International Journal of Transportation Science and Technology 8 (2019) 202–218

Contents lists available at ScienceDirect

International Journal of Transportation Science and Technology

journal homepage: www.elsevier.com/locate/ijtst

Adaptive functional testing for autonomous trucks

M. Elgharbawy [a,c,*], I. Scherhaufer [a], K. Oberhollenzer [b], M. Frey [c], F. Gauterin [c]

[a] Truck Product Engineering, Daimler AG, 70372 Stuttgart, Germany
[b] Autonomous Truck Strategy, Daimler AG, 70372 Stuttgart, Germany
[c] Institute of Vehicle System Technology, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

ARTICLE INFO

Article history:
Received 18 February 2018
Received in revised form 30 September 2018
Accepted 22 November 2018
Available online 2 December 2018

Keywords:
Autonomous trucks
Adaptive functional testing
Data- and knowledge-driven test methods
XiL technologies
Homologation

ABSTRACT

Long-distance trucks are predestined for automated driving due to their high driving performance and long monotonous routes. Automation has the potential to increase road safety, improve fuel efficiency, optimise vehicle utilisation, increase driver productivity and reduce freight costs. Although the widespread use of full automation is not imminent, the vision of accident-free driving accelerates the evolution of driver assistance systems to higher stages of automation on the global market. The status quo assessment refers to functional testing as one of the key challenges for an economical, reliable and safe deployment of autonomous driving in the series development of trucks. Therefore, systems engineering has established data- and knowledge-driven test methods to ensure the required reliability of its products. In this scheme, the evaluation of software releases must be carried out in various phases up to the start of production. Initially through XiL technologies, then through driving simulators, test drives with trained test supervisors on test tracks and public roads, test drives by intended users and finally the homologation of vehicle types. This paper quantifies the conflict of objectives between the requirements of the test concept. Thus, a trade-off between efficiency and effectiveness criteria is achieved through adaptive test coverage of these driving functions in truck product engineering. The basics of the adaptive functional testing are presented, including commonly used verification and validation procedures. The industry-proven framework facilitates the criteria for evaluating the performance of automated driving functions and the measures for achieving a sufficient degree of maturity within the software quality management process.

1. Introduction

Commercial trucks are vehicles designed to create economic value. They are highly specialised in fulfilling specific tasks and are primarily controlled by economic efficiency. In addition, commercial vehicles are characterised by a large number of

Abbreviations: AEBS, Autonomous Emergency Braking System; ACC, Adaptive Cruise Control; AD, Automated Driving; BSA, Blind Spot Assist; CAN, Controller Area Network; CPVS, Cyber-Physical Vehicle System; CoP, Code of Practice; DESTATIS, German Federal Statistical Office; DuT, Device under Test; ECU, Electronic Control Unit; E-Horizon, Electronic Horizon; E/E, Electric/Electronic; FMI, Functional Mockup Interface; FuSa, Functional Safety; GSN, Goal Structuring Notation; HiL, Hardware-in-the-Loop; N-FOTs, Naturalistic-Field Operational Tests; MC/DC, Modified Condition/Decision Coverage; MiL, Model in the Loop; MTBF, Mean Time Between Failures; MV, Monocular Vision; ROI, Region Of Interest; SiL, Software in the Loop; SOTIF, Safety Of the Intended Functionality; SoP, Start of Production; TSR, Traffic Sign Recognition; PoS, Proof of Safety; UNGA, United Nations General Assembly; UNECE, United Nations Economic Commission for Europe; XiL, Something (X)-in-the-Loop.

* Corresponding author.
E-mail address: mohamed.elgharbawy@daimler.com (M. Elgharbawy).

https://doi.org/10.1016/j.ijtst.2018.11.003
2046-0430/

Nomenclature

Variables

a^x_{ego}	absolute longitudinal acceleration of the ego-vehicle;
A_i	algorithm version with incorrect detection of the oncoming vehicle;
A_{i+1}	algorithm version with correct detection of the oncoming vehicle;
A_j	algorithm version with lane departure functionality;
B_i	recorded video sequence with incorrect detection of the oncoming vehicle before regression test;
B_{i+1}	recorded video sequence with correct detection of the oncoming vehicle after regression test;
C	confidence level of the travel distance for providing proof of safety;
C1	context identifier for defining AD stages with respect to their required safety integrity;
C2	context identifier for defining functional requirements to be fulfilled by the AD function;
C3	context identifier for defining system context including its quality gate within the product engineering;
C4	context identifier for defining effectivity and efficiency criteria of the scenario-based test concept requirements;
C5	context identifier for defining field-based observations using clustering of multivariate time series analysis;
C6	context identifier for defining acceptable pass/fail criteria using criticality matrix;
d_t	travel distance for providing proof of safety;
d^l_y	lateral distance to the left lane;
E1	solution identifier for functional requirements coverage;
E2	solution identifier for software structural coverage;
E3	solution identifier for system integration and variation coverage;
E4	solution identifier for system performance coverage;
E5	solution identifier for training data and uncertainty coverage;
E6	solution identifier for driving scenario coverage;
G1	goal identifier to accept residual risk associated with individual hazards of an AD function;
G2	sub-goal identifier to prove the functional correctness using Cluster-HiL and vehicle tests;
G3	sub-goal identifier to prove the back-to-back consistency of algorithmic-based software structures using back-to-back(MiL/SiL) tests;
G4	sub-goal identifier to prove the system integration and variation using System HiL tests and proving grounds;
G5	sub-goal identifier to prove the software robustness using fault injection techniques;
G6	sub-goal identifier to prove sensor availability and functional effectiveness using regression tests and big-data re-simulations;
G7	sub-goal identifier to prove software reliability and functional safety using field-based observations with N-FOTs;
γ	system reliability of an automated heavy-duty truck;
$lane^r_{ref}$	reference right lane from ground truth simulation data;
$lane^r_{ctb}$	measured right lane at the Cluster-HiL test bench;
$lane^l_{ref}$	reference left lane from ground truth simulation data;
$lane^l_{ctb}$	measured left lane at the Cluster-HiL test bench;
λ	vehicle failure rate;
λ_A	failure rate of AD function without driver supervision;
λ_H	failure rate of AD function with human driver supervision;
Λ	fatality rate;
m	number of failures during travelled distance;
M_k	model version before code generation for back-to-back testing;
m_{req}	requested mode;
S1	strategy identifier for the required sucessful test completion through adaptive test coverage;
S_k	software version after code-generation for back-to-back testing;
t	time;
ϱ	estimated distance at which a critical traffic event occurs at a specified confidence level;
υ^l_y	lateral velocity to the left lane;
ζ	failure event of AD function;

series and models with tractors, semi-trailers or trailer combinations (Trigell et al., 2017). In most nations, the legislation regulates the concepts and functions of commercial vehicles up to a specific vehicle system. The current challenges are to improve the utilisation of existing infrastructure, enhance the use and combination of assistance functions and make the truck driver profession more attractive.

1.1. Motivation

Road accidents cause almost 1.3 million deaths and 20–50 million injuries every year (Asirt, 2018). Therefore, the UNGA has launched a decade of road safety measures between 2011 and 2020 to reduce the risk of road accidents and injuries. On the one hand, freight traffic continues to increase globally and is the dominant means of transport. According to the traffic forecast for 2030, road freight transport performance in Germany will increase by 38% compared to the level of 2010. On the other hand, truck accidents often have serious consequences such as injury and death, as well as considerable financial impacts and environmental risks. Between 1992 and 2014 the number of truck accidents involving seriously injured road users has fallen by more than 45.8%. While the volume of truck traffic increased by 85.3% over the same period, the number of people who died in these accidents fell by more than 59.7%, as shown in Fig. 1.

The evolution of automation in civil aviation meeting the increased safety requirements can be considered as an indication of the challenges of the same expansion in trucking. Both sectors focus on goods and passenger transport in a scalable environment. According to statistical studies on the number of accidents in civil aviation, there has been a significant decline in the global accident rate, despite the increased number of aircrafts. The life cycle statistics for each aircraft generation show that the lowest first-generation accident rate was around 3.0 accidents per million flights. The second generation reduced the rate to 0.7 accidents per million flights corresponding to 80% reduction in fatal accidents. On the other hand, third-generation aircrafts achieve about 0.2 accidents per million flights. The number of accidents in the fourth-generation jets is the lowest, with a stable average of about 0.1 fatal accidents per million flights, as illustrated in the Fig. 2. Dashed lines indicate accident data with less than 1.0 million flight cycles per year.

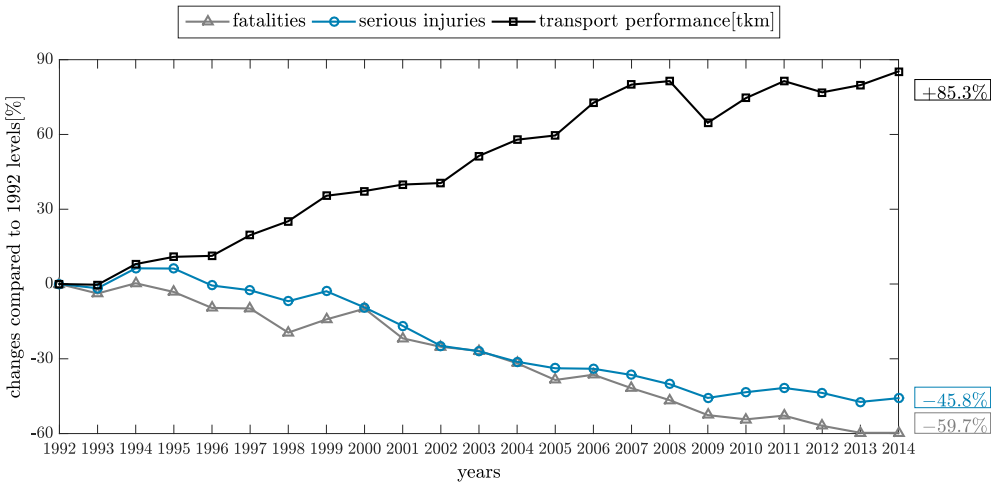

Fig. 1. Fatalities and seriously injured persons in truck accidents on German roads compared to truck transport performance between 1992 and 2014 (Destatis, 2015).

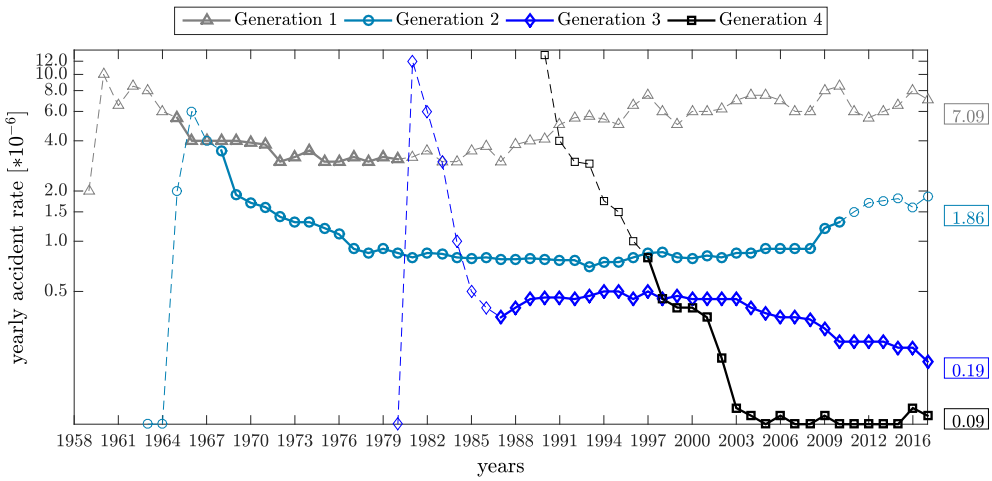

Fig. 2. Airbus statistical analysis of commercial aviation accidents per million flight departures between 1958 and 2016 (Airbus, 2016).

1.2. Problem definition

The conversion from driver assistance systems of levels 0, 1 and 2 to higher levels of automation in accordance with SAE J3016 represents a new challenge for the type-approval of automated commercial trucks. The main difference is that driver assistance can have unintended interventions, where the driver can override these interventions at any time if functional limitations appear. Their functions are therefore designed to be controllable, but this can reduce their benefits. The controllability of system interventions and the effectiveness in the field with minimal undesired consequences are therefore decisive for the series development of these driving functions (Winner et al., 2018). Moreover, long-haul commercial vehicles are heavier, larger and less maneuverable than passenger cars. Commercial truck characteristics (e.g. dimension, low-speed transient off-tracking, braking distance, type variability, etc.) pose therefore new challenges for automated driving functions. Accordingly, systems engineering requires state-of-the-art evaluation procedures to verify and validate these systems. N-FOTs are carried out to define thresholds for intervening systems, based on the collected data. On the one hand, trigger algorithms can be optimised to minimise the frequency and impact of falsely triggered interventions and, on the other hand, to maximise the number of legitimate responses. Nevertheless, AD requires the system to exploit the limits of dynamic driving tasks and to master most environmental conditions controlled by a human driver (Schöner, 2016). The ISO 26262:2018 standard extends the functional safety regulations of E/E systems for heavy-duty commercial vehicles. However, the safety standard is limited to avoiding potentially safety-critical situations caused by systematic software and random hardware failures. Safety violations due to technological and system-technical deficiencies remain outside the scope of ISO 26262:2018 (e.g. insufficient robustness, uncertainty issues with perception sensors, etc.) (Burton et al., 2017). In particular, AD without driver monitoring can also lead to potentially safety-critical situations resulting from deficiencies in the estimation, interpretation and perception processes. While there are, at present, no generally accepted test procedures that enable AD functions to be validated with affordable efforts, ongoing research projects (e.g. PEGASUS, L3Pilot, ENABLE-S3, SafeMove, TAF BW, etc.) show the relevance of research for new test methods. For this reason, the primary question is: "How can automated driving functions be verified efficiently and effectively to achieve the required test completion criteria?"

1.3. Contribution

The statistical analysis of road accidents predicts the required mileage for levels of automation without driver involvement as a basis for the safety of new systems compared to their predecessors. These technologies face an unsolved challenge when it comes to proving safety during the development phase by means of field operational tests. While the uncertainties of machine learning remain before automated driving is released for widespread use, it is essential to develop performance assessments for the safety record. Furthermore, highly automated test approaches are integrated to verify the reliability of the automated driving software. Black box, grey box and white box tests are combined with their respective test objectives and form the basis for an adaptive verification concept. The proposed procedure offers an optimised test strategy for the systematic extension of the requirement-based test coverage resting upon a modular verification framework with continuous knowledge enhancement from field observations. Using an ontology-based method, a category of adequate and relevant logical scenarios for existing field tests is extracted. A semantic representation of concrete scenarios can be obtained using data mining techniques, and systematically processed in executable requirements for adaptive test coverage.

1.4. Structure of the paper

The paper is structured as follows: Section 2 gives a brief overview of the evolutionary stages of automated driving and their safety requirements. Section 3 provides a literature overview and compares environmental perception and situation prediction sensors. The different stages of automated truck driving and their safety requirements are classified in Section 4. Moreover, Section 5 provides a general overview of the main challenges in assessing software reliability and function safety for automated truck driving and elaborates on the proposed methodology and the implementation details of the framework. Furthermore, Section 6 presents quantitative results and evaluates the capability of the entire system. The paper concludes with a summary in Section 7.

2. Towards autonomous trucks

Automated driving functions are software components that interact with the real traffic environment to support or automate human driver tasks. Given the high mileage and long monotonous distances of long-distance trucks, various business cases of cooperative automated driving are demonstrated, such as truck platooning. Despite the fact that the widespread use of full automation is not imminent, the vision of accident-free driving expedites the further development of driver assistance functions to evolutionary autonomy stages on the global market. These stages are expected to overlap and are not sequentially available on the market. In spite of strong support from industry and academia, questions about their business cases, ethical dilemmas, legal liability and safety are frequently asked. For example, it is necessary to further adapt the vienna convention on road traffic to provide an automated steering system, which is prohibited in the UNECE R797 for use above 10 km/

h (Kirschbaum, 2015). The sense-plan-act robot control method provides a functional view of the data flow in the sensor and control system of an automated heavy-duty truck. Parts of the environment perception and model are responsible for recognising the truck's environment and the associated situation awareness. While the planning part is responsible for determining the driving trajectory, the motion control part implements this plan (Smith, 2017). Therefore, equipped with automated driving, a commercial truck can be identified as a CPVS whose driving functions enable the intelligent handling of dynamic traffic situations in an extremely safety-critical environment. Automated driving can be represented as a composition of three evolutionary elements (simple scenarios, low speed, and high-risk situations), as shown in Fig. 3 (Winner, 2015).

The AEBS is an active safety system for high-risk situations, which uses RADAR and vision sensor systems to monitor and detect the proximity of the preceding vehicles and pedestrians. The system warns the driver by a combination of optical, acoustic or haptic signals. The system automatically determines the time required to perform the warning cascade and emergency braking in this situation to prevent the collision. The ACC is categorised as an intervention system for simple scenarios that maintains a set cruise control speed, unless the sensor systems detect a slower vehicle ahead. The ACC identifies the relevant preceding vehicles and calculates both the deceleration as well as the possible acceleration required to maintain a safe distance. The BSA assists the truck driver in turning at low speeds when an object is laterally next to the heavy-duty truck or when the visibility is restricted by the length of the heavy-duty truck or poor weather conditions. Each automation stage has a value of its own and the potential to create economic benefit. However, automated commercial vehicles does not require only comprehensive safeguarding against realistic driving scenarios but also dealing with uncertainties. Though a human driver is not perfect when learning to drive, every human driver is adaptive to create its predictive mental model from years of driving experience. One of the safety skills is that the human driver has enough self-awareness to recognize an unclear driving situation and to minimize the risk until the uncertainty is eliminated. Therefore, continuous supervision and learning from field observations help in coping with rare and dangerous events. As a result, understanding the system's own limits is essential when dealing with unknowable black swans. Table 1 illustrates the safety integrity requirements for each stage of automation. The evolutionary stages of automated driving can be summarised as follows (Damm and Heidl, 2017):

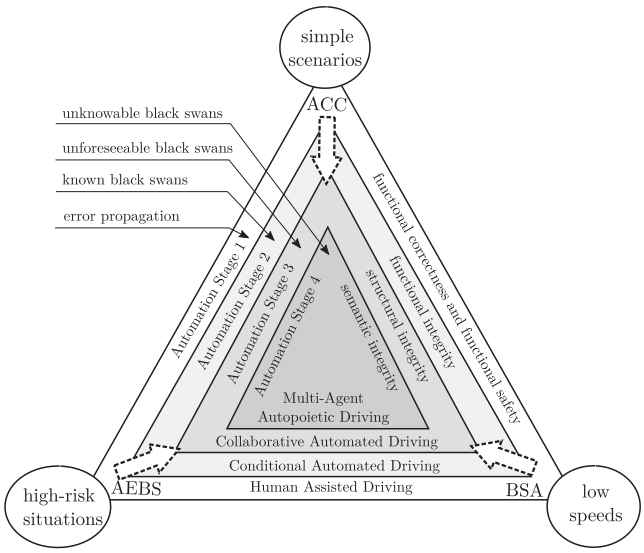

Fig. 3. Evolutionary triangle of automated truck driving using the sense-plan-act control methodology.

Table 1
Safety integrity requirements for each automation stage adapted from SAE automation standards (J3016).

Stage	Description	Responsibility			Safety requirements
		Execution of steering and acceleration/deceleration	Monitoring of driving environment	Fallback performance of dynamic driving task	
1	Human Assisted Driving	Human driver and\or System	Human driver	Human driver	Functional safety
2	Conditional Automated Driving	System	System	Human driver	Functional integrity
3	Collaborative Automated Driving	System	System	System	Structural integrity
4	Multi-Agent Autopoietic Driving	System	System	System	Semantic integrity

2.1. Automation Stage 1

Human Assisted Driving handles tasks of limited complexity autonomously in a precisely specified context and has three sub-categories. Functional information and warning systems refer to the first sub-category in which the driver is fully engaged (e.g. TSR, LDWS, BSA, etc.). The second sub-category focuses on the functional intervention and assistance systems (e.g. AEBS, ACC, etc.). The third sub-category relates to combinations of functions and multi-interacting driver assistance systems. All these systems perform limited tasks in a defined context and do not learn during operation. Since the collaboration is a restricted task in a determined context, cooperation is therefore limited to the exchange of information on the system context. The safety integrity of HAD is aimed at ensuring the functional correctness and safety of the system. Critical driving situations due to systematic software and random hardware failures can be handled within the ISO 26262:2018 standard. The SOTIF regulates the absence of unreasonable risks from hazards due to performance limitations and insufficient awareness of the driving situations.

2.2. Automation Stage 2

Conditional Automated Driving accomplishes a sequence of tasks, in which every single task is controllable, but whose sequence and transitions between them are situation-dependent. While the system is not learning during operation, it optimises its trajectories during the control process according to defined objectives such as time or other resources. The cooperation with other systems is therefore limited to the exchange of information about the system context and the system itself. The safety integrity of conditional automated driving shall ensure functional integrity to deal with known black swan events in fail-operational mode using cautionary and precautionary risk management principles.

2.3. Automation Stage 3

Collaborative Automated Driving is able to work together with other systems to perform their task. They negotiate their goals, plans and actions with other systems and adapt their behaviour to the negotiated procedure. Since the system boundaries change dynamically due to the collaborative relationship, mechanisms for distributed planning and coordination of interpretations are required to ensure safe system functionality. Beyond the need to follow accepted safety engineering practices, collaborative automated driving focuses on structural integrity to ensure reasonable behaviour against unforeseeable black swans using improved risk assessment and knowledge discovery principles.

2.4. Automation Stage 4

Multi-Agent Autopoietic Driving can expand the environmental perception, situational awareness and actions with the ability of unsupervised learning and has some sort of fail-operational autonomy capability. The semantic integrity shall be required for the autopoietic driving to safeguard a possible online expansion. The semantic integrity demonstrates the self-recognition and dealing with unknowable black swans by defining the reasonable behaviour. Therefore, the system needs to be good enough to recognise surprises and ensure that the behaviour remains relatively modest until the uncertainty is resolved.

3. Deficiencies in environmental perception

Since automated driving depends on the perception of the truck's environment, safety violations can be caused by system restrictions due to physical or technical limitations of the intended functionality. Furthermore, object recognition and classification tasks are performed by machine learning techniques to extract relevant characteristics in an unstructured operational context. While machine learning paradigms offer a promising perception performance, high levels of false positive and false negative rates can decisively influence the functional safety of the overall system. Therefore, the performance evaluation of the environment perception should be defined to ensure a sufficiently safe level of residual risk associated with functional deficiencies in machine learning algorithms. Hence, the various sensors must be verified not only concerning their failure rates, but also about possible causes of technical shortcomings in machine learning. Consequently, the quantitative evaluation of perception sensors and algorithms should consist of false positive and false negative rates, in which some assumptions about the system context are implied. Therefore, the robustness in real traffic can be achieved by the creative fusion of sensor data as well as appropriate system design. Range Sensors such as LiDAR, RADAR calculate distance, angle and signal power to detect targets in a particular ROI. Automotive RADAR sensors observe the position and velocity of moving objects as well as stationary roadside objects with precise range information and high resistance to poor weather conditions. However, RADAR detection is afflicted with limited angular resolution in the case of stationary or longitudinally moving pedestrians. Today's automotive cameras play a vital role in environment perception due to the high information density present in images. Computer vision methods are based on two-dimensional scene to detect and classify objects in the frontal environment as well as to track the lane markings. The visual processing system provides a relatively high angular resolution, which would improve the classification performance of the pedestrian detection. However, its drawbacks are, similar to

automotive LiDAR, weather sensitivity and limited detection range. Since today's standard vehicle sensor measurements contain relatively limited information content, the E-Horizon data serves as a predictive sensor to anticipate the driving path. Fig. 4 refers to a comparable evaluation with scale ranges from poor to optimal using the following twelve commercial vehicle sensor capability criteria: maximum longitudinal range, lateral field of view, longitudinal range accuracy, relative object speed perception, moving object dimension, moving object classification, poor weather conditions, behaviour at darkness, sensor installation flexibility, sensor cost requirements and road classification. The perception and map-based prediction sensors have different measurement principles, which will be classified, as follows, into five sensor types.

Object recognition and classification are primarily performed by machine learning techniques to extract relevant features in an unstructured operational context. Although machine learning paradigms offer promising perceptual performance, high values of false-negative and false-positive rates can have critical safety consequences within the overall system. Therefore, the performance evaluation of the environment sensing should be established to provide a sufficiently safe level of residual risk associated with functional deficiencies in machine learning based functions. Accordingly, the different sensors have to

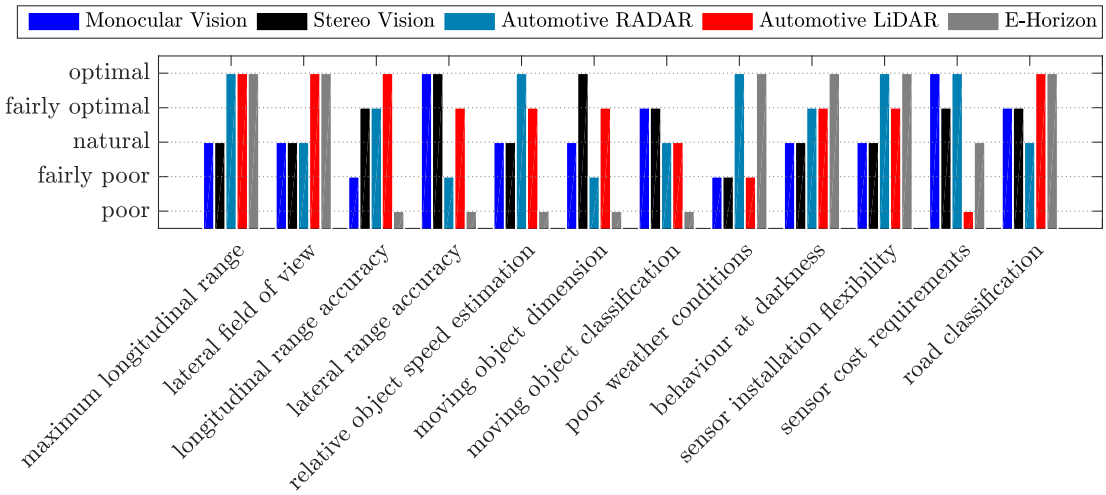

Fig. 4. Comparable evaluation of environmental perception and situation prediction sensors Steinmeyer, 2014.

Table 2
Potential causes of uncertainty within the environmental perception and situation prediction sensors.

Sensor Type	Low Specificity(False Positives)	Low Sensitivity(False Negatives)
Monocular Vision	- incorrect object hypotheses (e.g. ghost objects, bright lights etc.) - underexposed backgrounds (e.g. color distortion, etc.)	- poorly illuminated objects - no pattern matching within the training dataset - overexposed backgrounds (e.g. due to direct sunlight exposure, etc.) - during bad weather conditions (fog, rain, snowfall, etc.)
Stereo Vision	- ambiguous disparity matching with repetitive patterns - underexposed backgrounds (e.g. color distortion, etc.)	- poorly illuminated objects - no pattern matching within the set of training data - low disparity by homogenous object surfaces - low-height objects(no plane separating) - overexposed backgrounds (e.g. due to direct sunlight exposure, etc.) - during bad weather conditions (fog, rain, snowfall, etc.)
Automotive LiDAR	- large sensor pitching motion - large road gradient	- light-absorbing objects - planar surface objects - during bad weather conditions (fog, rain, snowfall, etc.)
Automotive RADAR	- underdrivable metalic objects (e.g. road sign gantries, road bridges and overpasses, tunnel fans, corrugated metal sheets, etc.) - driven-over metalic objects (e.g. guard rails, movable manhole covers, drive-over drinks can, etc.) -ambiguities of object classification (e.g. due to alleys situations) - higher deceleration time due to truck kinematic	- objects with low radar cross-section - aging influenced radome behind the bumper
Electronic Horizon	- discrepancies between the GPS positioning data and matched maps	- not updated map data present in memory (e.g. speed limits on construction sites)

be verified not only regarding the failure rates but also for possible causes of technical shortcomings in machine learning. The potential causes for uncertainty for different sensors are classified, as seen in Table 2.

4. Safety dilemmas of automated truck driving

The validation process begins with the selection of a validation target which is calculated by the system use case, the crash statistics and a safety margin. In 2015, the DESTATIS recorded a total of 29,480 accidents involving personal injury with the participation of at least one heavy-duty truck in Germany. In spite of the accident variety with heavy-duty trucks, the statistics show that rear collisions and unintended lane departures are the common types of commercial vehicle accidents with 68% of 32,500 truck drivers (Destatis, 2015). For a particular driving function, human drivers encounter an average number of kilometres between events. The stopping rule assumes that the failure rate has a Binomial distribution. It can be shown that the system has a failure rate greater than or equal to the benchmark reference with a specified confidence level. Therefore, the validation distance required to provide statistical PoS of an automated driving system can be calculated by a benchmark reference for the expected interval between accidents of respective severity.

4.1. Statistical proof of safety

The total fatality rate in Germany caused by heavy-duty trucks in 2015 was 787 fatalities, totalling 58.93 billion kilometres. According to the Binomial distribution, the reliability γ of an automated heavy-duty truck with m failures during the travel distance d_t at confidence level C is:

$$C_{(\zeta=m)} = 1 - \sum_{\zeta=0}^{m} \frac{d_t!}{\zeta!(d_t-\zeta)!} \lambda^{\zeta}(1-\lambda)^{d_t-\zeta} \tag{1}$$

If the failure rate of a truck is λ, then the reliability γ is $(1-\lambda)$ and can be interpreted as the probability that there is no failure in the route driven. A hypothesis about the scenario "no failures driving" can be used to estimate a lower limit for the number of failure-free kilometres to determine the reliability of automated trucks with a confidence level C. Consequently, the safety can be claimed for a certain number of failure-free kilometres at a particular confidence level.

$$C_{(\zeta=0)} = 1 - (1-\lambda)^{d_t} \tag{2}$$

The required test distance d_t without failures is determined for given confidence C and reliability γ, as represented in Eq. (3).

$$d_t = \frac{ln(1 - C_{(\zeta=0)})}{ln(1-\lambda)} \tag{3}$$

substituting λ with $\frac{787}{58.934 * 10^9} = 1.34 * 10^{-8}$ and confidence level C with 95%.

$$d_t = \frac{ln(1-0.95)}{ln\left(1-\left(1.34*10^{-8}\right)\right)} \approx 220*10^6 \tag{4}$$

Hence, the required test distance d_t is approximately 220 million km. Fig. 5 represents the failure rate factor $(\lambda_A \div \lambda_H)$, where λ_A is the failure rate of an automated driving system and λ_H is the benchmark failure rate of a human driver. For today's trucks, there is no necessity for such long validation distances, at which the controllability of the driver provides the neces-

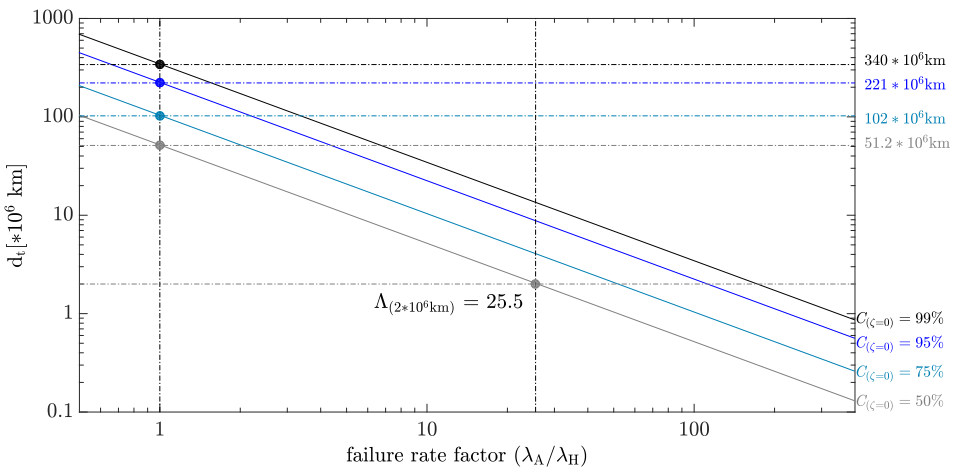

Fig. 5. Failure free kilometers for a failure rate factor compared to human-driven truck fatality rate of year 2015.

sary proof of safety. Nevertheless, the 2 million kilometres used to validate current driver assistance systems are sufficient to prove a fatality rate of $(\Lambda_{(2\ *\ 10^6 km)} = 25.5)$ times that of humans with 50% confidence, in case of a fully automated truck. To prove that an automated truck has a failure rate similar to that of humans in 2015 as a benchmark failure rate and assuming that the truck has no failure $(m = 0)$ during endurance testing, with 99% confidence approximately 340 million kilometres are needed. This analysis applies to failure-free kilometres. For this reason, it is economically impossible to demonstrate the safety of automated driving systems with widespread usage statistically before introduction, defined as an approval trap.

4.2. Distance between critical events

While critical traffic events are typically rare and not reproducible, early identification of functional deficiencies is essential for automated driving. Despite the difficulty of predicting a priori all possible operating scenarios, the coverage of critical driving scenarios needs to be adequately investigated. Recent research suggests the hypothesis of Poisson distribution to calculate the required validation distance with the following assumptions. The Poisson distribution presents discrete probability distribution that expresses the probability of a given number of events in a continuum of time or space. On the one hand, the route used is representative; on the second hand, critical events occur independently of each other within a random process. In the Eq. (5), m corresponds to the number of accident events and ϱ is the predicted distance at which this event occurs at a given confidence level.

$$C_{(\zeta=m)} = \frac{\varrho^{\zeta}}{\zeta!} e^{-\varrho};\ \zeta = 0, 1, 2, \ldots, m \tag{5}$$

The MTBF can be determined at a given confidence level using the hypothesis of the Chi-square distribution according to ISO 26262:2018. The Chi-square distribution presents the probability density function that calcluates the MTBF failure rate based on observed failures. Accordingly, an exponential failure distribution with a constant failure rate is assumed. Regarding the safeguarding of driver assistance systems, there are no legal requirements for the validation distance. Since unintended reactions are rare events, a Chi-square distribution can be applied. If no critical event occurs at a sample distance with a required failure rate of one million kilometres each, the necessary validation requires around three million kilometres. In this case, no event should occur during the driven interval to argue the residual risk with a confidence level of 95%. The required mileage will increase if more events occur during validation (e.g. $d_{t(\zeta=1)} = 4.8\ *\ 10^6$ km, $d_{t(\zeta=2)} = 6.3\ *\ 10^6$ km, $d_{t(\zeta=3)} = 7.8\ *\ 10^6$ km, etc.), as illustrated in Fig. 6. In practice, the validation distance does not play the central role, but the variance of test conditions as much as possible (e.g. different weather conditions, time of day, road conditions, traffic conditions, pedestrian conditions, etc.) to cover rare operating situations. Therefore, route diversity in physical road tests is a significant measure of the probability distribution. However, statistical evidence of the accumulated road kilometres

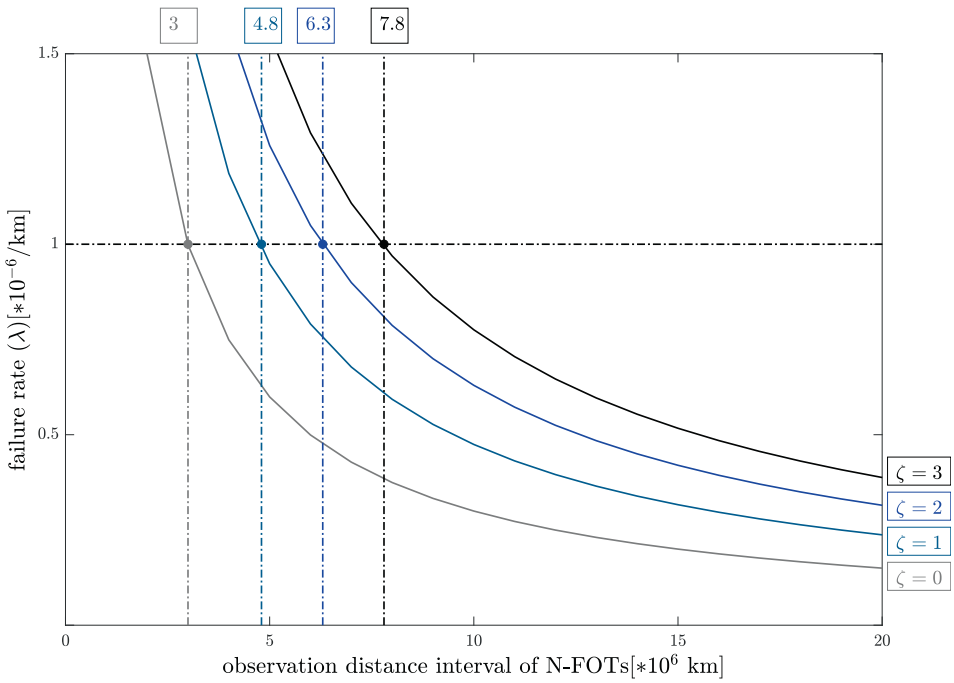

Fig. 6. Required validation distance for various accident events using the Chi-square distribution with confidence level = 95%.

is potentially invalid with each software upgrade. Even if the N-FOTs continue in the spiral model of software development until no more errors are found, the safety case argument does not provide any proof that the automated driving function is absolutely safe due to the Pesticide paradox phenomenon. The Pesticide paradox presents an error detection phenomenon in software testing, if the same test matrix is repeated over and over again, eventually the same test matrix finds no more errors. It means that an automated driving algorithm that passes the same repetitive tests eventually builds up resistance to them. Consequently, it will not be viable to prove safety of the required level of system performance through driving test hours alone during the development phase. Furthermore, there is no complete, public set of machine-interpretable traffic regulation with exception-handling rules. In particular, learning-process results from machine learning data sets are used to perform algorithmic operations, whereby the functional requirements for automated driving functions are implicit and incomplete. On the one hand, the reliance on data-driven mileage accumulation as the only credible safety argument points to an impractical safety validation strategy. In addition, the real-world testing may not accumulate enough hours of exposure to observe critical scenarios that occur by chance. On the other hand, knowledge-driven assessments can accelerate exposure to known critical scenarios but suffer from the possibility of not verifying unanticipated scenarios. Alternative methods of safety assessment are therefore required, as the validation distance in N-FOTs will increase dramatically by using the current test concepts for automated driving without driver engagement.

5. Methodology

The V-model is a software development model that combines requirements and design on the left side with verification and validation on the right side. Although ISO 26262:2018 and its V-framework reflect generally accepted practices for ensuring automotive safety, AD functions present unique challenges in mapping the technical and functional requirements to the V-approach. Therefore, the validation procedures offer a range of activities to generate confidence that a function can achieve its intended purpose and goals.

5.1. Big testing database

The meta-level learning process identify and cluster specific scenarios within bigger data sets to learn from the experiences of the systems in the field. Fig. 7 presents the structure of a comprehensive database for driving scenarios from different sources (e.g. N-FOTs, Cluster-HiL tests, etc.), which are stored in a consistent de-facto standard format.

Big data tools, architectures, and analytics provide the measurement system and database infrastructure for recording, storing, accessing and processing data of big data re-simulations. For an objective assessment of the driving function used, suitable evaluation criteria are integrated into the database. Triggering events serve as a supplementary source of information, to identify relevant driving scenarios occurring in the N-FOTs. The first step comprises the transfer of recordings of triggering events to the big data server platform. The second step is to extract and cluster the multivariate time-series data to provide it as a complementary source of relevant driving scenarios for different other test environments. Time series analysis can be divided into three sub-categories (clustering, regression and classification). Clustering divides triggering events into one of several categorical clusters. The regression then fits each group with the corresponding signal prototype. Ontology-

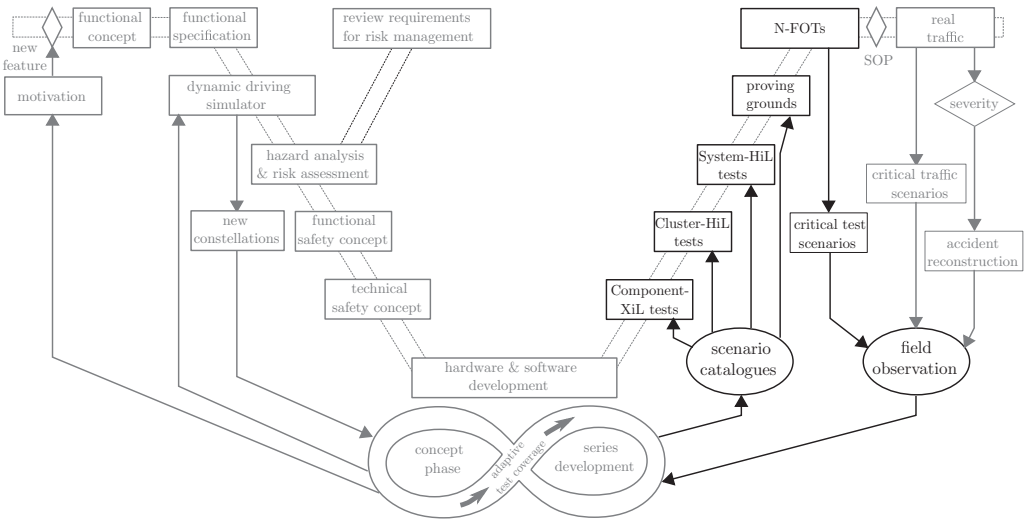

Fig. 7. Meta-level learning process using the V-model and safety evaluation methods using a big testing database.

based transformation rules provide the syntheses of relevant driving scenario based on the prototypes for the Cluster-HiL scenarios. In case of further induced traffic situations, the classification maps them into one of the predefined categorical classes. Subsequently, AD functions can be verified effectively and efficiently through an adaptive test coverage using test oracles based on envelope components of pass and fail criteria. On the one side, the approach of safety validation for AD functions beyond the mileage accumulation is in high demand. Thereby, a falsification approach shall be coupled with concrete, verifiable safety objectives and requirements. On the other side, the verification procedures according to ISO 26262 V-model assume that high-quality requirements for implementation are further developed. Therefore, the traditional V-model engineering process can pose a challenge in articulating the functional requirements of machine learning algorithms. With the V-model, the training set is more related to the functional requirements and the validation set to a test plan. The verification arguments with sufficient training and validation data leads to the need to develop the data ingestion system according to safety-critical software standards.

5.2. Test completion criteria

The GSN presents an assurance case to highlight the verification methods for automated truck driving. The GSN is a graphical notation for structuring an assurance case in connection with argument, context, assumptions and evidence. The assurance case is a reasoned, verifiable artifact that supports the contention that its top level claim is satisfied, including systematic reasoning, its underlying evidence, and explicit assumptions that support the claim according to ISO/IEC 15026-2:2011. Therefore, the assurance case ensures that sufficient evidence is systematiscally gathered to argue a tolerable residual risk through adaptive verification for AD functions, as illustrated in Fig. 8. Each hypothesis identifies the residual risks for a test or simulation environment. The assumptions that are covered by other verification approaches are part of the safety argumentation chain. The `G1` is the main goal to constitute the top-level claim of the verification case scope with the context elements {`C1` ,···, `C4` }. Furthermore, the `G1` argues a sufficiently low level of residual risk associated with individual hazards in AD, since all possible driving situations might not be verified during the development phase. Therefore, the `S1` describes the verification case strategy to define the required successful test completion through adaptive test coverage with the context elements {`C5` , `C6` }. Subsequently, the following sub-goals {`G2`,···, `G7`} provide the evidence arguments {`E1`,···, `E6`} of test coverage within the verification case.

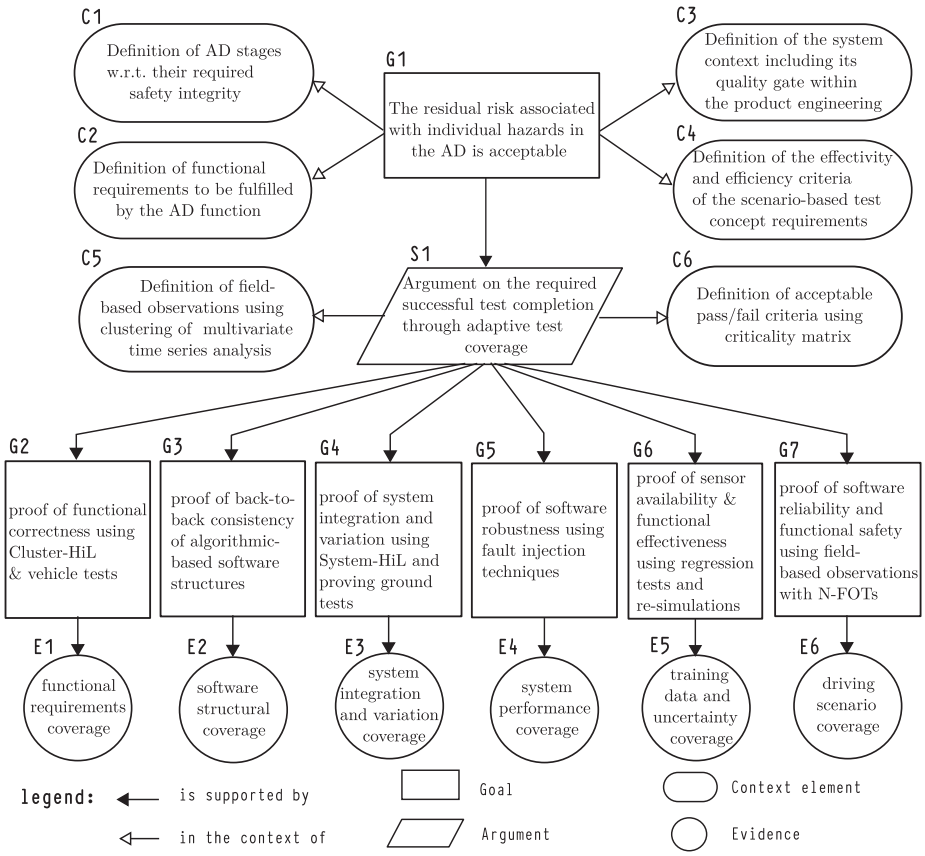

Fig. 8. Adaptive test coverage argument using goal structuring notation.

5.3. Adaptive scenario-based test concept

The scenario-based test concept follows a decomposition approach on functional, logical and concrete levels. Therefore, test scenarios can be described either as functional with natural language description without values, logical with an assignment of value ranges or concrete with an association of fixed values. A major challenge in achieving the decomposition approach is to classify the test objectives and coverage criteria according to their respective test environments. Therefore, the test objectives imply measurable quality criteria for the verification and validation strategy. The selection of suitable test environments from XiL to N-FOTs depends on the effectiveness and efficiency criteria of the test conditions and their validity. The effectiveness criteria indicate the intended results such as representative valid and observable interfaces. However, the efficiency criteria reflect the desired performance in comparison with the resources used to achieve economic use, reproducibility and promptness. Table 3 explains the assignment of different possible test environment with the respective test objectives. The proof of functional correctness verifies that the test object fulfils the required functionality according to its specifications. The proof of functional safety refers to the FuSa requirements of ISO 26262 in order to avoid systematic software and random hardware failures. The proof of functional controllability shall provide a sufficient probability for coping with system boundaries and failures. The proof of the software's reliability is the proof that the test object is robust enough with respect to functionality and safety mechanisms. The proof of back-to-back consistency is the verification of the required consistency between the various execution platforms (e.g. model and code) within the permissible discrepancies by back-to-back tests. The proof of sensor availability focuses on the presence of the environment perception sensor within the defined deviation and tolerance limits of the specified time and range. The test coverage criteria provide an indicator of the software testing effort during a test run within the verification and validation strategy. Table 4 explains the assignment of different possible test suites with the respective test coverage criteria. The functional requirement coverage defines a relationship between functional requirements and executed test cases, whereby at least one test case is defined for each requirement. The software structure coverage provides the code coverage of model-based software structure components, such as MC/DC code coverage. The system integration coverage includes detected failures in the interfaces and interactions between integrated components, subsystems or systems. The system variation coverage defines the robustness against variations

Table 3
Assignment of potential test environments to the corresponding test objectives.

Test environments	Test objectives						
	Proof of functional correctness	Proof of functional safety	Proof of functional controllability	Proof of software reliability	Proof of back-to-back consistency	Proof of functional effectiveness	Proof of sensor availability
MiL tests	●	●	○	○	●	○	○
SiL tests	●	●	○	○	●	○	○
Component-HiL tests	●	●	○	●	○	○	●
Cluster-HiL tests	●	●	○	●	○	○	●
System-HiL tests	●	○	○	○	○	○	○
Vehicle and proving ground tests	●	●	●	●	○	○	●
Driving simulator tests	○	○	●	○	○	○	○
N-FOTs	○	●	●	●	○	●	●
Regression tests/ big data re-simulations	●	●	○	●	●	●	○

●: Recommended test objective.
○: Not useful.

Table 4
Assignment of possible test environments to the corresponding test coverage criteria.

Test suite	Test coverage criteria							
	Functional requirement coverage	Software structural coverage	System integration coverage	System variation coverage	Software performance coverage	Training data coverage	Driving scenario coverage	Uncertainty coverage
MiL tests	●	●	○	○	○	○	○	○
SiL tests	●	●	○	○	○	○	○	○
Component-HiL tests	●	○	○	○	●	○	●	○
Cluster-HiL tests	●	○	●	○	●	○	●	○
System-HiL tests	●	○	●	●	○	○	○	○
Vehicle and proving ground tests	●	○	●	●	○	●	○	○
Driving simulator tests	○	○	●	○	○	○	●	○
N-FOTs	○	○	●	●	●	●	●	●
Regression tests/big data re-simulations	○	○	○	●	●	●	○	●

●: Recommended test coverage.
○: Not useful.

in the system context. For example, when AD functions are developed in a variety of system variants that can include static and dynamic tolerances within the Ego-vehicle. The software performance coverage determines the robustness of the software using fault injection techniques. The training data coverage specifies which training data is required for a particular application and which data leads to the most accurate results, such as training of neural networks for image processing. The driving scenarios coverage identifies the known critical scenarios, which should exhibit similar behaviour, and minimises unknown critical scenarios. Uncertainty coverage quantifies the aleatoric uncertainty and epistemic uncertainty of machine learning algorithms.

6. Evaluation

6.1. Situation-based open-loop verification

While the environmental perception sensors react sensitively to the target hardware constraints, the MV sensor ECU without camera optics is integrated for regression tests with recorded sequences. Fig. 9 depicts the general data flow of a regression test using the example of an eliminated software error within the object detection of a MV sensor ECU. The relevant situations with the algorithm (A_i) are recorded by the Ego-vehicle with appropriate measurement equipment and collected within a data ingestion process. Following an algorithm update within a software development process, the open-loop HiL test bench stimulates the external interfaces of the MV sensor ECU with recorded data to verify the algorithm (A_{i+1}) functionality. The open-loop HiL stimulates the external interfaces in real-time with a recorded sequence (B_i) to generate a new sequence (B_{i+1}). Thus, original and new sequences can be compared to decide whether the error is fixed according to defined pass-/fail criteria. The situation-based open-loop testing generates driving situations from the test-case description and evaluates the behavioural response without feeding it back into future situations.

6.2. Scenario-based Closed-Loop Verification

The scenario-based test method verifies the behaviour in a closed-loop to prove functional correctness of an artifact against its functional specification. Therefore, closed-loop testing specifies an entire scenario in a test case that contains a sequence of scenes, actions, events and goals for the AD function. The behavioural reactions are used to influence future scenes and thus also future situations. The HiL test method integrates the AD function into the traffic environment and vehicle dynamics simulations by combining the simulation models and the ECU hardware into a real-time ECU test bench. However, the HiL test method integrates the executable software code generated from the same source as for the vehicle ECU. Scenario-based testing requires various technical requirements for the simulation environment of roads, traffic objects, environmental perception sensors, Ego-vehicle driver, commercial vehicle dynamics and actuators. Furthermore, the traffic environment shall be animated in 3D perspective to display the traffic scenes in a virtual world in the form of a video sequence recorded by a MV sensor ECU, as illustrated in Fig. 10. The MV sensor ECU is connected to the Cluster-HiL via

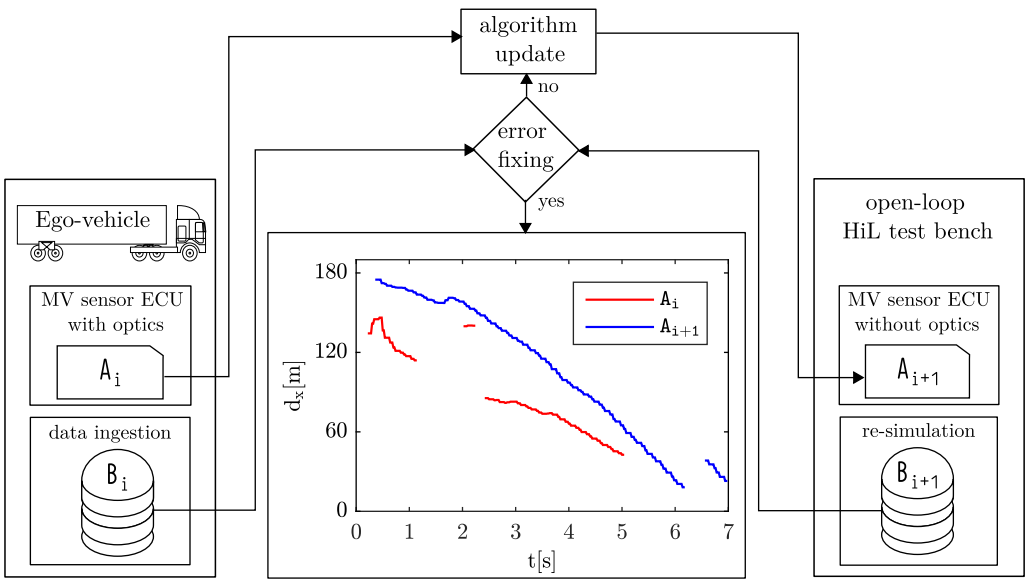

Fig. 9. Regression test process with the open-loop HiL test bench using the example of shift and loss of oncoming object detections with a MV sensor ECU (Elgharbawy et al., 2016).

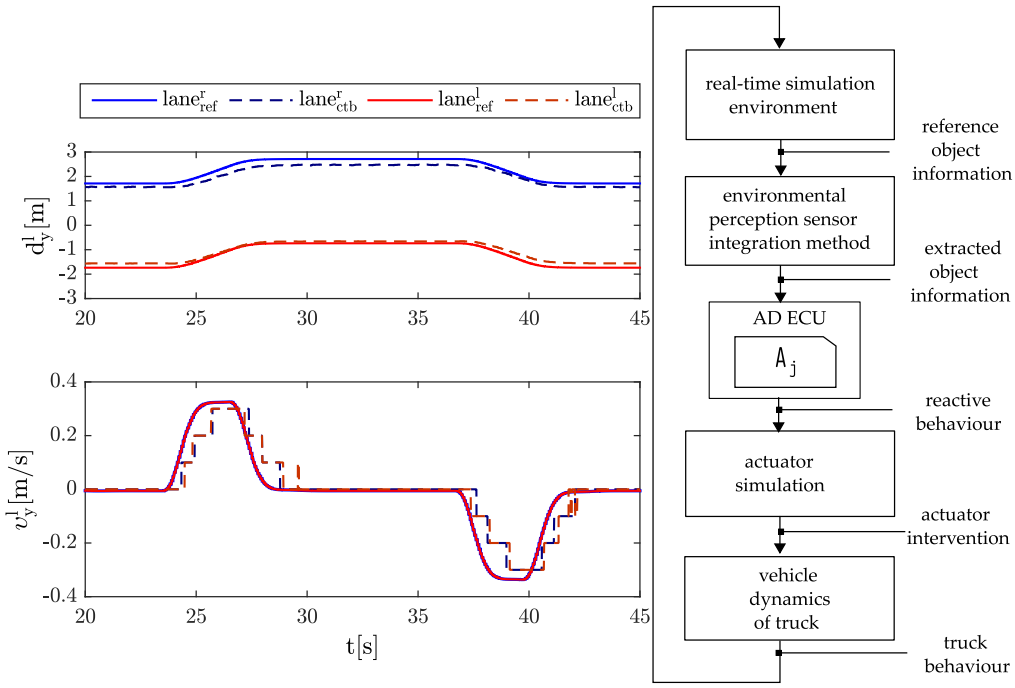

Fig. 10. Verification process with the closed-loop HiL test bench using the example of lane departure in a lane change scenario with a MV sensor ECU (Elgharbawy et al., 2017).

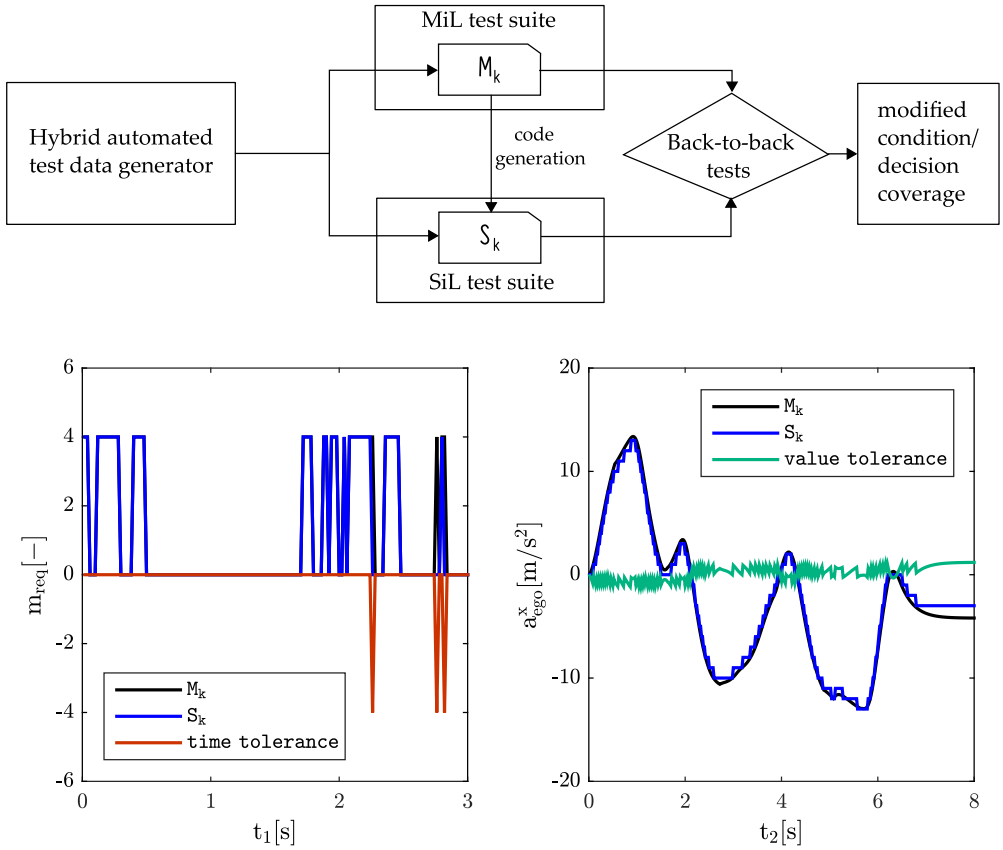

Fig. 11. Time and value tolerances obtained by automatic test data generation of model/software back-to-back tests.

CAN interface and is located in the front of a flat monitor, while the camera processes the images displayed in front. The lane change scenario is applied to algorithm $(\mathtt{A_j})$ and compares the observed parameters $d_y^l[m]$ and $\upsilon_y^l[m/s]$ from the MV sensor ECU through the camera projection with its reference parameters from the real-time simulation environment.

6.3. Automated test data generation techniques

Automated test data generation can be applied, if a test oracle can be defined automatically with its reference information. For example, structural testing is typically employed to generate test data based on the internal structure of the test object. Therefore, the identification of input values depends on a selected path or branch that is executed within the test object. Model-based software development in the automotive industry uses tools such as Simulink or TargetLink to implement a software module within the AD functions. Tools such as Simulink Coder or TargetLink are then used to automatically generate C software source code from the resulting models. Therefore, ISO 26262:2018 demands that coverage metrics shall be taken into account when testing at the model and software code level, such as MC/DC code coverage. A major cause of such semantic differences is the application of scaling to variables during software code generation to optimise code efficiency and value precision. Back-to-back tests generate a collection of structural test cases to compare software generation behaviour with the behaviour of the model underlying the software. Therefore, both model and software are executed with the same input data and then corresponding output data entries are compared. Wilmes introduces a hybrid test data generation approach to combine static analysis and dynamic test data generation (Wilmes, 2016). On the one hand, the test data finding problem is converted into an optimisation problem by defining a cost function. As a result, the generated test data is evaluated to distinguish between relevant and irrelevant test data and to generate new test data in each iterative cycle. On the other hand, the static analysis serves to accelerate the automatic search by identifying unreachable model states. Fig. 11 illustrates an example for defining a test oracle to evaluate the automatically generated test cases according to the accepted time and value tolerances, which can be determined from the signals m_{req} and a_{ego}^x respectively. The model $(\mathbb{M}_k)$ runs with the MiL test suite and is verified back-to-back with the software $(\mathtt{S_k})$ running on the SiL test suite.

7. Executive summary

The status quo evaluation refers to large-scale verification as one of the decisive challenges for the economical, reliable and safe use of automated driving functions in truck product engineering. Therefore, the systems engineering has established data- and knowledge-driven test methods to assure the required dependability of their products. However, the reliance on N-FOTs is inadequate and, in particular, time and cost-intensive when applied to the next generation of automated driving functions, e.g. AEB and truck platooning. Therefore, innovative approaches are required to enable systematic testing under reproducible conditions with regard to robustness, reliability and safety. In addition, functional decomposition is also necessary to support the argumentation for a reasonably low residual risk resulting from imperfections of the environmental perception sensors. The presented framework structure utilises a back-end database filled with catalogues of relevant driving scenarios from different sources of field-based observations. In this scheme, the processing chain includes clustering of multivariate time series datasets and finding critical driving situations to identify and allocate the necessary test cases for various suitable test environments. Afterwards, these new test cases complement the existing test cases developed from expert knowledge in an adaptive test coverage manner. The platform-independent mechanism is intended to offer a consistent scenario description format for the various test environments. The proposed framework provides a potential trade-off between efficiency and effectiveness criteria of a scenario-based test concept. Fig. 12 shows an integration of adaptive verification within an agile development process that uses standard quality gates for automated driving algorithms in complex automotive sensor networks. The proposed approach employs an ontology for generation of test scenario catalogues using event-based time series analysis for the Cluster-HiL co-simulation framework. The proposed procedure offers an optimised test strategy for the systematic extension of the requirements-based test coverage based on a modular verification framework with continuous knowledge enhancement from field observations. In this scheme, the processing chain includes hierarchical clustering of time series triggering events to identify and assign the necessary test cases for different appropriate test environments. Using an ontology-based method, a category of adequate and relevant scenarios for existing field tests is extracted. A semantic representation of worst-case scenarios can be obtained by using data mining techniques and systematically processed in requirements for adaptive test coverage. The industry-proven framework facilitates a functional verification of automated driving functions precisely and more efficiently on the target ECU in the laboratory. The real-time framework can benchmark the performance of the automated driving algorithms at the electronic system level using the proposed HiL co-simulation platform. The presented research provides a quantitative approach for a trade-off between physical realism and computational efforts of the real-time synthetic simulation. The proposed framework illustrates a generic architecture of fault injection of environmental perception sensors for robustness testing of the DuT. Recently the heterogeneous simulation environments become more convenient to cover the multidisciplinary nature of the CPVS system. The proposed concept is generic and can be extended to any object-list-based sensor. The platform-independent mechanism is intended to offer a consistent interface between the simulation components via a similar FMI interface. The industry-proven framework facilitates a functional verification of automated driving functions precisely and more efficiently on

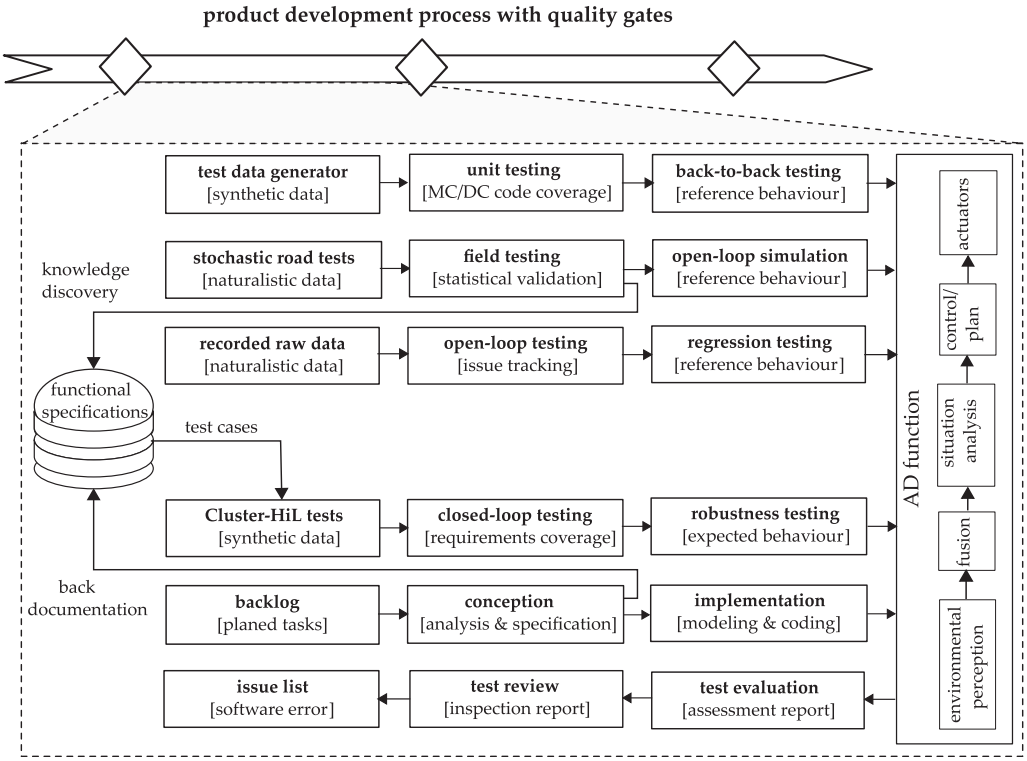

Fig. 12. Adaptive verification integration in an agile development process for automated driving functions.

the target ECU in the laboratory. Therefore, the proposed methods can reduce the verification efforts in the public space to a minimum and support appropriate functional testing in virtual environments.

8. Outlook and future work

The status quo of automated driving functions extends to partial automation (level 2) according to the SAE automation levels. In these functions, e.g. AEB and highway pilot, the driver still retains control over the vehicle and remains obliged to monitor the functional intervention regularly and, if necessary, to resume vehicle control. Till now, the controllability of system interventions and their effectiveness in the field with minimisation of undesired consequences play the decisive roles for the series product development of these driving functions. Accordingly, the series production based on systems engineering requires state-of-the-art evaluation procedures to verify and validate these features. The quality of automated driving depends primarily on the environmental sensors providing the vehicle's environmental perception as the basis for decision making and situation analysis. An acceptable level of maturity of these sensors must be accomplished as a prerequisite for an adequate field validation. Systematic, random faults or functional deficiencies are assessed by objective and subjective evaluation criteria. User-oriented assessment procedures are the current de-facto standard for the validation of automated driving functions. These procedures provide system performance matrices, e.g. confusion matrices for all possible system reactions with classification as intended functional interventions or unintended side effects. In this scheme, the evaluation of software releases must be carried out in various phases up to the SOP. Initially by XiL technologies, followed by driving simulators, test drives with trained test supervisors on both test tracks and public roads, test drives by intended users and eventually ending with the vehicle type approval homologation. Since it is not possible to guarantee absolute safety for automated trucks, one of the major challenges in automated truck driving is to argue for a reasonably low residual risk resulting from imperfections of the environmental perception sensors. Such arguments are currently not supported by the relevant safety norms. An optimised test strategy demands a selection of the necessary test methods (simulation/laboratory, proving ground or field testing) for different scenarios and their interaction with other test methods. Consequently, new innovative approaches need to be established, especially in simulation and in the laboratory. Evidence should be provided by scenario coverage of the tests combined with statistical extrapolation techniques, field-based observations, component and integration tests including simulation as well as reasonable safety measures. The presented work raises several issues that require substantial future research activities. Further research will also include the application of clustering of multivariate time-series data. These activities have to be integrated into a system engineering approach that supports the structure of the adaptive verification. This research work needs to be complemented by activities with standard organisations to form a consensus

on risk evaluation and acceptable argumentation structures that would feed into future standards and CoP guidelines for the verification of automated driving functions.

References

Airbus, 2016. Statistical evaluation of accidents in commercial aviation.

Asirt, R.C.S., 2018. Association for safe international road travel.

Burton, S., Gauerhof, L., Heinzemann, C., 2017. Making the case for safety of machine learning in highly automated driving. International Conference on Computer Safety, Reliability, and Security. Springer, pp. 5–16.

Damm, W., Heidl, P., 2017. Safetrans working group "highly automated systems:test, safety, and development processes", Recommendations on Actions and Research Challenges

Destatis, 2015. Federal statistical office.

Elgharbawy, M., Bernier, B., Frey, M., Gauterin, F., 2016. An agile verification framework for traffic sign classification algorithms in heavy vehicles. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1–8. https://doi.org/10.1109/AICCSA.2016.7945719.

Elgharbawy, M., Schwarzhaupt, A., Frey, M., Gauterin, F., 2017. A real-time multisensor fusion verification framework for advanced driver assistance systems. Transp. Res. Part F: Traffic Psychol. Behav. https://doi.org/10.1016/j.trf.2016.12.002.

Kirschbaum, M., 2015. Highly automated driving for commercial vehicles. In: Pfeffer, P. (Ed.), 6th International Munich Chassis Symposium 2015. Springer Fachmedien Wiesbaden, Wiesbaden, pp. 5–15.

Schöner, H.-P., 2016. Challenges and approaches for testing of highly automated vehicles. In: Langheim, J. (Ed.), Energy Consumption and Autonomous Driving, Lecture Notes in Mobility. Springer, pp. 101–109.

Smith, B.W., 2017. Automated driving and product liability. Mich. St. L. Rev. 1.

Steinmeyer, S., 2014. Probabilistische Fahrzeugumfeldschätzung für Fahrerassistenzsysteme (Ph.D. thesis). Technische Universtität Braunschweig, Braunschweig.

Trigell, A.S., Rothhämel, M., Pauwelussen, J., Kural, K., 2017. Advanced vehicle dynamics of heavy trucks with the perspective of road safety. Vehicle Syst. Dyn. 55, 1572–1617.

Wilmes, B., 2016. Tasmo: Automated test data generation for simulink model coverage. In: Gühmann, C., Riese, J., Rüden, K.v. (Eds.), Simulation and testing for vehicle technology. Springer, Cham, pp. 123–133.

Winner, H., 2015. ADAS, Quo Vadis? Springer International Publishing, Cham, pp. 1557–1584.

Winner, H., Wachenfeld, W., Junietz, P., 2018. Validation and introduction of automated driving. Automotive Systems Engineering II. Springer, pp. 177–196.

Further reading

Bengler, K., Dietmayer, K., Farber, B., Maurer, M., Stiller, C., Winner, H., 2014. Three decades of driver assistance systems: review and future perspectives. IEEE Intell. Transp. Syst. Mag. 6, 6–22.

Committee, S.O.-R.A.V.S. et al, 2014. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. SAE Int.

Duraisamy, B., Schwarz, T., 2015. On track-to-track data association for automotive sensor fusion. International Conference on Information Fusion.

Dokic, J., Müller, B., Meyer, G., 2015. European roadmap smart systems for automated driving. Eur. Technol. Platform Smart Syst. Integration.

Elgharbawy, M., Schwarzhaupt, A., Scheike, G., Frey, M., Gauterin, F., 2016. A generic architecture of adas sensor fault injection for virtual tests. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1–7. https://doi.org/10.1109/AICCSA.2016.7945680.

Elgharbawy, M., Schwarzhaupt, R., Arenskrieger, A., Elsayed, H., Frey, M., Gauterin, F., 2018. A testing framework for predictive driving features with an electronic horizon. Transp. Res. Part F: Traffic Psychol. Behav. https://doi.org/10.1016/j.trf.2017.08.002.

Feilhauer, M., Haering, J., Wyatt, S., 2016. Current approaches in hil-based adas testing. SAE Int. J. Commer. Veh. 9, 63–69.

Horita, Y., Schwartz, R.S., 2015. Extended electronic horizon for automated driving. In: 14th International Conference on ITS Telecommunications IEEE, pp. 32–36.

Jo, K., Sunwoo, M., 2014. Generation of a precise roadway map for autonomous cars. IEEE Trans. Intell. Transp. Syst. 15, 925–937.

Kessels, J., van den Bosch, P., 2007. Electronic horizon: Energy management using telematics information. IEEE Vehicle Power and Propulsion Conference. IEEE, pp. 581–586.

Kienle, M., Franz, B., Winner, H., Bengler, K., Baltzer, M., Flemisch, F., Kauer, M., Weißgerber, T., Geyer, S., Bruder, R., Hakuli, S., Meier, S., 2014. Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. IET Intel. Transport Syst. 8, 183–189.

Kritzinger, D., 2017. 2 - safety assessment strategy (with goal structuring notation). In: Kritzinger, D. (Ed.), Aircraft System Safety. Woodhead Publishing, pp. 23–35.

Miegler, M., Schieber, R., Kern, A., Ganslmeier, T., Nentwig, M., 2009. Hardware-in-the-loop test of advanced driver assistance systems. ATZelektronik worldwide 4, 4–9.

Pütz, A., Zlocki, A., Bock, J., Eckstein, L., 2017. System validation of highly automated vehicles with a database of relevant traffic scenarios. 12th ITS European Congress, vol 1.

Pütz, A., Zlocki, A., Küfen, J., Bock, J., Eckstein, L., 2017. Database approach for the sign-off process of highly automated vehicles. 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration.

Sasse, V., 2017. Autonomous driving from individualism towards collectivism. ATZelektronik Worldwide 12, pp. 72–72.

Sinclair, I.M., 1984. The Vienna Convention on the Law of Treaties. Manchester University Press.

Stellet, J.E., Zofka, M.R., Schumacher, J., Schamm, T., Niewels, F., Zollner, J.M., 2015. Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions. In: IEEE 18th International Conference on Intelligent Transportation Systems. IEEE, Piscataway, NJ, pp. 1455–1462.

Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., Maurer, M., 2015. Defining and substantiating the terms scene, situation, and scenario for automated driving. 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 982–988.

Varshney, K.R., Alemzadeh, H., 2017. On the safety of machine learning: cyber-physical systems, decision sciences, and data products. Big Data 5, 246–255.

Wagner, P., 2016. Challenges in autonomous vehicle testing and validation. 2016 SAE World Congress.

Zinoune, C., Bonnifait, P., Ibanez-Guzman, J., 2012. Detection of missing roundabouts in maps for driving assistance systems. IEEE Intelligent Vehicles Symposium. IEEE, Piscataway, NJ, pp. 123–128.