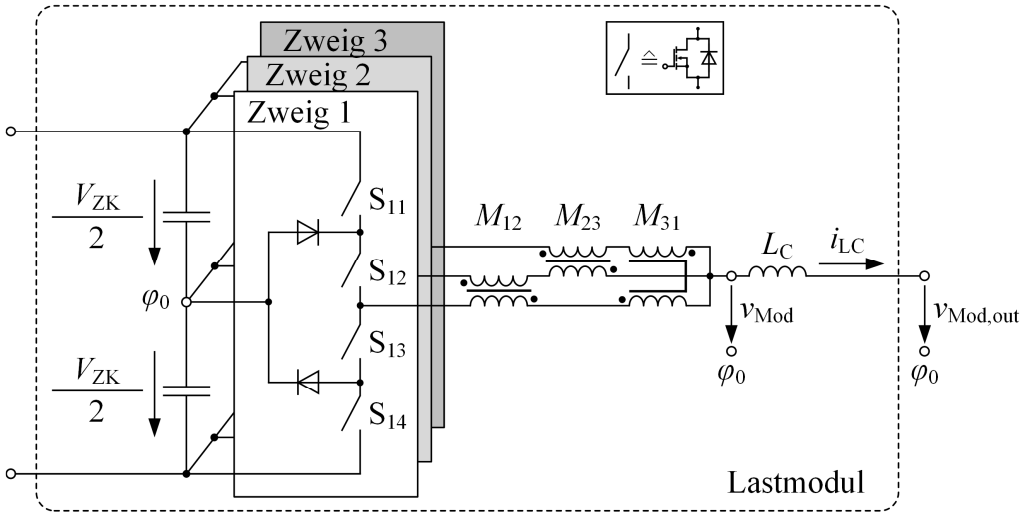


Abbildung 3: Schaltbild des LESG eines Lastmoduls

4 Echtzeitmodelle

Die Genauigkeit der Maschinen- und Batteriemodelle bestimmt die Qualität der Emulationsergebnisse. Eine notwendige Bedingung ist, dass alle Modelle in Echtzeit lauffähig sein müssen. In dieser Arbeit wird eine dreiphasige, permanentmagneterregte Synchronmaschine (PMSM) nach [BOE12, PLO13] simuliert. Diese Auswahl schränkt jedoch nicht die Nutzung anderer echtzeitfähiger Maschinenmodelle ein. Die Nachbildung anderer Maschinentypen ist genauso möglich wie die Emulation von Maschinen mit einer beliebigen Anzahl von Phasen oder weiteren Nichtlinearitäten.

Der elektrische Teil des Modells ist in Abb. 4 dargestellt. In Abhängigkeit der Ausgangsspannungen $v_{DUT,x}$ des DUT werden die Motorströme i_d und i_q im feldorientierten dq-System berechnet. Einflussparameter sind der Wicklungswiderstand R_S sowie die variablen Induktivitäten L_d und L_q und der Fluss des Permanentmagneten Ψ_{PM}, die jeweils nichtlinear abhängig von den Motorströmen i_d und i_q sind. Die Abhängigkeiten werden in 3D-Look-up-Tabellen hinterlegt.

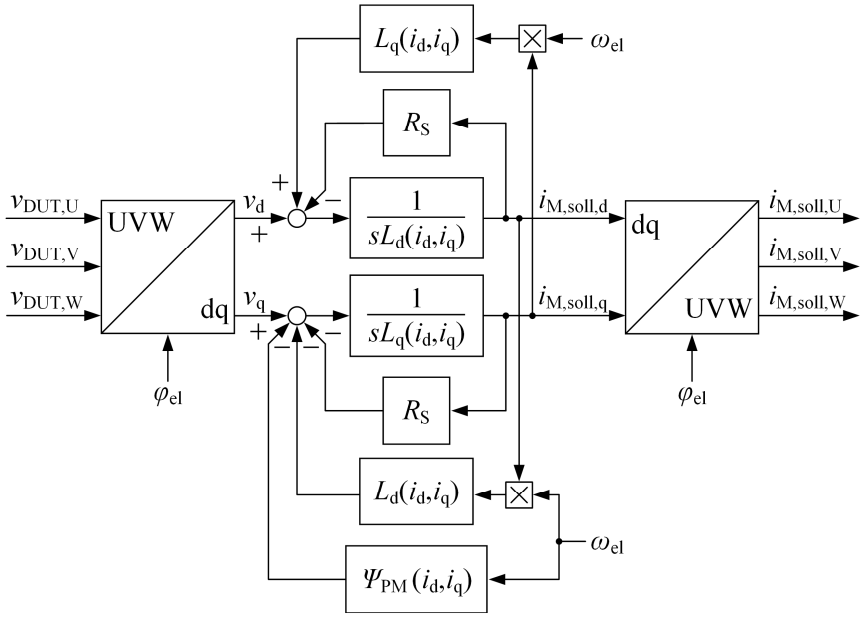


Abbildung 4: Blockdiagramm des nichtlinearen Maschinenmodells

Die Batterie wird mit dem in Abb. 5 dargestellten dynamischen Modell simuliert. V_{Zelle} stellt die innere Zellspannung dar, die direkt vom Ladezustand (State of Charge, SOC) abhängt. Außerdem berücksichtigt das Modell den

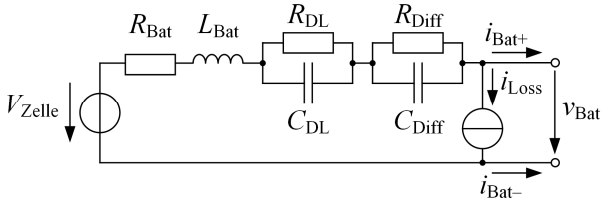


Abbildung 5: Schaltbild des dynamischen Batteriemodells

Widerstand R_{Bat} und die Induktivität L_{Bat} der Batterie, das dynamische Verhalten ihrer Diffusions- (R_{Diff}, C_{Diff}) und ihrer Doppelschicht (R_{DL}, C_{DL}) sowie zusätzliche Verluste (i_{Loss}). Genauere Erklärungen sind in [FIS22] zu finden.

5 Hardware-Aufbau und Messergebnisse

Das implementierte PHIL-System kann bei Spannungen über V_{ZK} = 1000 V betrieben werden. Abb. 6 zeigt den Hardware-Aufbau eines Lastmoduls. Jedes Lastmodul ist für Ströme bis zu 75 A_{RMS} ausgelegt und erreicht Anstiegsgeschwindigkeiten bis zu 5 A/µs. Werden höhere Ströme oder höhere Anstiegsgeschwindigkeiten des Stroms benötigt, kann eine beliebige Anzahl von Modulen parallel betrieben werden. Auf dem FPGA ist die Stromregelung implementiert.

Die Echtzeitmodelle sind auf einem leistungsfähigen FPGA-Board mit einer Abtastzeit von 8 ns und schnellen I/O-Schnittstellen implementiert. Von diesem Echtzeit-FPGA werden lediglich die Referenzströme an die dezentralen FPGAs auf den Hochvoltmodulen übertragen. Ein Echtzeitprozessor dient als Schnittstelle zwischen dem Echtzeit-FPGA und dem Leitrechner. Dadurch können Modellparameter auch während der Laufzeit geändert werden.

Für die Messergebnisse in diesem Beitrag wird der Aufbau aus Abb. 2 mit einem einzelnen Modul pro Batteriepol und pro Maschinenphase aufgebaut. Das DUT ist ein dreiphasiger Maschinenwechselrichter. Da der Schwerpunkt auf den Maschinenströmen liegt, wird die Batteriespannung auf einen konstanten Wert geregelt. Die Parameter des DUT und des PHIL-Systems sind in Tabelle I zusammengefasst.

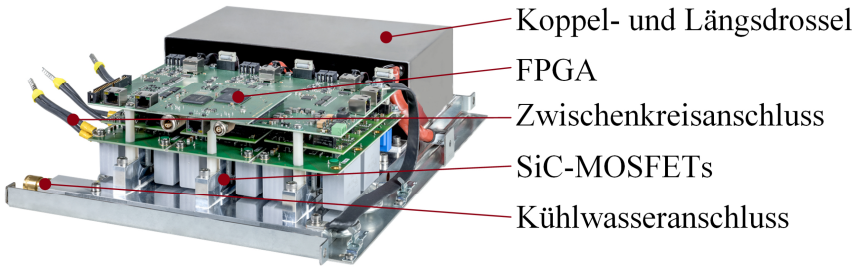


Abbildung 6: Hardware-Aufbau des Lastmoduls

Die virtuelle PMSM wird auf eine konstante Drehzahl von n = 1.000 U/min eingestellt. Der Prüflingswechselrichter arbeitet stromgeregelt. Seine Sollströme werden mit $i_{DUT,soll,d}$ = 0 A und $i_{DUT,ref,q}$ = 50 A vorgegeben. Dadurch ergibt sich ein nahezu konstantes virtuelles Drehmoment von M = 195 Nm.

Die Messergebnisse des Ausgangsstroms $i_{M,U}$ der Phase U des DUT und seines Sollwerts $i_{M,soll,U}$ aus dem Maschinenmodell sind in Abb. 7 dargestellt. Die Regelung des DUT führt zu einem sinusförmigen Stromverlauf mit einer Amplitude von 50 A wie gewünscht. Der reale Laststrom folgt seinen Sollwerten aus dem Maschinenmodell ohne sichtbare Abweichung. Abb. 8 zeigt eine detaillierte Betrachtung der Ströme. Aufgrund seiner hohen Bandbreite ist das PHIL-System in der Lage, den Sollstrom genau nachzubilden.

Tabelle I: Parameter des Hardware-Aufbaus

Parameter des Power-HIL	
Zwischenkreisspannung V_{ZK}	800 V
Taktfrequenz der Lastmodule	≤ 800 kHz
Parameter des DUT	
Batteriespannung v_{Bat} (konstant)	500 V
Taktfrequenz	5,2 kHz
Parameter des Maschinenmodells	
Maschinentyp	PMSM
Polpaarzahl z_p	3
Wicklungswiderstand R_S	0,01 Ω
Längsinduktivität L_d	2,7 mH
Querinduktivität L_q	2,7 mH
Fluss des Permanentmagneten Ψ_{PM}	0,87 Vs

Sogar die dreieckförmigen Stromrippel, die durch die taktende Arbeitsweise des DUT hervorgerufen werden, sind emuliert. Eine geringe zusätzliche Stromwelligkeit ist zu beobachten und resultiert aus dem Schaltverhalten der Lastmodule.

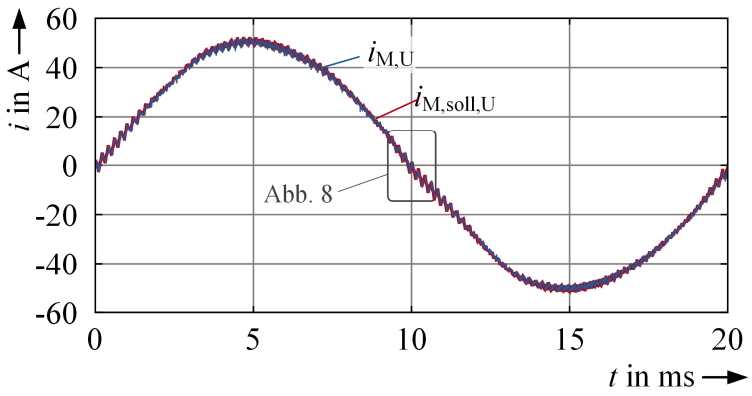


Abbildung 7: Messung des Phasenstroms

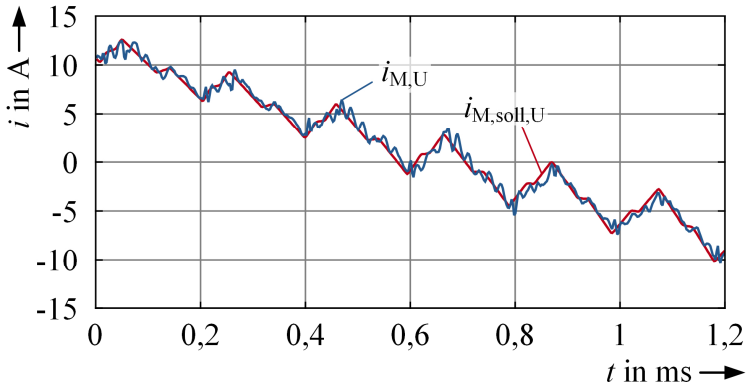


Abbildung 8: Detaillierte Betrachtung des Phasenstroms

6 Zusammenfassung

In diesem Beitrag wird das Konzept eines leistungsfähigen PHIL-Systems vorgestellt, das für Hochfrequenzanwendungen geeignet ist. Es wird der Aufbau eines Lastmoduls vorgestellt, das sich wie eine modulare Stromquelle mit hoher Bandbreite verhält. Jedes Lastmodul ist stromgeregelt, so dass mehrere Lastmodule parallel betrieben werden können, um die Gesamtleistung des Systems zu erhöhen. Die benötigten Sollwerte werden durch echtzeitfähige Maschinen- und Batteriemodelle berechnet. Das gesamte PHIL-System wird in Hardware aufgebaut. Seine Leistungsfähigkeit wird durch Messergebnisse nachgewiesen, die insbesondere die hohe Echtzeitfähigkeit und Emulationsgenauigkeit hervorheben.

7 Literaturverzeichnis

[AMI19] K. S. Amitkumar, et al. Linear Amplifier-Based Power-Hardware-in-the-Loop Emulation of a Variable Flux Machine, IEEE Transactions on Industry Applications, Vol. 55, Nr. 5, S. 4624–4632, 2019.

[BOE12] Matthias Boesing, et al. Modeling spatial harmonics and switching frequencies in PM synchronous machines and their electromagnetic forces, XXth International Conference on Electrical Machines, 2012.

[FIS19] Manuel Fischer, et al. Investigation on a Three-Phase Inverter with LC Output Filter for Machine Emulation, EPE Europe 2019.

[FIS22] Manuel Fischer, et al. High-Bandwidth Power Hardware-in-the-Loop for Motor and Battery Emulation at High Voltage Levels, EPE Europe 2022.

[FIS23] Manuel Fischer. Emulation elektrischer Maschinen mit exakter Nachbildung rippelbehafteter Phasenströme, Dissertation, ISBN: 978-3-8440-9113-7, 2023.

[MEY17] Gerrit Meyer. Enhanced Power Electronics System for High-Performance Testing of Motor Control Units in a Power HIL Environment, PCIM Asia 2017.

[NEM16] Christian Nemec. Leistungselektronische Nachbildung von Drehstrommaschinen mit variablem induktiven Verhalten, Dissertation, ISBN: 978-3-8440-4288-7, 2016.

[PLO13] Markus Plöger und Matthias Deter. Highly precise real-time simulation of E-motors, ATZ Elektronik, 2013.

Thermische Testsimulation der CCS-Dose für große Ladeströme zur Überprüfung von Optimierungsansätzen

Dipl.-Ing. Jochen Krings, Daimler Truck AG
Abteilung: Charging Components E-Mobility DTB
jochen.krings@daimlertruck.com

Daimler Truck AG
Fasanenweg 10
70771 Leinfelden-Echterdingen

Prof. Dr.-Ing. Hans-Christian Reuss, Universität Stuttgart / FKFS Stuttgart
Leitung Vorstandsbereich Fahrzeugmechatronik
info@fkfs.de

FKFS (Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren) Stuttgart
Pfaffenwaldring 12
70569 Stuttgart

Dipl.-Ing. Peter Ziegler, Daimler Truck AG
Teamleitung: Charging Components E-Mobility DTB
peter.z.ziegler@daimlertruck.com

B. Eng. Paul Steinmetz, Daimler Truck AG
Abteilung: Charging Components E-Mobility DTB
paul.ps.steinmetz@daimlertruck.com

Daimler Truck AG
Fasanenweg 10
70771 Leinfelden-Echterdingen

1 Zusammenfassung

Der CCS Ladestandard stellt in Europa die am weitesten verbreitete Variante zum Laden von Elektrofahrzeugen dar. Durch die Weiterentwicklung der Ladekomponenten können Ladeleistungen erzielt werden, die die Grenzen des CCS Standards erreichen.
Eine Möglichkeit zur Steigerung der Ladeleistungen ist, den Zeitpunkt bis zum Erreichen der maximal zulässigen Grenztemperatur an der Ladedose möglichst lange hinauszuzögern, bevor infolgedessen der Ladestrom reduziert werden muß, um einer unzulässigen Überhitzung entgegenzuwirken.
In dieser Arbeit werden mittels einer eigens entwickelten thermischen Matlab/Simscape Testsimulationssoftware, die Konstruktionsansätze zur Bauteiloptimierung an der CCS-Ladedose im Vorfeld getestet und die Ergebnisse validiert.

Abbildung 1: links: Mercedes Benz eActros 600
rechts: Detailansicht CCS-Ladedose mit Ladepistole
Quelle: Daimler Truck

Die Gegenüberstellungen der thermischen Testergebnisse von der aktuellen CCS-Ladedose und der Version mit Optimierungsansätzen, zeigen eine langsamere Erwärmung.
Durch den langsameren Anstieg der Ladedosentemperatur beim Laden mit hohen Strömen, kann dieser Strom über einen längeren Zeitraum gehalten werden, bevor die zulässige Grenztemperatur erreicht wird.
Die Testsimulationsresultate bestätigen somit die Ansätze der Konstruktionsoptimierung, die über einen längeren Zeitraum höhere Ladeströme ermöglicht. Die Anpassungen der Konstruktion können dazu beitragen die Ladezeit zu verringern.

2 Einleitung

Im Zuge der weltweiten Elektrifizierung des Automobilbereichs werden in den letzten Jahren auch vermehrt Nutzfahrzeuge mit elektrischen Antrieben entwickelt [1].
Im Vergleich zum Auto für den Individualverkehr, stellt im Nutzfahrzeugbereich insbesondere das schnelle Laden der großen Energiespeicher eine große Herausforderung dar [2]. Eine kurze Standzeit, durch schnelle Ladezeiten, ermöglicht eine schnellere Verfügbarkeit der Nutzfahrzeuge. Dies ist für die Wirtschaftlichkeit im Logistikbetrieb von großer Bedeutung [3].

Aufgrund der europaweiten Verbreitung von CCS (Combined Charging System) verwenden auch elektrifizierte Nutzfahrzeuge diesen Ladestandard.
Für eine schnelle Ladung der Energiespeicher über die CCS-Ladedose sind entsprechend hohe Ladeleistungen erforderlich, die mit hohen Ladeströmen einhergehen. Der Nutzfahrzeugbereich geht dabei an die Grenzen des CCS Normierungsstandards [4].

Einhergehend mit Ladeströmen von mehreren hundert Ampere und dem elektrischen Widerstand der Ladekomponenten entstehen auch Verluste in Form von Wärme. Zwecks Steigerung der Ladeeffizienz ist es prinzipiell erstrebenswert die gesamte Wärmeentwicklung möglichst gering zu halten. Neben der Einsparung von Energiekosten können auch fahrzeugseitig die Kühlkomponenten kleiner dimensioniert werden, was sich in einem leichteren Fahrzeuggewicht und damit in einem niedrigeren Energieverbrauch im Fahrbetrieb äußert [5].
Gleichzeitig führt ein geringerer und langsamerer Temperaturanstieg in den Ladekomponenten auch zu einer geringeren Materialspannung durch Temperaturdifferenzen, was die Langlebigkeit der Ladekomponenten begünstigt [6].
Ziel bei der Auslegung der Ladepfadkomponenten ist dabei immer, möglichst lange den maximal, zulässigen Ladestrom aufrecht zu halten und dabei den Temperaturgrenzwert von berührbaren Bauteilen, zum Schutz des Menschen, nicht zu überschreiten.

Um bereits frühzeitig in der Ladekomponentenentwicklung thermische Optimierungsansätze zu überprüfen, bieten sich thermische Simulationstests an. Durch das simulationsbasierte Testen der neuen Konstruktionsansätze zur Bauteiloptimierung können im Vorfeld die zu erwartenden Ergebnisse quantifiziert und im besten Fall Iterationsschleifen in der Bauteilentwicklung reduziert werden.

Im Folgenden wird ein thermischer Optimierungsansatz der CCS-Ladedose mittels Softwaresimulation getestet und die Ergebnisse vorgestellt.

3 Aufbau der CCS Ladedose

Der Aufbau einer CCS Ladedose ist in Abbildung 2 dargestellt.
Von der Ladestation über die Ladepistole kommend, wird der Strom über die Kontaktstifte ins Innere der Ladedose zur Verbindungsplatte geleitet. Diese ist über eine Schraubverbindung mit der Schiene für das Stromkabel verbunden. Die Temperaturüberwachung erfolgt separat für Plus und Minus über Sensoren, die sich jeweils zwischen den Kontaktstiften und den Verbindungsplatten befinden. Die Maximaltemperatur beträgt 90°C, laut DIN EN IEC 62196-1, [7].

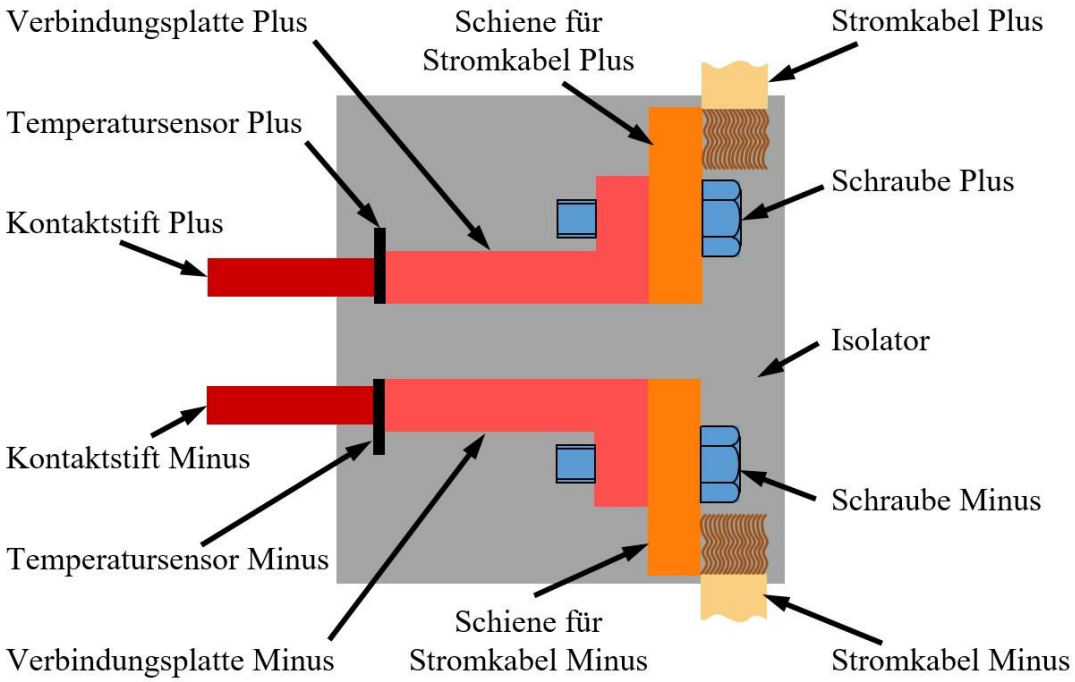


Abbildung 2: Schema innerer Aufbau CCS Ladedose
Quelle: Daimler Truck

4 Modellbildung und Simulationsaufbau

Aufgrund des elektrischen Widerstands von stromdurchflossenen Bauteilen kommt es zu inneren Verlusten, die zu einer Erwärmung führen [8]. Durch das Ohm'sche Gesetz Gl. (4.1), lassen sich mit der Spannung U, der Stromstärke I und dem elektrischen Widerstand $R_{elektrisch}$, die elektrische Leistung $P_{elektrisch}$ durch Gl. (4.2) beschrieben. Analog zur elektrischen Leistung läßt sich die Joule'sche Erwärmung $P_{thermisch}$, beziehungsweise der entstehende Wärmestrom $\dot{Q}$, durch Gl. (4.3) beschreiben:

Ohm'sche Gesetz: $$U = I * R_{elektrisch} \quad (4.1)$$

Elektrische Leistung: $$P_{elektrisch} = I * U = I^2 * R_{elektrisch} \quad (4.2)$$

Joule'sche Erwärmung: $$P_{thermisch} = \dot{Q} = I^2 * R_{thermisch} \quad (4.3)$$

Der thermische Widerstand $R_{thermisch}$ wird in Gl. (4.4) [9] durch die Länge l des Bauteils, die Wärmeleitfähigkeit λ als Materialeigenschaft und die Bauteilquerschnittsfläche A beschrieben:

Thermischer Widerstand: $$R_{thermisch} = \frac{l}{\lambda * A} \quad (4.4)$$

Anhand des „Technischen Leitfaden Thermosimulationsmodelle“ des ZVEI (Zentralverband Elektrotechnik- und Elektroindustrie) lassen sich die Kopplung von elektro-thermischen Simulationen wie folgt darstellen [10]:

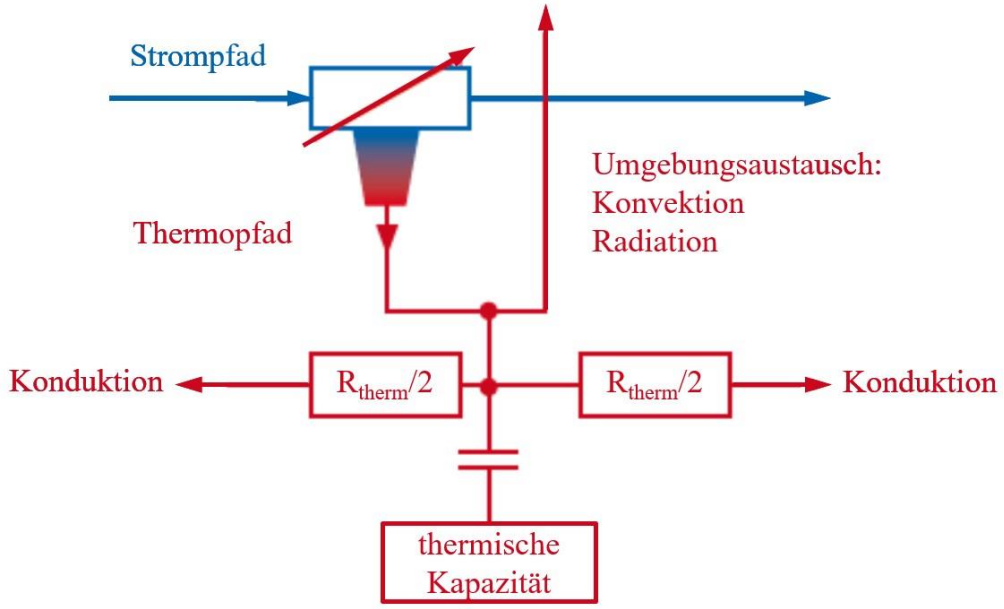


Abbildung 3: ZVEI Leitfaden Schema für elektro-thermische Kopplung
Quelle: ZVEI

Dabei wird ein elektrisches Bauteil mit seiner Wärmeproduktion als thermische Quelle interpretiert. Ein Teil der Wärme kann durch das Bauteil selber aufgenommen werden, die als thermische Kapazität bezeichnet wird. Des Weiteren kann Wärme beispielsweise mit der Umgebungsluft über Konvektion ausgetauscht oder über Wärmestrahlung durch Radiation abgestrahlt werden [11].
Die Wärmeleitung des Bauteils wird als Konduktion bezeichnet. Da in der Modellbildung der Wärmeeintrag in der Mitte des Bauteils angenommen wird, wird der thermische Widerstand der Konduktion jeweils hälftig auf beide Richtungen verteilt.

Für den Aufbau der thermischen Simulation in Matlab/Simscape können alle stromdurchflossenen Bauteile als Reihenschaltung von elektrischen und thermischen Widerständen, gemäß Abbildung 3, dargestellt werden. Die Bauteile stehen dabei in Wechselwirkung untereinander durch Konduktion, Konvektion und Radiation.

5 Konstruktion Verbesserungspotential: Erhöhung thermische Masse

Der Zielkonflikt des schnellen Ladens des Energiespeichers bedingt bei hohen Strömen eine erhöhte Wärmeentwicklung die mit der zweiten Potenz steigt, Gl. (4.3). Eine Möglichkeit um das Erreichen der limitierenden Grenztemperatur hinauszuzögern ist, die thermische Kapazität C_{th} der Verbindungsplatte und der Schiene für das Stromkabel zu erhöhen. Die thermische Kapazität C_{th} beschreibt die Fähigkeit eines Körpers thermische Energie zu speichern und wieder abzugeben. Sie steht dabei im proportionalen Verhältnis zu der Materialeigenschaft der spezifischen Wärmekapazität c und der Masse m des Bauteils, Gl. (5.1), [11]:

$$C_{th} = m * c \tag{5.1}$$

Die Multiplikation der thermischen Kapazität C_{th} mit der Temperaturänderung dT als Ableitung nach der Zeit dt, ergibt den Wärmestrom $\dot{Q}$ den der Körper aufnimmt oder abgibt, Gl. (5.2), [12]:

$$\dot{Q} = C_{th} * \frac{dT}{dt} = m * c * \frac{dT}{dt} \tag{5.2}$$

Die Masse der stromführenden Teile läßt sich leicht durch die Materialdicke erhöhen und erfordert in dem gegebenen Bauraum nur geringe Anpassungen, ohne die Außenkontur des Gehäuses zu beeinflussen.

Neben der Aufnahme von Energie in Form von Wärme, können die stromführenden Teile die Wärme auch innerhalb des Bauteils weiterleiten oder auch an andere Bauteile weitergeben. Dieser Effekt wird als Konduktion bezeichnet. Der Wärmestrom $\dot{Q}_{Konduktion}$ kann mit dem thermischen Widerstand $R_{thermisch}$ Gl. (4.4) und der Temperaturdifferenz ΔT bestimmt werden, Gl. (5.3), [12]:

$$\dot{Q}_{Konduktion} = \frac{1}{R_{thermisch}} * \Delta T = \frac{\lambda * A}{l} * \Delta T \tag{5.3}$$

Die Vergrößerung der Materialdicke der stromführenden Teile bewirkt eine proportionale Vergrößerung des Querschnitts A. Dadurch ergibt sich ebenfalls ein größerer Wärmestrom durch Konduktion, der die Abfuhr vom Wärme begünstigt.

6 Simulationsbasierter Test: Erhöhung thermische Masse

Für den simulationsbasierten Test wird die Materialdicke der Verbindungsplatten und die Schienen der Stromkabel um den Faktor 1,5 erhöht. Dadurch erhöht sich ebenfalls die Masse und der Bauteilquerschnitt um den Faktor 1,5. Absolut steigt die Masse der Bauteile dabei nur um wenige Gramm. Die Temperaturerwärmung der aktuellen Konstruktion als Basis wird mit den Ergebnissen aus der Simulation für die Variante mit der erhöhten thermischen Masse verglichen.
Als Randbedingungen werden eine Start- und Umgebungstemperatur von 25°C angenommen. Mit ausreichendem Abstand zur maximal zulässigen Temperatur von 90°C, wird eine Endtemperatur von 85°C an den Kontaktstiften festgelegt.

Der Ladestrom beträgt zu Beginn konstant 500A und wird nach Erreichen von 85°C solange reduziert bis sich ein konstanter Temperaturwert am Kontaktstift einstellt. Der eingestellt Kühlwert pro Kontaktstift beträgt 3,5W und wird über die Ladepistole der Ladestation abgeführt [13].

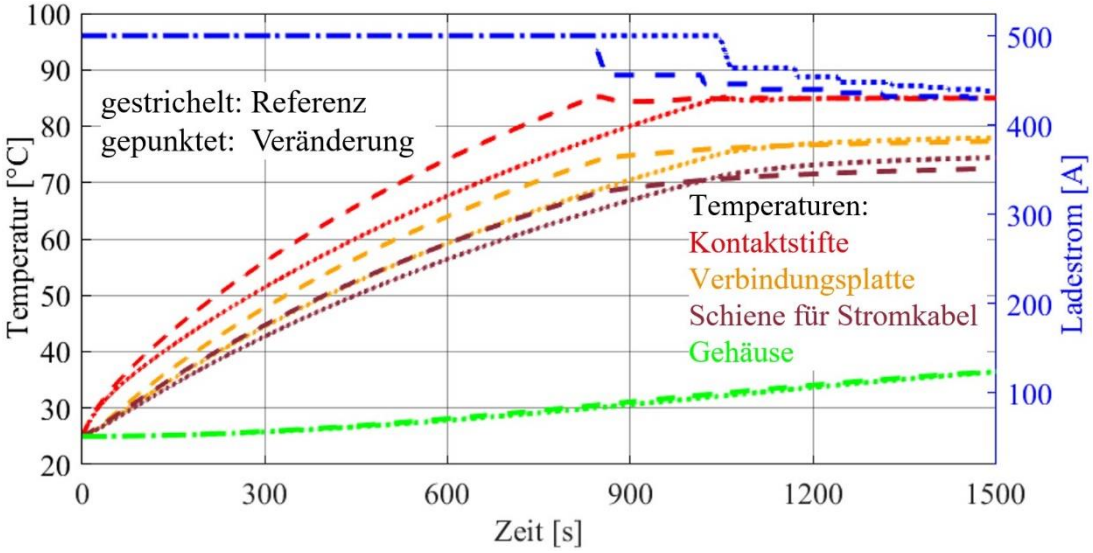


Abbildung 4: Vergleich von Basisversion mit Anpassung von Verbindungsplatte und Schiene für Stromkabel mit 1,5-facher thermischer Masse

7 Auswertung der Simulationstestergebnisse

Die Start- und Umgebungstemperatur beträgt, sowohl für die Basisversion als auch die angepaßte Version, 25°C. Bei einem konstanten Ladestrom von 500A, blaue Graphen, erreicht der Kontaktstift der Basisversion, rot gestrichelter Graph, innerhalb von 837s die für die Simulation festgelegte Endtemperatur von 85°C. In der angepaßten Version, rot gepunkteter Graph, wird die Endtemperatur von 85°C erst 209s später erreicht. Dies bedeutet, daß die Ladezeit bei 500A um 25% länger gehalten werden kann.

Zurückzuführen ist dies auf die Erhöhung der thermischen Masse von der Verbindungsplatte und der Schiene für das Stromkabel. Nach Gl. (5.1) wird durch die Erhöhung der Massen die thermische Kapazität vergrößert, die einen höheren Wärmestrom $\dot{Q}$ ermöglicht Gl. (5.2) und damit als größere Wärmesenke dient. Durch die Fähigkeit mehr Wärme aufnehmen zu können, ergibt sich sowohl für die Verbindungsplatte, oranger Graph, und die Schiene für das Stromkabel, violetter Graph, eine langsamere Erwärmung. Die Reduktion des thermischen Widerstands $R_{thermisch}$ vergrößert dabei auch den konduktiven Wärmestrom $\dot{Q}_{Konduktion}$ Gl. (5.3). Dadurch kann die Wärme an die benachbarten Bauteile und das Gehäuse schneller abgeleitet werden.

Um einer Überschreitung der Endtemperatur entgegenzuwirken, muß der Ladestrom, nach Erreichen einer Temperatur am Kontaktstift von 85°C, reduziert werden. In der Basisversion wird der Ladestrom für 170s zunächst auf 456A reduziert, bis sich nach einer weiteren Reduktion nach 103s auf 440A eine konstante Temperatur am Kontaktstift von 85°C einstellt.

In der angepaßten Version wird der Ladestrom zunächst für 123s auf 464A reduziert. Nach der anschließenden Reduktion auf 448A nach 71s stellt sich am Kontaktstift ebenfalls eine konstante Temperatur von 85°C ein. Damit liegt der dauerhafte Ladestrom um 8A höher als bei der Basisversion. Insgesamt ergibt sich dadurch eine größere elektrische Energiemenge die in das System eingeführt wird, im Vergleich zu der Basisversion. Dies ist gleichbedeutend mit einem größeren Wärmstrom laut Gl. (4.3). Das Resultat ist in den zwei Graphen der Verbindungsplatte, orange, und der Stromschiene für das Kabel, violett, zu erkennen. Beide gepunkteten Linien liegen über dem Temperaturniveau der jeweiligen Basisversion, gestrichelte Graphen.

Ein ähnlicher Verlauf ist auch an der Erwärmung des Gehäuses zu erkennen, auch wenn der Temperaturunterschied nur schwach ausgeprägt ist. Der gesteigerte elektrische Energieeintrag äußert sich auch hier über die erhöhte Abwärme in einer höheren Gehäusetemperatur. Da die Masse und die ableitende Oberfläche des Gehäuses im Vergleich zu den stromführenden Teilen groß sind, lassen sich die geringen Temperaturunterschiede erklären.

8 Fazit

Durch das thermisch-simulationsbasierte Testen, des neuen Konstruktionsansatzes zur Erhöhung der thermischen Masse von stromführenden Leitern in der CCS Ladedose, konnte im Vorfeld die zu erwartenden Ergebnisse quantifiziert werden.

Der Vergleich zwischen der Ausgangsversion und des Konstruktionsansatzes, zeigte durch die höhere, thermische Kapazität, eine langsamere Erwärmung. Dadurch konnte der konstante Ladestrom von 500A um 209s, beziehungsweise 25%, länger aufrecht gehalten werden, bis die definierte Grenztemperatur von 85°C erreicht wurde. Dieses Ergebnis ist gleichbedeutend mit einer größeren, geladenen Energiemenge in gleicher Zeit. Da die Anpassungen an den Stromleitern trotz der Vergrößerung um den Faktor 1,5, absolut gesehen nur wenige Gramm mehr Material bedeuten, fallen die Änderungen insgesamt gering aus. Somit läßt sich dieser Konstruktionsansatz in dem vorhandenen Gehäuse umsetzen.

Durch den modularen Aufbau der thermischen Testsimulation, besteht die Möglichkeit weitere Vergleiche und Untersuchungen von neuen Lösungsansätzen, zur thermischen Optimierung der CCS Ladedose im Speziellen oder vergleichbaren Bauteilen aus dem Ladepfad im Allgemeinen, vorzunehmen.

Die Testsimulation kann somit dazu betragen die Anzahl der Iterationsschleifen von Konstruktionsansätzen für die Bauteiloptimierung zu verringern und dadurch Entwicklungszeit und Kosten zu reduzieren.

9 Literaturverzeichnis

[1] Infineon Technologies AG, Neubiberg, Deutschland, https://www.infineon.com/cms/en/discoveries/electrified-commercial-vehicles/, online 09.09.2024

[2] 0Daimler Truck AG, Leinfelden-Echterdingen, Deutschland, https://hub.mercedes-benz-trucks.com/de/de/trucks/eactros-600.html, online 09.09.2024

[3] S. Schulze, “Ist Megawatt Laden die Zukunft? Potentiale, Herausforderungen und Risiken”, Daimler Truck AG, tme (Lehrstuhl für Thermodynamik mobiler Energiewandlungssysteme), RWTH Aachen University, Aachen, März 2024

[4] B. Horrmeyer, DKE (Deutsche Kommission Elektrotechnik Elektronik Informationstechnik), https://www.dke.de/de/arbeitsfelder/mobility/news/megawatt-charging-system-elektrifizierung-schwerlastverkehr, online 09.09.2024

[5] R. Collin, Y. Miao, A. Yokochi, P. Enjeti, A. v. Jouanne, “Advanced Electric Vehicle Fast-Charging Technologies“, MDPI: Energies, Mai 2019

[6] M. Thoben, F. Sauerland, K. Mainka, S. Edenharter, L. Beaurenaut, “Lifetime modeling and simulation of power modules for hybrid electrical/electrical vehicles”, Volume 54, Issues 9–10, 2014, Seiten 1806-1812, Science Direct, London, ISSN 0026-2714, https://doi.org/10.1016/j.microrel.2014.07.009, 2014

[7] DIN e.V., DIN EN IEC 62196-1:2022, Deutsche Fassung: Stecker, Steckdosen, Fahrzeugkupplungen und Fahrzeugstecker - Konduktives Laden von Elektrofahrzeugen - Teil 1: Allgemeine Anforderungen, Berlin, 2022

[8] M. Vidmar, Lectures on the scientific fundamentals of electrical engineering, University Ljubljana, Slowenien, Print ISBN 978-3-642-52572-8, 1928

[9] C. Gerthsen, D. Meschede, Gerthsen Physik, Die ganze Physik zum 21. Jahrhundert, Springer, Berlin, ISBN 978-3-540-02622-8, 2004

[10] ZVEI, Zentralverband Elektrotechnik- und Elektroindustrie, Technischer Leitfaden Thermosimulationsmodelle Version 1.1, Köln, 2020

[11] J.P. Huang, Transformation Thermotics for Thermal Conduction, Convection and Radiation, Springer, Singapore, Online ISBN 978-981-15-2301-4, Januar 2020

[12] C. Borgnakke, R. E. Sonntag, Fundamentals of Thermodynamics, Weinheim, Wiley, 10th Edition, ISBN: 978-1-119-72365-3, Juli 2020

[13] P. Steinmetz, Thermische Simulation einer CCS (Combined Charging System) Ladedose für Schwerlastkraftwagen, Hochschule Bonn-Rhein-Sieg, unveröffentlicht, April 2024

Importance of Interoperability Testing in Smart and Bidirectional Charging Systems

Jan Roßa, Yannik Mensing

P3 Energy Lab
P3 automotive GmbH
Heilbronner Str. 86
70191 Stuttgart
jan.rossa@p3-group.com
yannik.mensing@p3-group.com

Abstract: Interoperability is crucial for integrating EVs and EVSEs into the energy ecosystem, ensuring seamless communication among energy management and grid requirements. In a smart and bidirectional charging context, where EVs not only consume electricity but also have the capability to feed power back into the grid, interoperability becomes even more crucial. Standardized protocols like ISO15118 and OCPP enable efficient charging communication and energy exchange. Interoperability enhances consumer convenience, and grid stability by facilitating dynamic power flow management and supporting distributed energy resources. Prioritizing interoperability, P3's cutting-edge “Energy.Lab” offers automated testing solutions to unlock the full potential of smart and bidirectional charging technologies throughout energy innovations. Core purpose of this paper is to share methodologies and best practices for testing and validation of smart and bidirectional charging functionalities - ranging from application of automated test tools over use-case specific test concepts to standards in configuration and error codes.

1 Introduction

Interoperability is a cornerstone in the integration of electric vehicles (EVs) and electric vehicle supply equipment (EVSE) into the wider energy ecosystem. This concept refers to the ability of different systems and devices to communicate and function with each other without restrictions, regardless of the manufacturer or technology used. As electric vehicles become more widespread around the world, the need for a seamless and connected charging infrastructure is becoming increasingly important.

At the heart of interoperability is facilitating communication between EVs, EVSEs and energy management systems. In a world where there are multiple EV models from different manufacturers and different charging station providers, interoperability ensures that an EV can be charged at any compatible station. This is important not only for the convenience of consumers, but also for the efficiency and reliability of the entire energy system. Without interoperability, the charging experience could become fragmented, with compatibility issues leading to user frustration and potential barriers to wider EV adoption.

Beyond consumer convenience, interoperability is essential for optimizing energy management within smart grids. Smart grids rely on the dynamic flow of information to manage electricity demand and supply effectively. Interoperable systems enable real-time data exchange between EVs, charging stations, and grid operators, allowing for intelligent scheduling of charging sessions, load balancing, and demand response. This is particularly crucial during peak demand periods when the grid is under stress, as well as during times when renewable energy generation, such as solar or wind power, fluctuates.

Moreover, interoperability is a key enabler of advanced technologies like Vehicle-to-Grid (V2G). V2G technology allows EVs to not only draw power from the grid but also to return electricity back to it, acting as mobile energy storage units. This bidirectional flow of energy is instrumental in stabilizing the grid, especially as the penetration of renewable energy sources increases. For V2G to function effectively, a high degree of interoperability is required, ensuring that vehicles, charging infrastructure, and grid systems can all communicate seamlessly.

The importance of interoperability is further underscored by the development of standardized protocols such as ISO 15118, OCPP (Open Charge Point Protocol), and EEBUS. These standards have been designed to promote uniformity across the EV ecosystem, ensuring that different systems can operate together smoothly. ISO 15118, for example, facilitates communication between EVs and charging stations, enabling functionalities like Plug & Charge, which allows for automatic identification and billing of the vehicle without the need for user interaction. OCPP, on the other hand, is widely adopted for managing communications between charging stations and central systems, providing a standardized interface for interoperability across different manufacturers. EEBUS is focused on ensuring interoperability within the broader energy management context, allowing various energy devices, including EVs, to communicate with each other effectively.

Regulators and industry organizations recognize the importance of interoperability in achieving a scalable and efficient EV infrastructure. By adhering to these standards, manufacturers, energy providers, and other stakeholders can ensure that the systems they develop are compatible with each other, reducing the risk of fragmentation in the market. This alignment not only supports the current integration of EVs into the energy ecosystem but also lays the groundwork for future advancements in smart grid technologies and the broader energy transition.

2 Smart and Bidirectional Charging

Smart and bidirectional charging are transformative developments in the field of electric mobility, fundamentally altering how Electric Vehicles (EVs) interact with the energy grid. Traditional EV charging involves a one-way flow of electricity, where power is drawn from the grid to charge the vehicle’s battery. However, bidirectional charging introduces a new dynamic, allowing EVs not only to consume electricity but also to discharge stored energy back into the grid. This bidirectional flow of energy is particularly valuable in the context of modern energy systems, where the integration of renewable energy sources, such as solar and wind, introduces variability and complexity into grid management.

In a bidirectional charging system, EVs can function as mobile energy storage units, contributing to the grid's stability and resilience. During periods of high renewable energy generation, such as midday when solar power is abundant, EVs can be charged using excess renewable energy. Conversely, during peak demand periods or when renewable generation is low, these vehicles can return stored energy to the grid, helping to balance supply and demand. This capability supports the broader adoption of renewable energy by mitigating some of the challenges associated with its variability.

Smart charging complements bidirectional charging by adding layers of intelligence and automation to the charging process. Smart charging systems are designed to optimize charging schedules based on various factors, such as electricity prices, grid demand, and the availability of renewable energy. These systems can prioritize charging during off-peak hours when electricity is cheaper and the grid is under less strain, or they can delay charging to align with periods of high renewable energy production. This not only reduces the cost of charging for consumers but also helps to flatten demand peaks, easing the burden on the grid.

Moreover, smart charging systems can communicate with grid operators to provide demand response services, where charging is adjusted in real-time to respond to grid conditions. For example, if the grid is experiencing high demand, smart charging systems can temporarily reduce or pause charging to alleviate pressure on the system. In return, EV owners may be compensated for their participation in these demand response programs, creating an incentive for them to contribute to grid stability.

The effective implementation of smart and bidirectional charging hinges on the interoperability of the systems involved. Interoperability ensures that EVs, charging stations, energy management systems, and grid operators can communicate and coordinate seamlessly, regardless of the manufacturer or technology used. Standardized communication protocols, such as ISO 15118, OCPP and EEBUS, are essential in this context, as they provide the necessary framework for these interactions.

Furthermore, the potential of bidirectional charging extends beyond supporting the grid. It can also play a role in home energy management systems, where EVs can be used to power homes during outages or peak pricing periods, reducing reliance on the grid and lowering energy costs. This capability, known as Vehicle-to-Home (V2H) or Vehicle-to-Building (V2B), highlights the versatility of bidirectional charging technologies and their potential to transform not only transportation but also energy consumption patterns.

In conclusion, smart and bidirectional charging represent a paradigm shift in how electric vehicles interact with the energy grid, offering benefits for grid stability, renewable energy integration, and consumer cost savings. Their success, however, is contingent on the development and widespread adoption of interoperable systems and standardized protocols, which are crucial for enabling the seamless operation of these advanced charging technologies.

3 Standardized Protocols for Interoperability

There are many protocols and standards for smart charging and grid integration of electric vehicles that are already in use or under development. Figure 1 shows the most important standards and how they relate to each other. The following subsections take a closer look at the standards presented and explain the special features of each. [FfE23]

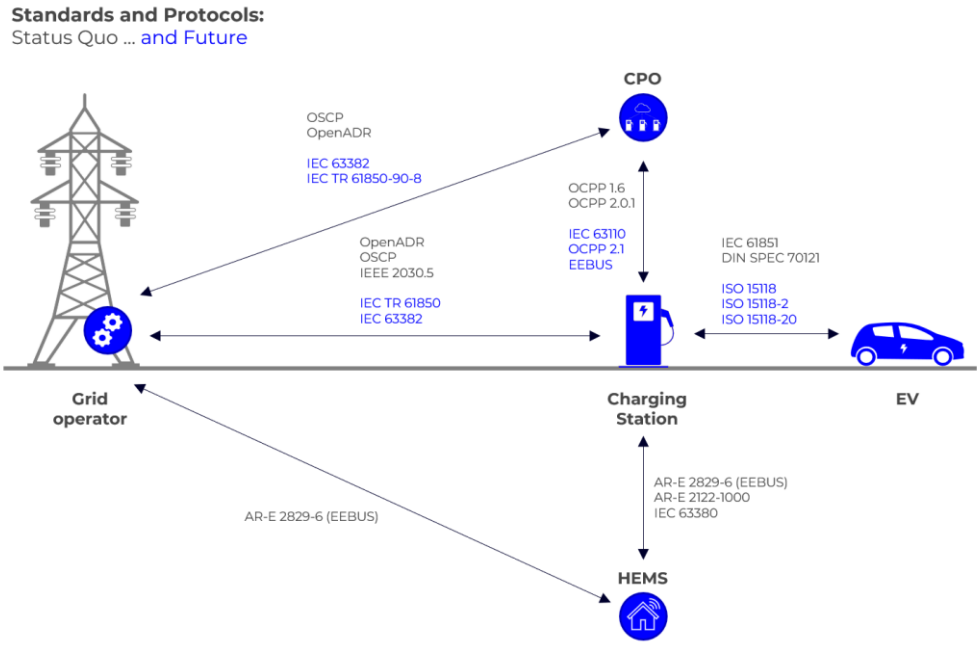


Figure 1: Overview of the relevant communication interfaces and charging infrastructure standards for electromobility.

3.1 Communication between EV and Charging Station

The IEC 61851 standard is currently supported by all electric vehicles in Europe. In this low-level communication standard, signals are transmitted via pulse width modulation. It only offers basic charging functions and does not enable any advanced authentication and authorization functions. IEC 61851 does not enable digital data transmission that goes beyond simple control signals. It is expected in the industry that IEC 61851 will only remain relevant as a fallback solution in the future. [FfE23]

ISO 15118 is a high-level communication standard for the communication interface between electric vehicles (EVs) and charging infrastructure. This protocol encompasses several key features that enhance the user experience and operational efficiency. One of the most notable features is Plug & Charge, which automates the authentication and billing process by allowing the EV to automatically identify itself and initiate charging upon connection to a compatible charging station. This eliminates the need for manual intervention, making the charging process seamless and user-friendly. Additionally, in 2022, the responsible committees released the successor to ISO 15118-2, ISO 15118-20. The standard supplements the existing standard with additional functionalities that will be essential for electromobility in the future. These include bidirectional power transfer (BPT), which enables the feeding of energy back into the power grid. This capability allows EVs to not only draw power from the grid but also return excess stored energy back to it, thus contributing to grid stability and facilitating the integration of renewable energy sources. By enabling these advanced functionalities, ISO 15118 plays a crucial role in promoting interoperability, enhancing the efficiency of energy use, and supporting the development of smart grids. At present, market acceptance is still very low, but as ISO 15118 is supported by several major car manufacturers, this will change in the future. [ISOa] [ISOb]

3.2 Communication between Charging Station and CPO

The Open Charge Point Protocol (OCPP) from the Open Charge Alliance is currently the de facto standard for communication between charging stations and CPOs. It offers compatibility with some other charging infrastructure protocols, such as OpenADR. OCPP is mainly used for authentication and authorization and offers application possibilities for smart charging and demand response. The latest OCPP version 2.0.1, which was released in 2020, supports ISO 15118-2 and thus enables functionalities such as Plug & Charge. Version 2.0.1 also offers increased safety requirements that go beyond version 1.6, which currently prevails on the European market. Bidirectional charging is not yet supported in version 2.0.1. A successor version currently under development will make OCPP compatible with ISO 15118-20 and support bidirectional charging. [OCPP]

OCPP is currently widely used and accepted internationally. However, there are also manufacturers of home energy management systems (HEMS) that rely on the EEBUS communication interface for communication between electric vehicles and charging devices. EEBUS is a communication protocol designed to ensure interoperability within the broader energy management landscape. It focuses on enabling seamless data exchange and coordination between various energy-consuming and energy-generating devices, including EVs, home energy management systems, and the electrical grid. EEBUS facilitates the integration of EVs into smart grids by allowing these vehicles to interact with other energy devices and systems, optimizing energy usage and supporting demand response initiatives. For instance, an EV connected to a home energy management system via EEBUS can adjust its charging times based on the availability of renewable energy, such as solar power, or according to the overall energy consumption patterns of the household. This level of integration and coordination helps in balancing energy loads, reducing peak demand, and enhancing the efficiency and sustainability of energy use. Currently EEBUS is not widespread on the market, but it offers great potential for bidirectional charging. [EEBUS22]

IEC 63110 is a standard published in 2023 that was developed as an alternative to OCPP. It covers the definitions, use cases and architecture for the management of charging and discharging infrastructures for electric vehicles. IEC 63110 addresses the general requirements for building an electric mobility ecosystem and therefore covers the communication flows between different electric mobility stakeholders as well as the data flow with the power grid. It offers some advantages over the OCPP, but market penetration is still very low. In addition, the OCA and the IEC recently entered into a cooperation agreement under which OCPP version 2.0.1 is to be published as IEC 63584. It is therefore unlikely that IEC 63110 will continue to prevail. [IECa] [IECb]

3.3 Communication with a Home Energy Management System (HEMS)

The VDE application rule AR-E 2122-1000 has been in place since 2021 for communication between electric vehicles and the local energy management system downstream of a grid connection point, taking grid-side requirements into account. This application rule serves as the basis for the IEC 63380 standard, which was published in January 2024. [IECc]

Application rule AR-E 2829-6 is a communication protocol for the interface to the electrical low-voltage distribution grid at the subscriber's property boundary. It is compatible with AR-E 2122-1000 for communication between the local energy management system and the charging infrastructure. AR-E 2829-6-1 to 2829-6-4 were published in 2021/2022. AR-E 2122-1000 and AR-E 2829-6 are based on the specifications of the EEBUS communication interface. Compatibility of AR-E 2829-6 with OpenADR is being sought.

3.4 Communication with the electricity grid

The OSCP protocol from the Open Charge Alliance has been used for communication between CPOs and the power grid for several years. Version 1.0 was released in 2015 and version 2.0 in 2020. OSCP is not yet widely used. It is used by the distribution system operator to transmit the physical grid capacity to the CPO. [OSCP]

There is also the OpenADR protocol for communication with the electricity grid, which is used for communication for demand response (DR) and distributed energy resources (DER) applications. It is currently used primarily for peak load management, and other DR and DER applications are being tested. OpenADR has been officially accredited as IEC 62746-10-1 ED1 since 2018. [IECd]

Another protocol for grid communication is IEEE 2030.5, also known as Smart Energy Profile 2.0 (SEP 2.0), it is a communication protocol that was developed for interoperability in smart energy grids. It enables communication between different devices in the smart grid, such as energy meters, electric vehicles, chargers and household appliances. IEEE 2030.5 supports bidirectional communication and makes it possible to exchange data on energy consumption, load control, pricing and other relevant information in real time. [IEEE]

Legislation on communication between charging stations and the power grid is still under development and standardization efforts are still in the early stages. The industry expects that the IEC 61850 series of standards, whose application for station automation is established in energy and telecontrol technology, will play a role. IEC 61850-7-420 describes communication for DR and DER applications. The IEC TR 61850-90-8 technical report builds on this and presents an object model for electromobility. The IEC 61850 series is a widely used standard that is constantly being further developed. [IECf]

Over the next few years, the IEC 63382 standard for the grid integration of electric vehicles is to be developed at international level. The JWG15 working group was founded for this purpose in 2021 and publication is planned for 2025. [IECg]

3.5 Key findings

As the previous chapters illustrate, there is a wide range of protocols and standards for the interoperability of components within the overall system of electromobility. However, these standards are only helpful for interoperability if they reflect a broad consensus among the involved stakeholders and, therefore, gain widespread acceptance. Furthermore, the existence of numerous different and competing standards makes it more difficult to achieve and maintain broad interoperability across all manufacturers and system levels.

Particularly in the field of bidirectional charging, the standards currently relevant, according to P3 expertise, are those depicted in Figure 2. At the forefront is OCPP, as this protocol has already gained wide acceptance in the industry and has become the de facto standard. Furthermore, its scope is continuously expanding through ongoing development. IEC 63110, developed to replace OCPP, fundamentally offers all the functions needed for the future. However, this standard has so far seen little adoption in the industry. Additionally, the OCA and IEC are currently working together on a new standard, IEC 63584. From P3's perspective, IEC 63110 will therefore not become established in the future.

For communication with the electrical grid, the relevant standards are primarily OpenADR, IEEE 2030.5, and EEBus, which will compete in the future. It is currently unclear which of these standards will prevail or whether they will coexist. This underscores the importance of extensive interoperability testing, as the coexistence of different standards in particular makes development especially complex.

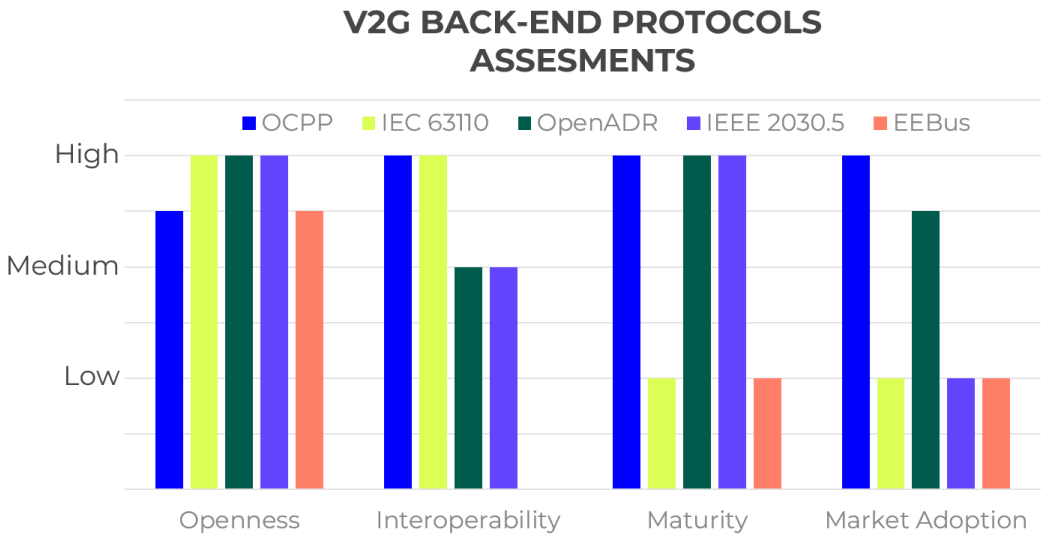


Figure 2: An assessment of the main EV back-end protocols for V2G.

4 Benefits of Interoperability

Interoperability in the electric vehicle (EV) ecosystem offers a range of significant benefits that extend across consumer experience, grid stability, and the integration of renewable energy sources. By ensuring seamless communication and compatibility between various EVs, charging stations, and energy management systems, interoperability enhances the overall convenience and reliability of EV charging. For consumers, this means the ability to charge their vehicles at any compatible station without worrying about compatibility issues, thereby simplifying the user experience and encouraging the adoption of electric mobility.

From an energy management perspective, interoperability plays a critical role in optimizing grid operations. It facilitates dynamic load balancing, allowing the grid to manage energy demand more efficiently, especially during peak usage times. This capability is essential for maintaining grid stability, as it helps prevent overloading and ensures a consistent supply of electricity. Furthermore, interoperability supports advanced functionalities such as Vehicle-to-Grid (V2G) technology, where EVs can return stored energy to the grid during high demand periods, thus acting as mobile energy storage units and enhancing grid resilience.

Interoperability also promotes the integration of renewable energy sources into the grid. By enabling EVs to communicate effectively with renewable energy systems, it allows for the synchronization of charging activities with periods of high renewable energy production. This not only maximizes the use of clean energy but also helps in balancing the intermittency of sources like solar and wind power. Through these capabilities, interoperability contributes to a more sustainable and efficient energy ecosystem, supporting the transition to greener energy solutions.

Additionally, standardized protocols that ensure interoperability, such as ISO 15118, OCPP, and EEBUS, provide a foundation for innovation and scalability within the EV industry. These standards enable different manufacturers and service providers to develop compatible products and services, fostering a competitive and diverse market. This, in turn, drives technological advancements and reduces costs, making electric mobility more accessible and appealing to a broader audience.

Overall, the benefits of interoperability are multifaceted, encompassing enhanced consumer convenience, improved grid stability, effective integration of renewable energy, and a supportive environment for innovation and market growth. By prioritizing interoperability, stakeholders in the EV ecosystem can create a more robust, efficient, and sustainable future for electric mobility and energy management.

5 P3 Energy.Lab

P3's Energy Lab is at the forefront of ensuring the reliability and effectiveness of smart and bi-directional charging technologies through its state-of-the-art testing solutions. The Energy Lab is dedicated to unlocking the full potential of these advanced charging systems through rigorous testing and validation of their functionalities in various scenarios. This includes the application of test tools that simulate real-world conditions to evaluate the performance and interoperability of electric vehicles (EVs) and electric vehicle supply equipment (EVSE).

The Energy Lab employs a comprehensive set of test methods that cover a wide range of use cases. These methods are tailored to validate the compatibility of EVs and charging stations with standardized protocols such as ISO 15118, OCPP and EEBUS. By ensuring compliance with these standards, P3 helps manufacturers and service providers to develop products that integrate seamlessly into the wider energy ecosystem.

In the future, it will also be possible to test and validate the development and use of vehicle-to-grid (V2G) technologies in the P3 Energy Lab. This not only contributes to grid stability, but also improves the efficiency and sustainability of energy use by facilitating the integration of renewable energy sources.

In addition, the Energy Lab focuses on the validation of intelligent charging functions, such as dynamic load balancing and demand response. By testing these functions, P3 ensures that charging systems can adapt to grid conditions in real time to optimize energy distribution and minimize the risk of grid overload. This capability is critical to maintaining a stable and resilient energy infrastructure, especially with the increasing uptake of EVs.

The Energy Lab also addresses the practical aspects of EV and EVSE deployment, such as configuration and fault code standards. By providing detailed insights into these areas, P3 helps players in the EV ecosystem to develop robust and reliable products that meet regulatory requirements and user expectations.

6 Conclusion

In conclusion, the integration of Electric Vehicles (EVs) and Electric Vehicle Supply Equipment (EVSE) into the energy ecosystem hinges on the interoperability of these systems. The adoption of standardized protocols such as ISO 15118, OCPP, and EEBUS is crucial for ensuring seamless communication and operation across diverse platforms. Interoperability enhances consumer convenience by allowing EVs to charge at any compatible station without issues, and it is instrumental in optimizing grid stability and efficiency through smart and bidirectional charging technologies.

P3's Energy.Lab plays a vital role in this landscape by offering comprehensive testing solutions that validate the performance and compatibility of EVs and charging infrastructure. Through rigorous testing, the Energy.Lab ensures that these systems meet the necessary standards and can operate reliably in real-world conditions. This not only supports the deployment of advanced technologies like Vehicle-to-Grid (V2G) but also facilitates the integration of renewable energy sources, contributing to a more sustainable and resilient energy grid.

The benefits of interoperability extend beyond immediate operational efficiencies, fostering innovation and market growth by creating a competitive and diverse environment. As the EV market continues to expand, prioritizing interoperability will be essential for scaling infrastructure and meeting the increasing demand for electric mobility. By adhering to standardized protocols and leveraging advanced testing solutions, stakeholders can ensure a robust, efficient, and user-friendly EV ecosystem, paving the way for a cleaner and more sustainable energy future.

7 Bibliography

[FfE23] Severin Sylla, Adrian Ostermann, Jeremias Hawran. Normenlandschaft für die Elektromobilität. Forschung für Energiewirtschaft, 2023

[ISOa] ISO 15118-1:2019. Road vehicles — Vehicle to grid communication interface. Part 1: General information and use-case definition

[ISOb] ISO 15118-20:2022. Road vehicles — Vehicle to grid communication interface. Part 20: 2nd generation network layer and application layer requirements

[OCPP] Open Charge Alliance. Open charge point protocol 2.0.1

[EEBUS22] EEBUS. Overview use cases, EEBus Initiative e.V., 2022

[IECa] IEC 63110-1:2022 ED1. Protocol for management of electric vehicles charging and discharging infrastructures - Part 1: Basic definitions, use cases and architectures

[IECb] IEC 63584 ED1. Open Charge Point Protocol (OCPP) (Fast track)

[IECc] IEC 63380-4 ED1. Local Charging station management systems and Local Energy Management Systems network connectivity and information exchange

[OSCP] Open Charge Alliance. Open smart charging protocol

[IECd] IEC 62746-10-1:2018. Systems interface between customer energy management system and the power management system - Part 10-1: Open automated demand response

[IEEE] IEEE 2030.5-2018. IEEE Standard for Smart Energy Profile Application Protocol

[IECf] IEC 61850:2024 SER. Communication networks and systems for power utility automation

[IECg] IEC 63382-1 ED1. Management of Distributed Energy Storage Systems based on Electrically Chargeable Vehicles (ECV-DESS)

Hochautomatisierte Test- & Integrationspyramide von Softwaremodultest bis Gesamtfahrzeugintegration

Max Leicht
max.leicht@bmw.de
+49 151 601 22367

BMW AG
BMW AG, Petuelring 130, 80788 München

Sebastian Dietze
sebastian.dietze@tracetronic.de
+49 351 205 768 248

tracetronic GmbH
tracetronic GmbH, Stuttgarter Str. 3, 01189 Dresden

Abstract

Die Durchführung von Software-, Integrations- und Systemtests sind elementarer Bestandteil der Fahrzeugentwicklung. Die stetig steigende Komplexität erfordert einen durchgängigen, effizienten, automatisierten und nachvollziehbaren Absicherungsprozess. Derzeit sind die Testprozesse und die in den einzelnen Entwicklungsstufen zum Einsatz kommenden Tools nicht vereinheitlicht und die Schnittstellen kaum standardisiert.

Dafür wird ein Plattformansatz vorgestellt, welcher sowohl virtuelle als auch reale Testumgebungen abdeckt und auf allen Integrationsstufen variabel einsetzbar ist. Der eine Teil der Plattform (Software Validation Platform) ermöglicht die kontinuierliche Validierung von Software sowohl in virtualisierten Testumgebungen mit Cloud-Infrastruktur als auch an klassischen Testinstanzen von einfachen Aufbauten bis zu Systemprüfständen. Das Ergebnis ist die Freigabe für die nächsthöhere Integrationsstufe. Der andere Teil der Plattform (Continuous Testing and Integration Platform) umfasst die gesamte Integrationspyramide vom Softwaremodul bis zur Gesamtfahrzeugintegration, denn kontinuierlich muss mit jeder neuen Softwareversion auch das Zusammenspiel überprüft werden. Damit dies möglich ist, werden zum einen standardisierte Schnittstellen für die Nutzung der Plattform und zum anderen geeignete Testautomatisierungstools sowie damit verbundene Prozesse vorgestellt, welche die gestellten Anforderungen an die durchgängige Automatisierung ermöglichen.

Schwerpunktthema: Testprozesse und Technologie

Neuigkeitswert:

Der vorgestellte Plattformansatz im Zusammenspiel mit den eingesetzten Automatisierungstools ermöglicht die kontinuierliche, nachvollziehbare und automatisierte Durchführung von Software- und Integrationstests in allen Stadien der Entwicklung (virtuelle Komponente bis Gesamtfahrzeugintegration).

Automotive Software Engineering & Automotive Testing – Kein One-fits-it-All

Die Automobilindustrie steht vor einem grundlegenden Wandel durch den Aufstieg von Software-Defined Vehicles (SDVs), in denen Software zunehmend zur treibenden Kraft wird. Im Gegensatz zur klassischen Fahrzeugentwicklung, die vor allem auf mechanische Systeme fokussiert war, rückt die Software nun in den Mittelpunkt der Innovation. Dabei unterscheidet sich die Entwicklung von Automobilsoftware grundlegend von der klassischen Softwareentwicklung, wie beispielsweise für Webshops. Während Webanwendungen oft in schnellen Iterationszyklen und mit weniger strikten Sicherheitsanforderungen entwickelt werden, erfordert Automotive Software strengste Sicherheitsstandards und Echtzeitfähigkeit. Fehler im Automobilbereich können fatale Folgen haben, was umfangreiche Testverfahren und zertifizierte Entwicklungsprozesse notwendig macht. Zudem müssen SDVs in einem hochkomplexen, vernetzten Ökosystem operieren, in dem Sensoren, Steuergeräte und cloudbasierte Dienste nahtlos zusammenarbeiten. Die Anforderungen an Skalierbarkeit, Zuverlässigkeit und Langlebigkeit sind hier deutlich höher als in klassischen IT-Projekten. Diese Unterschiede machen deutlich: Es gibt im Automotive Software Engineering kein „One-fits-it-all"-Ansatz. Jeder Aspekt, von der Softwarearchitektur bis hin zu den Testmethoden, muss auf die spezifischen Anforderungen der Automobilbranche zugeschnitten sein.

Automotive DevOps

Die Kernthemen von tracetronic sind Tests und deren Automatisierung in der Softwareentwicklung der Automobildomäne. Gestartet mit der reinen Testautomatisierung an Hardware-in-the-Loop Prüfplätzen im Bereich des Antriebsstrangs entwickelten wir uns über die Automatisierung der Testautomatisierung, die seit einigen Jahren unter dem Bezeichner Continuous Testing bekannt ist, hin zu Automotive DevOps.

Automotive DevOps beschreibt die Anwendung von DevOps-Prinzipien in der Automobilindustrie, wobei die spezifischen Anforderungen an Sicherheit, Compliance und Echtzeit-Funktionalität berücksichtigt werden. Im Vergleich zur klassischen IT ist Automotive DevOps besonders anspruchsvoll, da Software in komplexe Hardware- und Steuersysteme integriert werden muss, die hohen Sicherheits- und Qualitätsstandards wie ISO 26262 oder Automotive SPICE entsprechen. Der Fokus liegt auf kontinuierlicher Integration und Lieferung (CI/CD), um schnelle und zuverlässige Software-Updates zu gewährleisten, etwa für autonome Fahrfunktionen oder Over-the-Air (OTA) Updates. Diese Methode ermöglicht es Automobilherstellern, Entwicklungszyklen zu beschleunigen, Fehler schneller zu beheben und die Sicherheit von Fahrzeugen durch regelmäßige Software-Verbesserungen zu gewährleisten[EclipseCon][Intellias][DZone].

Automotive DevOps adressiert drei zentrale Herausforderungen in der Fahrzeugentwicklung: die Verkürzung der Feedbackzeit, die Optimierung der Testressourcen und -kosten sowie die zielgerichtete Aufbereitung von Feedback für verschiedene Stakeholder.

1. **Verkürzung der Feedbackzeit**: Durch den Einsatz von kontinuierlicher Integration und kontinuierlicher Lieferung (CI/CD) werden Software-Updates schneller getestet und freigegeben. Dies reduziert die Zeit, die zwischen der Entdeckung eines Problems und dessen Behebung vergeht, erheblich. Entwickler erhalten schneller Rückmeldungen zu neuen Codeänderungen, was die Entwicklungszyklen verkürzt[EclipseCon].
2. **Optimierung der Testressourcen und -kosten**: Automotive DevOps nutzt automatisierte Tests, um die vorhandenen Testressourcen effizienter zu nutzen. Virtuelle Testumgebungen,

die auf Cloud-Infrastrukturen basieren, ermöglichen es, Tests in einer kostengünstigen, skalierbaren Umgebung durchzuführen. So kann auf teure Hardwaretests in frühen Entwicklungsphasen verzichtet werden, während die Qualität sichergestellt bleibt[Intellias].

3. **Adressatengerechte Aufbereitung des Feedbacks**: Ein weiteres wichtiges Ziel von Automotive DevOps ist die Bereitstellung von auf die jeweiligen Zielgruppen abgestimmten Rückmeldungen. Entwickler benötigen detaillierte technische Informationen, während Projektmanager und Stakeholder oft einen Überblick über den Fortschritt und potenzielle Risiken erhalten müssen. DevOps-Lösungen sorgen dafür, dass jeder Beteiligte die relevanten Informationen in einer für ihn verständlichen Form erhält, was die Kommunikation im gesamten Entwicklungsprozess verbessert[DZone].

Insgesamt trägt Automotive DevOps dazu bei, die Entwicklungsprozesse zu beschleunigen, die Kosten zu senken und die Effizienz durch zielgerichtete Kommunikation zu steigern. Dafür ist es notwendig im Bereich Continuous Integration das Bereitstellen der Testobjekte mit der Durchführung von Continuous Testing eng zu verzahnen.

Während im Bereich der klassischen Softwareentwicklung mit GitLab und GitHub (und Azure DevOps) ein ganz klarer und einheitlicher Plattformansatz verfolgt wird, ist die Wahl einer einheitlichen Plattform oder Toolkette in der Automobilsoftwareentwicklung nicht einfach. Diverse Programmierumgebungen, Architekturen, Kommunikationskanäle und die damit verbundenen Anforderungen und Möglichkeiten machen eine einheitliche Toolentscheidung schwer bis unmöglich. Während in softwaregeprägten Bereichen wie dem Infotainment cloud-basierte state-of-the-art Lösungen nahezu anpassungslos übernommen werden können, hat die Absicherung des Antriebsstrangs auch heute noch starke Abhängigkeiten zur verwendeten Zielhardware. Dieser Spagat ist in den vorhandenen Marktlösungen meist nur schwer möglich. Tools, die aus der Automobilwelt kommen, sind zumeist nicht für die enormen Testdurchsätze optimiert. Tools aus der klassischen Softwareentwicklung dagegen sind mit der Vielzahl an Hardwareabhängigkeiten, Echtzeit- sowie Securityanforderungen überfordert.

Für die Entwicklung einer Automotive DevOps Plattform ergibt sich als essentielle Anforderung die Unterstützung einer Vielzahl an notwendigen Tools. Es bedarf eines Single-Point-Of-Truth mit einer Oberfläche die alle Informationen gemeinsam aufbereiten kann und einen guten Trade-off aus Generik und Spezialisierung anbietet, gesucht wird eine Möglichkeit eine Richtung vorzugeben, ohne dabei die notwendige Individualität einzuschränken.

Über vereinheitlichte Konzepte soll es ermöglicht werden, Nachvollziehbarkeit und Identifizierbarkeit von allen Objekten und Artefakten im Entwicklungs- und Testprozess zu erreichen. Zentraler Bestandteil dafür ist ein gemeinsames Build- und Abhängigkeitsmanagement. Dieses System muss im weiteren Verlauf lückenlos in die Nachfolgeprozesse integriert werden, um bspw. im Testprozess aufgetretene Fehler, auch nach dem Kompilieren und Verpacken, während der Verwendung auf einem Steuergerät, noch der richtigen Software-Unit in der richtigen Version zuordnen zu können. Dies ist eine elementare Anforderung, um hochautomatisierte Testprozesse über mehrere Teststufen zu ermöglichen. Außerdem muss ein Weg gefunden werden, dass verwendete Tools und Technologien auch nach der initialen Versionsdefinition noch aktualisiert werden können, um keine Sicherheitsrisiken zu erzeugen und auch nach dem ersten Release der Fahrzeugsoftware noch eine exzellente Entwicklungsumgebung zu schaffen.

Solange grundlegende Konzepte vorhanden sind, Nachvollziehbarkeit, Identifizierbarkeit und Reproduzierbarkeit sichergestellt werden kann und eine grundsätzlich gemeinsam verwendete

Umgebung bereitgestellt wird, kann von einer Automotive DevOps Plattform gesprochen werden. Der Anwender ist dadurch in der Lage seinen Fokus nicht auf Themen wie das Kompilieren oder Verteilen von Softwarefragmenten zu legen, sondern direkt an der Wertschöpfung und damit an der Modellierung oder Implementierung von Fahrzeugsoftware zu arbeiten. So entstehen schnell neue Endnutzerfunktionen – aber sicher.

Fallbeispiele

Im Rahmen des Vortrags wird nicht nur theoretisch beleuchtet, welche Anforderungen an eine Automotive DevOps Plattform für die Entwicklung und Absicherung moderner Automobilsoftware gestellt werden. Im Vortrag gibt es zusätzlich Einblicke in Konzepte, Automatisierungsmöglichkeiten und allgemein aus dem Testalltag. Stay Tuned.

Literaturverzeichnis

[EclipseCon] https://www.eclipsecon.org/2023/sessions/orchestration-automotive-devops-pipelines-chances-and-constraints-open-source

[Intellias] https://intellias.com/devops-cloud-computing-automotive/

[DZone] https://dzone.com/articles/the-scene-of-devops-in-the-automotive-industry

Glaubwürdigkeit von Software-in-the-Loop-Testsystemen: Potentiale und Herausforderungen

Anne Kieneke[1], Philipp Schenkel[2], Tobias Pett[3], Jacob Langner[2], Okan Ecin[1], Ina Schaefer[3]

[1]Dr. Ing. h.c. F. Porsche AG
[2]FZI Forschungszentrum Informatik
[3]Karlsruher Institut für Technologie

anne.kieneke@porsche.de, schenkel@fzi.de, tobias.pett@kit.edu,
langner@fzi.de, okan.ecin@porsche.de, ina.schaefer@kit.edu

Abstract: Software-in-the-Loop (SiL)-Testen ist eine Methode zum Testen von Software in einer Simulationsumgebung. Im Gegensatz zu den verbreiteteren Hardware-in-the-Loop (HiL) Testsystemen bieten SiL-Testsysteme einige Vorteile, wie z.B. einen frühzeitigeren Einsatz im Entwicklungsprozess von Steuergeräten, geringere Kosten, die Möglichkeit des Debuggings und die Skalierbarkeit. Dies führt, besonders im Bereich der Verifikation und Validierung von automatisierten Fahrfunktionen, zu einem vermehrten Einsatz von dieser Methode der virtuellen Absicherung. Um SiL-Tests zur Homologation und Freigabe von verteilten Funktionen nutzen zu können, muss gezeigt werden, dass die SiL-Testergebnisse mit den Ergebnissen von Fahrzeugtests übereinstimmen und somit glaubwürdig sind. Bisherige Arbeiten bewerten die Testergebnisse von SiLs bestehend aus einem Funktionsblock bzw. einer Komponente als Testobjekt. Dabei werden Aufzeichnungen der Signale des SiLs mit denen eines HiLs oder eines realen Fahrzeugs verglichen. Diese Methoden können nicht auf Testsysteme, die aus einem Verbund aus virtuellen Steuergeräten und Simulationsmodellen bestehen, übertragen werden. Ziel dieser Arbeit ist die Untersuchung der Problemstellung, wie die Glaubwürdigkeit von SiL-Testsystemen mit einem Verbund an V-ECUs analysiert werden kann, und die Erarbeitung einer Forschungsmethodik zur Lösung der Problemstellung.

1 Einleitung

Immer mehr Funktionen eines Autos werden durch Software, die auf einem Steuergeräten (engl. electronic control units) (ECU) ausgeführt wird, realisiert [Sax08]. Sowohl zur Sicherstellung der Qualität und Sicherheit der Software als auch als Anforderung für die Homologation des Fahrzeugs muss die Software der ECUs ausführlich getestet werden. Beim Testen in einem Prototypen-Fahrzeug lässt sich die Software in Kombination mit der Hardware, den Sensoren und Aktoren, sowie in realen Umgebungen testen. Da Testen im Fahrzeug zeitaufwendig und kostenintensiv ist, hat sich die Hardware-in-the-loop (HiL)-Technologie als Vorstufe zum Fahrzeugtest und Methode zum Testen

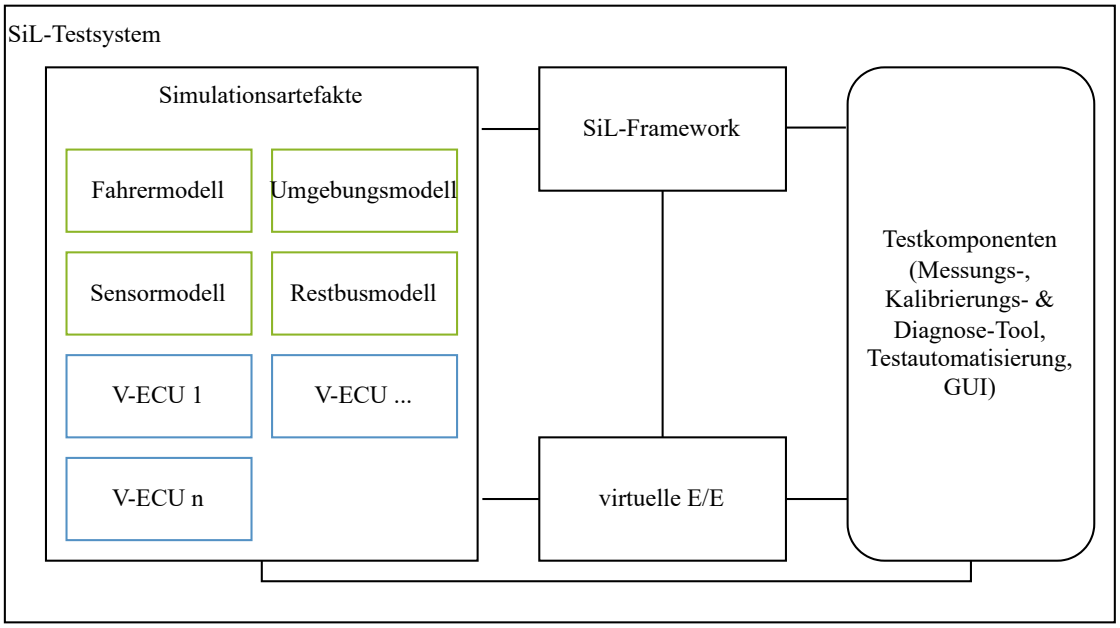


Figure 1: Architektur eines Software-in-the-loop (SiL)-Testsystem nach [VDA22]

von Steuergeräte-Software etabliert. Dabei werden die zu testenden ECUs mit einem Echtzeitrechner verbunden, auf dem verschiedene Simulationsmodelle, wie Fahrermodell, Umgebungsmodell oder Restbusmodell ausgeführt werden.

Eine Virtualisierung der HiL-Technologie ist die SiL-Technologie, bei der statt der realen ECU ein virtuelles Steuergerät (engl. virtual electronic control units) (V-ECU) verwendet wird. V-ECUs enthalten Software der realen Steuergeräte, je nach Level auf verschiedenen Abstraktionsebenen [iVi20]. Die Architektur eines SiL-Testsystems ist in Abbildung 1 dargestellt. Die V-ECUs und die verschiedenen Modelle werden im SiL-Framework ausgeführt. Die virtuelle E/E bildet das elektrische Boardnetz des Autos mit den Kommunikationsbussen und Versorgungsleitungen nach. Die Testkomponenten ermöglichen die automatisierte Bedienung des Testsystems von außen, sowie die Aufzeichnung der Messungen. Die Vorteile des SiL-Testens sind die Möglichkeit des frühzeitigen Einsatz im Entwicklungsprozess, die Hardwareunabhängigkeit, die daraus resultierenden geringeren Kosten und die einfache Skalierbarkeit.

Ein erstes Projekt versucht unter anderem SiL-Testsyteme zur Typgenehmigung zu nutzen[1]. Damit dies möglich ist, muss gezeigt werden, dass die Testergebnisse eines SiL-Testsytem denen eines Tests im realen Fahrzeug entsprechen, d.h. dass das SiL-Testsystem bzw. die SiL-Testergebnisse glaubwürdig sind. Erste Arbeiten [SVGN24, RGA+19, OS23, JKL16, PS12, SFB+23, KPJ+18, RBB+20] untersuchen die Glaubwürdigkeit anhand von Expertenwissen oder Referenzsystemen basierend auf Signalverläufen für SiL-Testsysteme mit einer einstelligen Anzahl an Signalen und nur einer V-ECU. SiL-Testsysteme zur Freigabe von verteilten Funktionen bestehen jedoch aus allen an der Funktion beteiligten V-ECUs, einem sogenannten Verbund aus zwei bis 100 V-ECUs und einer Vielzahl an Signalen. Daher skalieren die Methoden zur Analyse von Glaubwürdigkeit nicht für SiL-Testsysteme zur Freigabe von verteilten Funktionen. Daher fehlt eine Methodik zur

[1]Digital Loop: `https://www.digi-loop.com/`

Analyse der Glaubwürdigkeit von SiL-Testsystemen mit einer hohen Anzahl an Signalen bzw. der Testergebnisse von diesen über den gesamten Lebenszyklus, um SiL-Testsysteme für die Homologation oder die Freigabe von verteilten Funktionen nutzen zu können.

Um diese Herausforderung zu überwinden, muss (1) Glaubwürdigkeit für SiL-Testsysteme definiert, (2) eine Bewertungsmethode der Glaubwürdigkeit von SiL-Testsysteme oder SiL-Testergebnissen erarbeitet und (3) ein Prozess zur Sicherstellung der Glaubwürdigkeit über den gesamten Lebenszyklus eines SiL-Testsystems ausgearbeitet werden. Ziel dieser Arbeit ist die Untersuchung der Problemstellung, wie die Glaubwürdigkeit von SiL-Testsystemen mit einem Verbund an V-ECUs analysiert werden kann, und die Erarbeitung einer Forschungsmethodik zur Lösung der Problemstellung.

2 State of the Art

In diesem Abschnitt werden relevante Arbeiten zur Analyse der Glaubwürdigkeit von SiL-Testsystemen vorgestellt. Dazu betrachten wir (1) die Definition von Glaubwürdigkeit, (2) Methoden zur Analyse und Bewertung der Glaubwürdigkeit von SiL-Testsystemen und (3) Arbeiten zum Thema Glaubwürdigkeit über den Lebenszyklus von SiL-Testsystemen.

Definitionen von Glaubwürdigkeit. Der NASA Standard 7009b [NAS24] definiert Glaubwürdigkeit für Modelle und Simulation als Eigenschaft, Vertrauen in die Modell- und Simualtionsergebnisse schaffen zu können. Dafür sind vor allem Nachvollziehbarkeit und Verständlichkeit der Simulation und der Simulationsergebnisse wichtig [HS21]. Liu et al. [FMZ05] definiert Glaubwürdigkeit als Vertrauen des Nutzers in das Modell oder die Simulation für den vorgesehenen Einsatz, wobei sich die Glaubwürdigkeit aus der Validität, der Korrektheit, der Verlässlichkeit, der Integrierbarkeit und der Benutzerfreundlichkeit zusammensetzt. Die bestehende Definition von Glaubwürdigkeit für Modelle und Simulation kann für SiL-Testsysteme nicht vollständig angewendet werden, da die Simualtionsartefakte eines SiL-Testsystems gleichermaßen Modelle und V-ECUs sind [VDA22]. Anders als Modelle bilden V-ECUs ein Realsystem nicht nach, sondern enthalten die Software des realen Steuergeräts, je nach Level der V-ECUs in unterschiedlichen Ausprägungen [iVi20].

Analyse und Bewertung der Glaubwürdigkeit von SiL-Testsystemen. Wir identifizieren zwei wesentliche Ansätze für die Analyse und Bewertung der Glaubwürdigkeit von SiL-Testsystemen: (i) Analyse und Bewertung mit Hilfe von Expertenwissen [SVGN24, RGA+19, OS23, JKL16, PS12] und (ii) Bewertung anhand von Referenzsystemen [SFB+23, KPJ+18, OS23, PS12, RBB+20]

Bewertung mit Hilfe von Expertenwissen. In verschiedenen Arbeiten [SVGN24, RGA+19, OS23, JKL16, PS12] wird die Glaubwürdigkeit von SiL-Testsystemen mit Hilfe von Experten bewertet. Diese betrachten die Verläufe der einzelnen Signale oder bewerten die Reaktion des Testsystems auf ausgewählte Testszenarien. Für SiL-Testsysteme mit einem Verbund an V-ECUs zur Freigabe von verteilten Funktionen ist die Bewertung mit Hilfe von Expertenwissen nicht möglich, da die Anzahl der zu betrachtenen Testszenarien und die Komplexität der verteilten Funktionen zu groß ist.

Bewertung der Glaubwürdigkeit anhand von Referenzsystemen. Bei diesem Ansatz basiert die Bewertung auf einem Vergleich der Messdaten des SiL-Testsystems mit den Messdaten eines Referenzsystems. Als Referenzsystem dienen Versuchsfahrzeuge [SFB+23, KPJ+18], HiL-Testsysteme [OS23, PS12] oder Versuchsfahzeuge und HiL-Testsysteme [RBB+20]. Dabei werden die Signale graphisch oder anhand von verschiedenen Metriken mit den Signalen des Referenzsystems verglichen. In diesen Arbeiten werden SiL-Systeme mit einer V-ECU und einer einstelligen Anzahl an Signalen betrachtet. Der manuelle Vergleich skaliert nicht für SiL-Testsysteme mit einem Verbund an V-ECUs aufgrund der hohen Anzahl an Signalen. Außerdem stehen nicht immer Referenzsysteme mit gleichem Softwarestand zur Verfügung, da die Lieferzeitpunkte der V-ECUs vor den Lieferzeitpunkten der entsprechenden ECUs liegen.

Glaubwürdigkeit über den Lebenszyklus von SiL-Testsystemen. Raghupatruni et al. [RKG+21] beschreibt eine Continuous-development, -integration,- testing und -deployment-Pipeline) (CX-Pipeline) für SiL-Testumgebungen aus Sicht eines ECU Lieferanten. Die CX-Pipeline beginnt mit der Änderung des Codes einer ECU. Darauf folgen Unit-Tests und Code-Reviews und anschließend die Generierung der V-ECU. Nach der Integration der neuen V-ECU in das SiL-Testsystem werden Smoke-Tests und eine Qualifizierung in Form eines Vergleichs mit einem Referenzsystems durchgeführt. Nach erfolgreicher Qualifizierung und somit Sicherstellung der Glaubwürdigkeit der V-ECU kann die V-ECU zum Testen verwendet werden. Wenn jede V-ECU und jedes Modell solch eine CX-Pipeline durchlaufen kann die Glaubwürdigkeit eines SiL-Testsystems über Updates von einzelnen V-ECUs und Modellen oder auch des gesamten Verbunds an V-ECUs im Verbund-Release sichergestellt werden. Für den Einsatz der CX-Pipeline bei einem Automobilshersteller muss diese angepasst werden, da die Hersteller die V-ECU oft als Black-Box erhalten und dadurch Unit-Tests und Code-Reviews nicht möglich sind.

3 Problem Statement

SiL-Testsysteme bilden die Realität durch den Einsatz von Modellen nach. Da Modelle stets eine Abstraktion der Realität darstellen [DGK07], entsteht eine Differenz in der Glaubwürdigkeit zwischen dem realen Fahrzeug und dem SiL-Testsystem, siehe Abbildung 2. Für den breiten Einsatz von SiL-Testsystemen ist eine sorgfältige Prüfung der Glaubwürdigkeit der mit den SiL-Testsystemen erzeugten Testergebnisse notwendig [RBB+20]. Als Ergebnis dieser Prüfung werden belastbare, quantitative Aussagen über die Glaubwürdigkeit des SiL-Testsystems oder der SiL-Testergebnisse bzw. quantitative Aussagen über die Abweichung vom Fahrzeugtest benötigt. Um diese Herausforderung zu überwinden und Glaubwürdigkeit von SiL-Testsystemen sicherzustellen, wird (1) eine einheitliche Definition der Glaubwürdigkeit für SiL-Testsysteme, (2) eine Methode zur Bewertung von SiL-Testsystemen oder SiL-Testergebnissen mit quantitativen Aussagen über die Glaubwürdigkeit und (3) ein Prozess zur Sicherstellung der Glaubwürdigkeit über den gesamten Lebenszyklus eines SiL-Testsystems benötigt.

Die bestehenden Definitionen von Glaubwürdigkeit für Modelle und Simulationen [NAS24, HS21, FMZ05] kann auf SiL-Testsysteme nicht angewandt werden, da die im SiL-

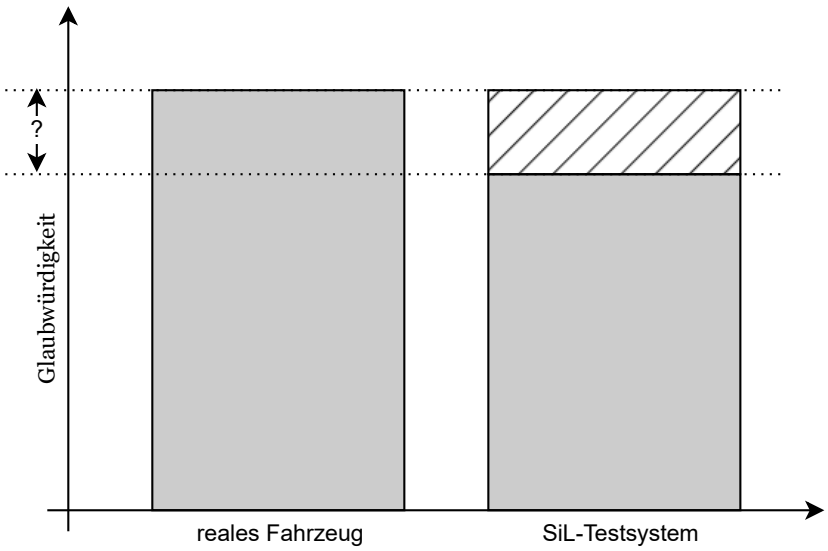


Figure 2: Glaubwürdigkeit von SiL-Testsystemen

Testsystem enthaltenen V-ECUs keine Modelle sind. Daher fehlt eine Definition für Glaubwürdigkeit von SiL-Testsystemen. Die verschiedenen Methoden zur Bewertung der Glaubwürdigkeit von SiL-Testsystemen [SVGN24, RGA+19, OS23, JKL16, PS12, SFB+23, KPJ+18, RBB+20] skalieren nicht für SiL-Testsystemen mit mehreren V-ECUs aufgrund der hohen Anzahl an Signalen, sodass eine Bewertungsmethode für SiL-Testsystemen mit einem Verbund an V-ECUs fehlt. Außerdem ist kein Prozess vorhanden, der die Glaubwürdigkeit von SiL-Testsystemen über deren gesamten Lebenszyklus betrachtet, von der die Erstellung des Testsystem über das Update von einzelnen V-ECUs und Modellen und das Update des gesamten Verbunds an V-ECUs im Verbund-Release. Um diese Herausforderungen zu lösen, wird die folgende zentrale Forschungsfrage (**MRQ**) aufgestellt:

MRQ: Wie kann die Glaubwürdigkeit von SiL-Testsystemen analysiert werden?

4 Forschungsmethodik

Zur Beantwortung der oben aufgestellten zentrale Forschungsfrage leiten wir drei Teilforschungsfragen (**RQ1-RQ3**) ab und beschreiben die Forschungsmethodik, um die drei Teilforschungsfragen zu bearbeiten:

RQ1: Was bedeutet Glaubwürdigkeit für SiL-Testsysteme?

Um die Glaubwürdigkeit von SiL-Testsystemen sicherzustellen zu können, muss zuerst Glaubwürdigkeit spezifisch für SiL-Testsysteme definiert werden. Teil der Definiton sollen quantifizierbare Kriterien eines glaubwürdiges SiL-Testsystem sein. Dazu wird mit einer Literaturstudie in Form einer systematischen Literaturrecherche (SLR) nach der Leitlinie von Kitchenham [Kit04] der Stand der Technik ermittelt. Die Ergebnisse der Literaturstudie werden im Anschluss kategorisiert und darauf aufbauend eine Definition der Glaubwürdigkeit von SiL-Testsystemen erarbeitet.

RQ2: Wie kann die Glaubwürdigkeit von SiL-Testsystemen bewertet werden?

Anhand der Definition der Glaubwürdigkeit kann im nächsten Schritt eine Bewertungsmethode für die Glaubwürdigkeit von SiL-Testsystemen erarbeitet werden. Die Bewertungsmethode soll als Ergebnis quantitative Aussagen über die Glaubwürdigkeit eines SiL-Testsystem oder der SiL-Testergebnisse liefern. Grundlage der quantitativen Bewertung der Glaubwürdigkeit bildet eine Metrik basierend auf den Kriterien der ersten Teilforschungsfrage (**RQ1**). Teile der Bewertungsmethode müssen ohne Referenzsysteme anwendbar sein, da in frühen Phasen der Fahrzeugentwicklung Prototypen-Fahrzeuge und HiL-Testsysteme als Referenz nicht zur Verfügung stehen. Basierend auf den Ergebnissen der Teilforschungsfrage (**RQ1**) wird eine quantitativen Bewertungsmethode Metrik erarbeitet. Diese wird an verschiedenen SiL-Testsystemen mit einem Verbund an V-ECUs empirisch evaluiert.

RQ3: Wie kann die Glaubwürdigkeit über den Lebenszyklus der SiL-Testsysteme sichergestellt werden?

Es wird ein Prozess erarbeitet, mit dem die Glaubwürdigkeit eines SiL-Testsystems über den gesamten Lebenszyklus sichergestellt werden kann. Teil dieses Prozesses ist eine ausführliche Bewertung der Glaubwürdigkeit des SiL-Testsystems mit den Ergebnissen der zweiten Teilforschungsfrage (**RQ2**) nach dem Aufbau des Testsystems, vor der ersten Verwendung. Außerdem enthält der Prozess erneute Bewertungen der Glaubwürdigkeit über den gesamten Lebenszyklus des Testsystems nach Änderungen in den Modellen oder bei Lieferungen von neuen V-ECUs. Dieser Prozess wird begleitend zu dem Aufbau eines SiL-Testsystem mit einem Verbund an V-ECUs erarbeitet und empirisch an verschiedenen SiL-Testsystemen evaluiert.

5 Zusammenfassung

In dieser Arbeit wurde die Problemstellung, wie die Glaubwürdigkeit von SiL-Testsystemen mit einem Verbund an V-ECUs analysiert werden kann, untersucht und eine Forschungsmethodik zur Lösung der Problemstellung erarbeitet, damit SiL-Testsysteme zur Homologation oder Freigabe von verteilten Funktionen genutzt werden können. Dafür muss über den gesamten Lebenszyklus des Testsystems sichergestellt werden, dass das SiL-Testsysteme bzw. die SiL-Testergebnisse glaubwürdig sind. Es muss (1) eine einheitliche Definition der Glaubwürdigkeit von SiL-Testsystemen, (2) eine Methode zur Bewertung der Glaubwürdigkeit von SiL-Testsystemen mit einem Verbund an V-ECUs und (3) ein Prozess zur kontinuierlichen Überwachung der Glaubwürdigkeit über den gesamten Lebenszyklus eines SiL-Testsystems vorhanden sein. Aus den drei identifizierten Anforderungen wurden drei Teilforschungsfragen (**RQ1-RQ3**) für die Zukunft abgeleitet und die Forschungsmethodik beschrieben.

References

[DGK07] Stephanie Demers, Praveen Gopalakrishnan, and Latha Kant. A Generic Solution to Software-in-the-Loop. In *MILCOM 2007 - IEEE Military Communications Conference*, pages 1–6, Orlando, FL, USA, October 2007. IEEE.

[FMZ05] Fei Liu, Ming Yang, and Zicai Wang. Study on Simulation Credibility Metrics. In *Proceedings of the Winter Simulation Conference, 2005.*, pages 2554–2560, Orlando, FL. USA, 2005. IEEE.

[HS21] Hans-Martin Heinkel and Kim Steinkirchner. Credible Simulation Process, August 2021.

[iVi20] ProSTEP iVip. Virtual Electronic Control Units: Smart Systems Engineering Requirements for the Standardization of Virtual Electronic Control Units (V-ECUs). Darmstadt, 2020.

[JKL16] Sooyong Jeong, Yongsub Kwak, and Woo Jin Lee. Software-in-the-Loop simulation for early-stage testing of AUTOSAR software component. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 59–63, 2016.

[Kit04] Barbara Kitchenham. Procedures for Performing Systematic Reviews. *Keele, UK, Keele Univ.*, 33, August 2004.

[KPJ+18] Soon Kwon, Jaehyeong Park, Heechul Jung, Jihun Jung, Min-Kook Choi, Iman R. Tayibnapis, Jin-Hee Lee, Woong-Jae Won, Sung-Hoon Youn, Kwang-Hoe Kim, and Tae Hun Kim. Framework for Evaluating Vision-based Autonomous Steering Control Model. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1310–1316, 2018.

[NAS24] Standard for Models and Simulations, 2024.

[OS23] Parth Pankajbhai Oza and Dhaval Shah. Model-Based Software Development for Predictive Battery Temperature Estimation. In *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–7, 2023.

[PS12] Em Poh Ping and Sim Kok Swee. Simulation and experiment of automatic steering control for lane keeping manoeuvre. In *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, volume 1, pages 105–110, 2012.

[RBB+20] Indrasen Raghupatruni, S. Burton, M. Boumans, T. Huber, and A. Reiter. Credibility of software-in-the-loop environments for integrated vehicle function validation. In Michael Bargende, Hans-Christian Reuss, and Andreas Wagner, editors, *20. Internationales Stuttgarter Symposium*, pages 299–313. Springer Fachmedien Wiesbaden, Wiesbaden, 2020. Series Title: Proceedings.

[RGA+19] Indrasen Raghupatruni, Thomas Goeppel, Muhammed Atak, Julien Bou, and Thomas Huber. Empirical Testing of Automotive Cyber-Physical Systems with Credible Software-in-the-Loop Environments. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1–6, 2019.

[RKG+21] Raghupatruni, Indrasen, Karjee, Sambuddha, Gupta, Anupam, Naik, Venkatesh, and Huber, Thomas. Towards Establishing Continuous-X Pipeline Using Modular Software-in-the-Loop Test Environments. In *Symposium on International Automotive Technology*. SAE International, September 2021. ISSN: 0148-7191.

[Sax08] Eric Sax, editor. *Automatisiertes Testen eingebetteter Systeme in der Automobilindustrie*. Hanser, München Wien, 2008.

[SFB+23] L. Scialacqua, L.J. Foged, P. Berlt, B. Altinel, C. Bornkessel, M. A. Hein, R. Hoppe, T. Hager, and J. Soler Castany. Combination of Measurement and Simulation for Fast Virtual Drive Testing. In *2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI)*, pages 25–26, 2023.

[SVGN24] Sajnani, Abhishek, Vernekar, Kiran, Gosavi, Rupesh, and Naik, Venkatesh. Electric Powertrain - Standardization of Adaptable Software in Loop System. In *Symposium on International Automotive Technology*. SAE International, January 2024. ISSN: 0148-7191.

[VDA22] Software-in-the-Loop (SiL) Standardisierung, 2022.

Tests und Fahrversuche für sehr leichte elektrische Nutzfahrzeuge

Prof. Dr. Andreas Daberkow, B.Sc. Verena Barske

Hochschule Heilbronn HHN
Studiengang Automotive Systems Engineering ASE
Max-Planck-Str. 39
74081 Heilbronn
andreas.daberkow@hs-heilbronn.de
vbarske@stud.hs-heilbronn.de

Abstract: Kleine emissionsarme elektrische Nutzfahrzeuge sind ein wichtiges Element einer künftigen urbanen Mobilität. Doch auch bei geringen Geschwindigkeiten können in Fußgängerzonen oder im gemischten Verkehr Ausweichmanöver erforderlich sein. Viele der in der Regel kleineren Hersteller haben nicht die Entwicklungsressourcen, um die aus dem Automobilbereich langjährig bewährten und kostenoptimierten Sicherheitssysteme zu entwickeln, zu adaptieren und in den Fahrzeugen zu implementieren. Am Beispiel eines elektrisch angetriebenen Lastendreirades wird gezeigt, wie neben einem Lenkwinkelsensor ein Smartphone als preisgünstiges Messgerät eingesetzt werden kann. Es werden aus der Automobilentwicklung abgeleitete Tests und Fahrversuche in der Kreisbahn durchgefahren. Schon bei geringen Kurvengeschwindigkeiten können Ausweichmanöver auf ein kritisches Kipp- und Schleuderverhalten des Fahrzeugs führen. Konzeptentwürfe zur Reduzierung des Verletzungsrisikos für Insassen und Passanten runden den Beitrag ab.

1 Einleitung

Leichte elektrische Nutzfahrzeuge als Transportalternative im Wirtschaftsverkehr von Handwerk, Dienstleistung und Handel werden im Projekt “Ich entlaste Städte” erforscht [DLR24]. Die nachfolgend beschriebenen Fahrversuche werden mit dem Typ des dreirädrigen Lastenrads aus Abb. 1 links durchgeführt. Das Lastenrad aus Abb. 1 Mitte ist vierrädrig und hat einen Wetterschutz. In Abb. 1 rechts ist ein dreirädriges Lastendreirad mit einem Wechselcontainer abgebildet. Alle Fahrzeuge werden mittels Muskelkraft und mit einem unterstützenden Elektromotor bewegt (maximal elektrisch unterstützte Geschwindigkeit 25 km/h). Ladevolumina und Gesamtgewichte zu den nachfolgend als eLastenräder bezeichneten Fahrzeugen finden sich in [DLR24], der Projektträger ist das DLR (Deutsches Zentrum für Luft- und Raumfahrt).

Abbildung 1: Leichte elektrische Nutzfahrzeuge, DLR-Projekt “Ich entlaste Städte 2” [DLR24], Bildquelle DLR / Amac Garbe

Das dreirädrige Prinzip mit einem lenkbaren Rad vorne nach Abb. 1 links erlaubt eine von der Last nicht beeinträchtigte Sicht. Die starre Konstruktion mit einer ungefederten Pritsche hat einen einfachen Aufbau mit vielen Übernahmekomponenten aus der Zweiradtechnik. Den größten Zuwachs im deutschen Markt 2023 findet man bei eLastenrädern in Höhe von 189.000 Stück mit einem Zuwachs von 14,5 %. Der Anteil von eLastenrädern an elektrisch unterstützten Fahrrädern ist von 7,5 % auf 9 % gestiegen [ZIV23]. Verkehrssichere Fahrzeuge müssen eine geeignete Ladungssicherung oder Maßnahmen zum Selbst- und Kontrahentenschutz (hier Fußgänger) aufweisen. Weitere Maßnahmen zur Fahrzeugsicherheit umfassen die Qualität von Bremsen und Reifen und ggf. Fahrerassistenzsysteme wie Antiblockier- oder Antischleudersysteme. Das Team der Hochschule Heilbronn hat für eLastenräder im gemeinschaftlichen Verkehr mit Fußgängern und anderen Fahrzeugen mögliche kritische Fahrsituationen herausgearbeitet, siehe Abb. 2.

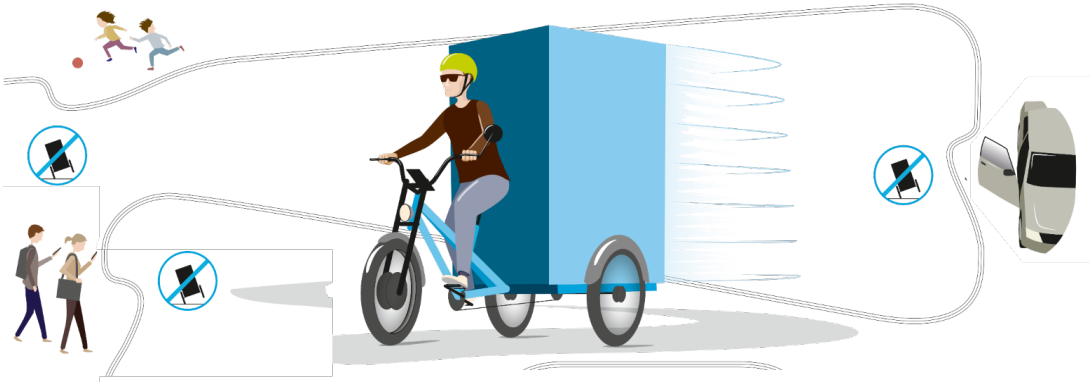


Abbildung 2: eLastenräder und kritische Fahrsituationen im gemeinschaftlich genutzten Verkehrsraum

Die DIN 79010 „Fahrräder – Transport- und Lastenfahrrad – Anforderungen und Prüfverfahren für ein- und mehrspurige Fahrräder“ [DIN79] enthält erste Sicherheitsregeln.

Die Richtlinie umfasst sicherheitstechnische Ausführungen wie z.B. zur Berechnung der Schwerpunktlage oder zur quasistatischen seitlichen Kippstabilität. Es ist eine Kreisfahrt ohne Abheben von Rädern auf einer ebenen Fläche mit einer konstanten Geschwindigkeit von 8 km/h auf einer Kreisbahn mit Durchmesser von 10 m durchzuführen. Dynamische Ausweichmanöver sind nicht vorgesehen.

Dies ist für das Team ASE an der HHN die Motivation für erste praktische Untersuchungen und Testfahrten mit einem eLastendreirad [Da22]. Alle kritischen Fahrsituationen nach Abb. 2 erfordern, dass FahrerInnen des eLastendreirades zur Vermeidung eines Zusammenstoßes reagieren müssen. Zwei Forschungsfragen dazu sollen beantwortet werden:

- Ist ein Kippen von eLastendreirädern in Kurven oder bei Ausweichmanövern eine Vermutung oder tatsächlich ein kritischer Fahrzeugsicherheitsaspekt?
- Wie kann der Aufwand einer Mess- und Sicherheitstechnik für eLastendreiräder durch Übernahme von Komponenten aus einer Automobilentwicklung gering gehalten werden ?

2 Testfahrzeug und erste Testfahrten

Das eLastendreirad des Herstellers [Radk24] hat einen elektrischen Vorderradantrieb, die Teleskopgabel ist gefedert und gedämpft und der Pedalantrieb wirkt auf ein Rad der ungefederten Hinterachse. Scheibenbremsen wirken am Vorderrad und an beiden Hinterrädern. Die elektrische Antriebsleistung beträgt 250 W bei einem maximalen Drehmoment von 70 Nm. Mit einem Achsabstand von 1818 mm und einer Spur von 940 mm ist das Fahrzeug kompakt. Für die fahrdynamischen Untersuchungen wird das Fahrzeug mit einer Stützkonstruktion ergänzt, siehe Abb. 3, welche das Fahrzeuggewicht und die Schwerpunktlage nur unwesentlich beeinflusst. Das Fahrzeug ist während der Testfahrten unbeladen. Das Kippen des Fahrzeugs wird dadurch bewirkt, dass während einer Fahrt mit konstanten Bahngeschwindigkeiten v von der Fahrerin ein sprungförmiger Lenkwinkel δ zum Kreismittelpunkt hinein ausgeführt wird.

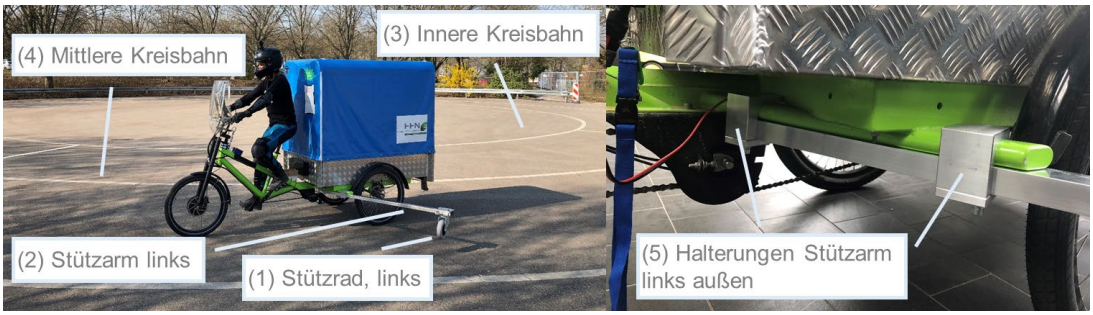


Abbildung 3: Stützkonstruktion des eLastendreirads [Radk24] für Testfahrten

Gummi-Stützräder (1) in Abb. 3 an jedem Stützarm (2) können das Fahrzeug nach dem Abheben eines Hinterrades führen. Die befahrenen Kreisbahnen des ASE-Testgeländes haben einem Durchmesser D_i = 24,1 m für die innere Bahn (3) sowie von D_m = 31,1 m für die mittlere Bahn (4). Die Halterungen (5) erlauben eine schnelle Montage bzw. Demontage der Stützarme.

Für die Tests wurde ein Gesamtprogramm mit sechs Testprogrammen und verschiedenen Fahrmanövern bei Geschwindigkeiten von 10, 15 und 20 km/h umgesetzt. Jedes Programm startet mit einer Einfahrt in die Kreisbahn und durchläuft die mittlere und innere Kreisbahn, an einem fixen Ort wird der Lenkwinkelsprung durchgeführt. Zum Abgleich werden Links- und Rechtskurven gefahren. Um den Einfluss von Bremsmanövern zu untersuchen, wird im Programm 3 eine Hinterradbremsung durchgeführt und im Programm 4 eine Bremsung vorne.

Bei allen Lenkwinkelsprüngen konnte schon bei den ersten Testfahrten eine deutliche Kippneigung festgestellt werden. Bereits bei v = 10 km/h und einem Lenkwinkelsprung auf der mittleren Kreisbahn hebt das kurveninnere Rad leicht ab. Die Sequenzen in Bild 4 zeigen von links nach rechts das Durchfahren der mittleren Kreisbahn bei einer Geschwindigkeit von v = 15 km/h gefolgt von dem Lenkwinkelsprung. Nur die Stützkonstruktion verhindert das vollständige Kippen.

Abbildung 4: eLastendreirad mit Lenkwinkelsprung und zweifacher Kippneigung

Derartige Ausweichmanöver können auch nach Expertise der professionellen Testfahrerin nicht durch ungeschultes Fahrpersonal bewältigt werden. Eine hohe Beladung kann bei günstiger Schwerpunktlage die Kippneigung verringern oder bei ungünstiger Schwerpunktlage sogar erhöhen. Die erste Forschungsfrage aus der Einleitung ist geklärt – die Kippneigung ist tatsächlich vorhanden und ein kritischer Fahrzeugsicherheitsaspekt, welcher beispielweise durch aktive Fahrwerks-komponenten verhindert werden kann.

3 Messtechnik und fahrdynamische Größen

Zur Auswahl einer geeigneten Messtechnik wurde der aktuell publizierte Stand von Forschungsarbeiten herangezogen. Nach den Arbeiten [Ho00] an einem Dreiradfahrzeug mit PKW-ähnlicher Bereifung sind weitere Forschungstätigkeiten zu dreirädrigen Fahrzeugen vorwiegend im außereuropäischen Raum identifiziert worden.

Von einer indonesischen Forschergruppe wird in [End22] ein sogenannter Rollover Index (RI) abgeleitet und bewertet. Dieser Index baut auf den rechen- oder messtechnisch zu ermittelnden Radaufstandskräften für die drei Räder auf. Seit langem befassen sich Forscher aus einem mexikanischen Team mit der theoretischen und simulatorischen Untersuchung der Kippstabilität von Dreiradfahrzeugen [Va18], [Ro21]. Auch hier wird der Rollover Index betrachtet. Die in diesen Arbeiten untersuchten Fahrzeuge haben Reifen und Räder aus der Automobiltechnik und decken einen höheren Geschwindigkeits- und Lastbereich als die betrachteten eLastendreiräder ab. Zu den Erprobungen im Jahr 2022 waren keine fahrdynamischen Experimente zur Kippstabilität von elektrisch unterstützten dreirädrigen Lastenfahrrädern bekannt.

Eine fahrdynamische Modellbildung und die Auswahl einer geeigneten Messtechnik sind eng miteinander verzahnt. Die Pritsche des Fahrzeugs ist ungefedert und ein Eintauchen der Vorderradgabel beim Lenkwinkelsprung kann vernachlässigt werden. Das Nicken und das Wanken des eLastendreirads wird daher vernachlässigt. Eine gute Näherung der Fahrzeugkinematik stellt daher das sogenannte Einspurmodell [Mit04] dar, welches nach Abb. 5 (unmaßstäblich) für die kinematische Modellierung des eLastendreirades zugrundegelegt wird.

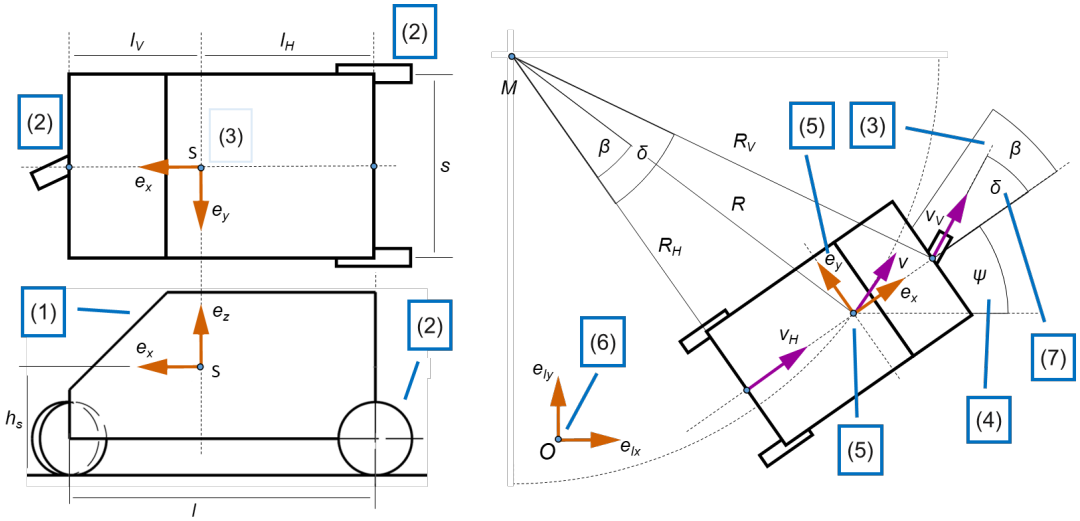


Abbildung 5: Modellbildung des eLastendreirads in Anlehnung an das Einspurmodell [Mit04] und kinematische Größen

Der ungefederte Aufbau mit Fahrer, Pritsche und Last sind durch einen starren Körper (1) angenähert. Die fahrradähnlichen Räder (2) werden als masselos und starr betrachtet. Die Vorderradgabel wird ebenfalls als masselos angenommen. Nach Abb. 5 kennzeichnen β den sogenannten Schwimmwinkel (3) und ψ den Gierwinkel (4) des Fahrzeugkoordinatensystems (5) gegenüber dem Inertialsystem (6). Das Fahrzeugkoordinatensystem hat die Abstände l_v und l_h zu den Achsmittelpunkten vorne und hinten sowie die Schwerpunkthöhe h_S.

In Abb. 5 erkennbar sind zudem der (momentane) Kreismittelpunkt M, die Kurvenradien R_V, R und R_H, die Starrkörpergeschwindigkeit v (7) sowie der Lenkwinkel δ (8).

Am eLastendreirad bleibt nur wenig Bauraum, um Sensorik und Datenerfassung am Fahrzeug zu implementieren. Ein neuer Weg für diese Fahrzeugart und deren Einsatz ist es, das von FahrerInnen mitgeführte Mobiltelefon als Sensoreinheit zu nutzen. Selbst kostengünstige Mobiltelefone als sogenannte Smartphones haben heute eine Vielzahl von Sensoren für kinematische Größen. Eine aus der technisch-wissenschaftlichen Ausbildung bekannte Applikation auf dem Smartphone ist Phyphox [Phy24]. Phyphox sensiert und speichert auf dem Smartphone die Gierrate als erste Ableitung des Gierwinkels ψ, die Bahnbeschleunigungen a_x, a_y und a_z in Bezug auf das geräte- bzw. fahrzeugfeste Koordinatensystem sowie den Betrag der Bahngeschwindigkeit v. Das Smartphone wird daher als Sensorikeinheit genutzt und so in der Nähe des gedachten Fahrzeugschwerpunktes in einer Halterung (1) am Fahrzeugrahmen befestigt, siehe Abb. 6.

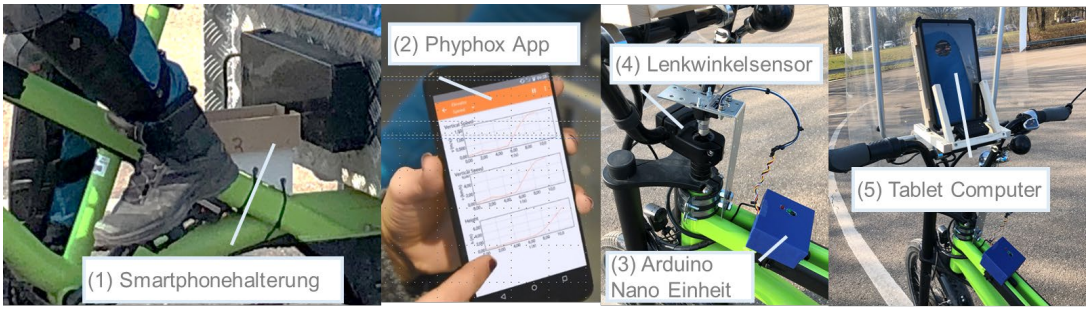


Abbildung 6: Sensorik und Messdatenerfassung am eLastendreirad

Eine Geräteebene des Smartphones ist durch die Halterung (1) zur Fahrebene näherungsweise parallel. Zusammen mit einer sogenannten Physical-Computing-Einheit Arduino [Ard24] wird der Lenkwinkelsensor (4) hard- und und softwaretechnisch integriert. Die Smartphone-App (2) sensiert redundant die vom Arduino gemessene Gierrate sowie die Beschleunigungen. Das Signal der Gierrate liegt damit zweifach vor und kann zur Kompensation eines zeitlichen Signalversatzes von Smartphone und Arduino-Einheit (3) genutzt werden.

Als dritte Sensorikeinheit wird nach Abb. 6 rechts ein am Lenker montiertes Tablet (5) genutzt, welches der Fahrerin die aktuell ermittelte Geschwindigkeit v als Richtgröße anzeigt. Damit kann nun das Testprogramm messtechnisch aufwandsarm unterstützt werden.

Beispielhaft für die Messdatenauswertung ist in Abb. 7 der gemessene und berechnete Verlauf der Gierrate für zwei Lenkwinkelsprünge über der Zeit dargestellt. Deutlich erkennbar ist die näherungsweise Konstantfahrt im Zeitabschnitt von 45 bis 70 s und von 88 bis 103 s. Dazwischen wurden zwei Lenkwinkelsprünge während der Fahrt einbracht und ein Kippen mit Gegenlenkbewegungen kompensiert. In blau dargestellt ist die durch das Smartphone gemessene Gierrate.

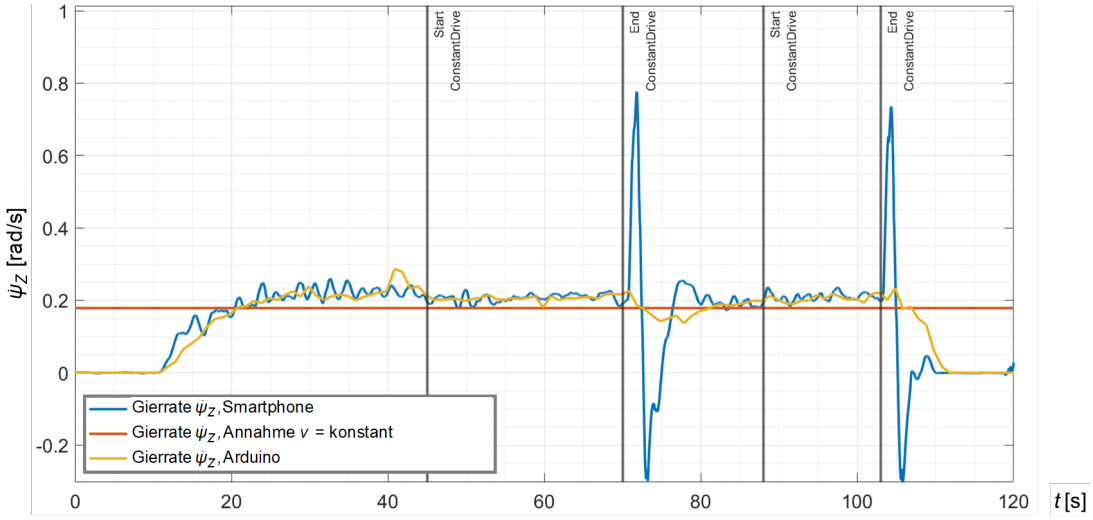


Abbildung 7: Gierrate bei Lenkwinkelsprung, Linkskurve v = 10 km/h, mittlerer Radius

Unerwartet ist, dass es trotz der Einfahrrunde in der Kreisbahn schwer fällt, eine konstante Fahrgeschwindigkeit v einzuhalten. Dies ist zum einen durch das ruckartige Einsetzen des Elektromotorantriebs und zum anderen durch die beobachtete Mittelwertbildung der Tachometer-Applikation (5), Abb. 6, bedingt. Die Übereinstimmung der Messgrößen mit den Approximationen des Einspurmodells aus [Mit04] wird insgesamt als ausreichend erachtet.

4 Zusammenfassung und Ausblick

Das Kippen von Lastenfreirädern bei Ausweichmanövern ist nachweislich ein wichtiger Fahrzeugsicherheitsaspekt. Lenkwinkelsprünge als Testsignal führen zu einem Abheben des jeweils kurveninneren Hinterrades und damit zum Kippen, welches hier nur durch die Stützkonstruktion und eine professionelle Fahrerin beherrscht werden kann. Das für FahrerInnen verfügbare Smartphone ist eine neue Option der Fahrerassistenz. Derzeit unverzichtbar ist ein Lenkwinkelsensor, der als neue Komponente im eLastendreirad konstruktive Maßnahmen am Rahmenkopf bedingt. Vorteilhaft ist eine Bremsung der Räder hinten.

Zur Erarbeitung der Patentschrift [Da23] wurden die kinetischen Gleichungen zum Einspurmodell für eine modellbasierte Ansteuerung von aktiven Sicherheitskomponenten aufgestellt. So könnte im Fahrbetrieb das Kippen sensiert und Gegenmaßnahmen wie ein Akustiksignal, eine automatische Bremsung oder ein Zusatzgewichtsausgleich ausgelöst werden [Da23]. Zusammen mit interessierten Herstellern sollte aufbauend auf diesen Untersuchungen ein sicherer Lastenradbetrieb und damit eine vermehrte Nutzung von klimafreundlichen Fahrzeugen im urbanen Verkehr weiter erforscht werden.

5 Literaturverzeichnis

[DLR24] N. N. https://www.lastenradtest.de/infos-downloads/ , abgerufen am 26.8.2024.

[ZIV23] N. N. ZIV – Die Fahrradindustrie. Marktdaten Fahrräder und E-Bikes für 2023 in Kooperation mit dem VSF – Verbund Service und Fahrrad. https://www.ziv-zweirad.de/wp-content/uploads/2024/03/ZIV_Marktdatenpraesentation_2024_fuer_GJ_2023.pdf , abgerufen am 26.8.2024.

[DIN79] DIN 79010:2020-02, Fahrräder – Transport- und Lastenfahrrad – Anforderungen und Prüfverfahren für ein- und mehrspurige Fahrräder. https://www.din.de/de/mitwirken/normenausschuesse/nasport/wdc-beuth:din21:315466805 , abgerufen am 26.8.2024.

[Da22] Andreas Daberkow, Verena Barske. Fahrdynamik eines elektrisch angetriebenen Lastendreirads. In: Beiträge zum 5. Aalener Kolloquium antriebstechnische Anwendung 2022, S. 113-127. Shaker, 2023.

[Radk24] N. N. https://www.radkutsche.de/musketier/ , abgerufen am 26.8.2024.

[Ho00] Peter Holdmann. Auswirkungen der Kraftübertragungseigenschaften von Personen-und Kraftradreifen auf die Fahrdynamik eines Dreiradfahrzeuges (Dissertation). Forschungsgesellschaft Kraftfahrwesen, 2000.

[End22] Fitri Endrasari et al. Rollover Stability Analysis and Layout Optimization of a Delta E-trike. Automotive Experiences 5(2), 137-149, 2022.

[Va18] Edgar A. Vázquez-Rodríguez et al. The Rollover Risk and its Mitigation in Rickshaws. In 2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (pp. 1-6), 2018.

[Ro21] Martín-Antonio Rodríguez-Licea et al. On the tripped rollovers and lateral skid in three-wheeled vehicles and their mitigation. Vehicles, 3(3), 357-376, 2021.

[Mit04] Manfred Mitschke, Henning Wallentowitz. Dynamik der Kraftfahrzeuge. Springer, 2004.

[Phy24] phyphox - Physical Phone Experiments. https://phyphox.org/de/home-de/, abgerufen am 26.8.2024.

[Ard24] N. N. https://www.arduino.cc/ , abgerufen am 26.8.2024.

[Da23] Andreas Daberkow, Nico Weselin. Stabilisierungssystem und Verfahren zur Stabilisierung eines Lastenrads, Lastenrad, Radanhänger und Computerprogrammprodukt. P 060 0002 DE, DE 10 2021 130 824.0, Patenterteilung am 9.11.2023