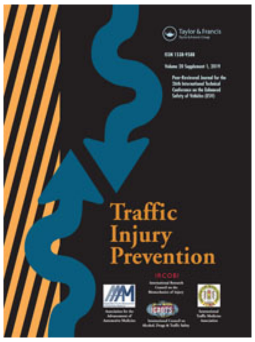

Traffic Injury Prevention

ISSN: 1538-9588 (Print) 1538-957X (Online) Journal homepage: https://www.tandfonline.com/loi/gcpi20

Functional decomposition—A contribution to overcome the parameter space explosion during validation of highly automated driving

Christian Amersbach & Hermann Winner

To cite this article: Christian Amersbach & Hermann Winner (2019) Functional decomposition—A contribution to overcome the parameter space explosion during validation of highly automated driving, Traffic Injury Prevention, 20:sup1, S52-S57, DOI: 10.1080/15389588.2019.1624732

To link to this article: https://doi.org/10.1080/15389588.2019.1624732

TRAFFIC INJURY PREVENTION
2019, VOL. 20, NO. S1, S52–S57
https://doi.org/10.1080/15389588.2019.1624732

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Functional decomposition—A contribution to overcome the parameter space explosion during validation of highly automated driving

Christian Amersbach and Hermann Winner

Institute of Automotive Engineering, Technische Universität Darmstadt, Darmstadt, Germany

ABSTRACT

Objective: Particular testing by functional decomposition of the automated driving function can potentially contribute to reducing the effort of validating highly automated driving functions. In this study, the required size of test suites for scenario-based testing and the potential to reduce it by functional decomposition are quantified for the first time.

Methods: The required size of test suites for scenario-based approval of a so-called Autobahn-Chauffeur (SAE Level 3) is analyzed for an exemplary set of scenarios. Based on studies of data from failure analyses in other domains, the possible range for the required test coverage is narrowed down and suitable discretization steps, as well as ranges for the influence parameters, are assumed. Based on those assumptions, the size of the test suites for testing the complete system is quantified. The effects that lead to a reduction in the parameter space for particular testing of the decomposed driving function are analyzed and the potential to reduce the validation effort is estimated by comparing the resulting test suite sizes for both methods.

Results: The combination of all effects leads to a reduction in the test suites' size by a factor between 20 and 130, depending on the required test coverage. This means that the size of the required test suite can be reduced by 95–99% by particular testing compared to scenario-based testing of the complete system.

Conclusions: The reduction potential is a valuable contribution to overcome the parameter space explosion during the validation of highly automated driving. However, this study is based on assumptions and only a small set of exemplary scenarios. Thus, the findings have to be validated in further studies.

ARTICLE HISTORY
Received 29 October 2018
Accepted 23 May 2019

KEYWORDS
Automated driving; functional decomposition; test coverage; PEGASUS; parameter space; validation

Introduction

The technical development of autonomous vehicles has reached a level that would soon allow a market introduction. Highly automated driving (HAD; i.e., SAE Level 3 and higher (Society of Automotive Engineers 2014) has been recently announced by different car makers but is not available on the market yet. This is despite the fact that car makers, suppliers, and research facilities have successfully demonstrated prototypes of automated vehicles in public. The high safety requirements are one reason why the technology is still not available on the market. These requirements are not yet quantified and they make validating highly automated vehicles a challenge.

Billions of kilometers until the release of vehicles with unsupervised automated driving

If the current test, concepts such as distance-based approval were maintained for HAD, the required test distances in real traffic would increase dramatically. Assuming that the automated driving function is twice as safe—based on the number of fatal accidents—as a human driver, around 6.6 billion test kilometers would have to be driven under representative conditions for the validation of a so-called Autobahn Chauffeur in Germany (Wachenfeld and Winner 2016). The validation of a Level 5 system in the United States would require distances up to 11 billion miles (Kalra and Paddock 2016). The mentioned examples show that a statistical validation of HAD is not feasible in practice before introduction, especially because the testing would need to be repeated upon modification of the system. Thus, alternative validation methods are required. For example, Schuldt (2017) proposed methods for systematic test case generation in virtual environments, and Althoff and Dolan (2014) proposed an approach for formal online verification. System theoretical process analysis has been used successfully for the safety analysis of automated vehicles (Bagschik et al. 2017), and

CONTACT Christian Amersbach amersbach@fzd.tu-darmstadt.de Institute of Automotive Engineering, Technische Universität Darmstadt, Otto-Berndt-Str. 2, Darmstadt 64283, Germany.

Associate Editor H. Clay Gabler oversaw the review of this article.

Supplemental data for this article can be accessed on the publisher's website.

the voluntary safety self-assessment introduced by NHTSA has led to safety reports from 10 companies so far (NHTSA 2019). Additionally, a standard that covers the behavioral safety of road vehicles, called SOTIF (safety of the intended functionality), has recently been developed by the International Organization for Standardization (2019).

Scenario-based approach

The scenario-based approach is an alternative to the above-mentioned statistical approach. It is assumed that scenarios that are challenging for the object under test (OUT; i.e., the automated vehicle) are quite rare and happen randomly in real traffic and the major part of mileage can be driven without any issues or critical situations. Because testing all of the aforementioned ordinary scenarios would not trigger any unknown faults at some point, it would not contribute to the validation process. Therefore, the long test driving distances needed for statistical validation should be significantly reduced by the identification of challenging or critical scenarios that can be reproduced in simulation or on test fields. Critical scenarios are identified by metrics (Junietz et al. 2017) or created automatically (e.g., Bagschik et al. 2018) and can be stored in a central scenario database (Pütz et al. 2017). Scenarios can be described on different levels of detail (Menzel et al. 2018):

- Functional scenarios (semantic description in linguistic notation)
- Logical scenarios (description in formal notation including parameter range)
- Concrete scenarios (description in formal notation including concrete parameter values).

Whereas functional scenarios are used to specify the operational design domain in the concept phase, logical scenarios are used during the system development and concrete scenarios are used as test cases.

Methods

To quantify the potential to reduce the number of required test cases by particular testing of a functionally decomposed driving function, in a first step the so-called parameter space explosion is analyzed. Secondly, the 3 main effects that lead to a reduction are investigated. Finally, the number of required test cases for an exemplary set of scenarios for both particular testing and testing of the complete system as a reference are estimated and compared to each other.

Parameter space explosion

The scenario-based approach can potentially reduce the approval effort for HAD. However, it still leads to a huge number of concrete scenarios that have to be evaluated in test cases, even for a single logical scenario. For example, in its safety report, Waymo (2017) states that they "create thousands of variations" of one single scenario. This so-called parameter space explosion is mainly caused by 3 factors, which are addressed in the following sections:

Influence parameters

Even simple logical scenarios are described by a multitude of parameters that affect the OUT. Those influence parameters can be structured in a 5-layered model (Bagschik et al. 2018). Schuldt (2017) described how those influence parameters can be selected and analyzed and noted different information sources, such as regulations for road constructions, vehicle catalogs, or expert knowledge. For each parameter p_i, v_i different values can be assigned.

Because the majority of the parameters have a continuous parameter space (e.g., speed), v_i depends on the chosen discretization step width. Choosing the correct discretization step width a priori (e.g., without simulating parameter sets with different discretizations and comparing the results) is a challenge yet to be solved, because too coarse discretization on the one hand will lead to gaps in the parameter space that might lead to undiscovered issues during testing. Too fine discretization, on the other hand, will lead to a higher number of test cases than necessary. The authors do not attempt to solve the discretization challenge here but assume values for v_i in this study based on expert knowledge. Influence parameters for exemplary scenarios can be found in Table 1 in the Appendix (see online supplement).

Systematic test case generation

A scenario that is defined by N parameters $(p_1, \dots, p_i, \dots, p_N)$ with v_i instances will lead to

$$S_N = \prod_{i=1}^{N} v_i \tag{1}$$

possible test cases (Grindal et al. 2005). Thus, with each additional parameter or discretization step, S_N progressively increases. For the exemplary parameter space analyzed in this study, S_N is around 10^{31}. Due to time and cost limits, it is not possible to test all theoretically possible test cases (Sommerville 2006). Therefore, systematic test case generation is required. Grindal et al. (2005) provided an overview of combined strategies that can be used for systematic test case generation. The most important constraint for choosing a combined strategy is the required test coverage, which will be handled in the following subsection.

Required test coverage

In this article, the so-called (100%) t-wise coverage is used as a metric to describe the coverage of a certain test suite; that is, a set of test cases. Grindal et al. (2005) defined (100%) t-wise coverage as follows: For (100%) t-wise coverage, "[…] every possible combination of all […] values of t parameters [has to] be included in at least one test case in the test suite" (p. 171).

In the remainder of this article the term t-wise always refers to 100% t-wise coverage. Having defined a metric for test coverage, the test coverage required to assess the safety of HAD functions still has to be determined. Gründl (2005) stated that traffic accidents are typically caused by a combination of several factors, following the so-called Swiss cheese model introduced by Reason (1990). Assuming that failures in HAD functions are multicausal as well, a 1-wise coverage is not sufficient. Kuhn et al. (2004) analyzed empirical data from error reports in various domains and introduced the failure-triggering fault interaction (FTFI) number. The FTFI number is "[…] the number of conditions required to trigger a failure" (Kuhn et al. 2004, p. 418). This means that any failure with an FTFI number smaller than or equal to t will be discovered by testing with t-wise coverage. In the data analyzed by Kuhn et al. (2004), the FTFI number does not exceed 6.

However, empirical data do not yet exist for HAD; thus, the FTFI numbers for such systems are unknown. Assuming that the findings from other domains can to some extent be transferred to HAD and lie in the same magnitude, this study assumes that 10-wise coverage in the worst case and 3-wise coverage in the best case are required to discover an adequately high fraction of all possible failures.

Equation (1) can only be used to calculate the size of a test suite with N-wise coverage. Kuhn et al. (2004) deduced that the size of a test suite with t-wise coverage can be estimated with v^t for the simple case that every parameter p_i has the same number of possible values $v_i = v$ and the test case generation is "perfectly efficient," meaning that there are no duplicates in the test suite.

Resulting parameter space explosion

Having discussed the contributing factors for the parameter space explosion in the previous section, the size of the resulting test suites can be calculated, based on the assumptions made in the current study. However, for a real application, the assumption that each parameter has the same number of possible values is not valid because, for example, the width of a traffic lane has fewer discretization steps than the position of the sun. Therefore, the derivation above is combined with Eq. (1) and the size S_t of a test suite with N parameters and t-wise coverage can be calculated as follows:

$$S_t = \prod_{i=1}^{t} \max_i(v_1, \ \ldots, v_N) \qquad (2)$$

Here, $\max_i(V)$ is defined as the i-greatest element of the set V.

Thus, with a given value t for the required coverage, only the t parameters with the greatest number of discretization steps have an influence on the size of the test suite. Therefore, S_t can be significantly reduced by eliminating the parameters with the finest discretization or by reducing the number of discretization steps.

Using Eq. (2) for some exemplary scenarios, values around $7.5 \cdot 10^5$ for S_3 and around 10^{13} for S_{10} are obtained, as illustrated in Figure 1. S_t grows exponentially with t. Therefore, the required coverage should be kept as low as possible.

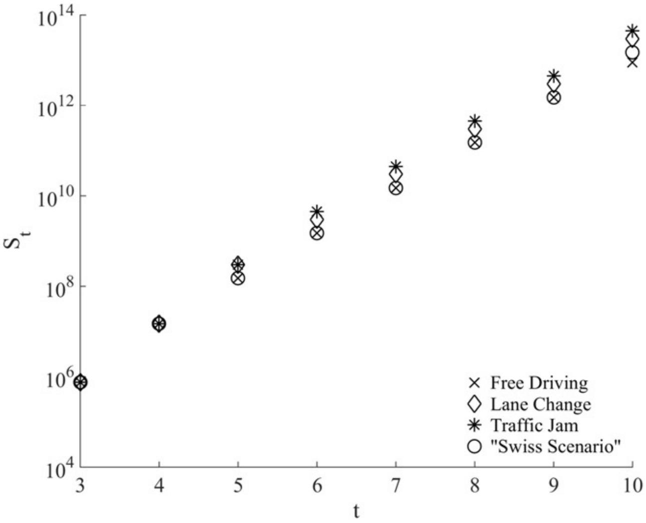

Figure 1. Size of test suits S_t for t-wise coverage of different scenarios.

Functional decomposition to overcome the parameter space explosion

One approach to overcome the previously outlined parameter space explosion is functional decomposition of the driving function to derive particular test cases, as introduced by the authors (Amersbach and Winner 2017).

Approach

The approach of functional decomposition is broadly used in different domains (e.g., mathematics or informatics) to decompose complex problems, functions, or systems into less complex subproblems, -functions, or -systems. The authors suggest transferring this method to HAD functions and using it to generate particular test cases for the individual functional layers (i.e., subsystems) rather than testing the complete function in a system test. Therefore, the driving function is decomposed into functional layers that are tested individually and are evaluated on their interfaces. For this particular testing, evaluation criteria for each functional layer have to be derived from evaluation criteria or safety requirements on a system level. This can be achieved by applying fault tree analysis or similar methods. For example, the system-level fail criterion "(physically avoidable) collision with an obstacle" can be decomposed to minimum required object detection (functional layer 1), classification (layer 2), and prediction (layer 3) criteria as well as criteria for the planning (layer 4) and execution (layer 5) of a successful evasion or braking maneuver. The functional layers and the interfaces between them are based on the decomposition of the human driving task by Graab et al. (2008) and can be seen in Figure 2. The concept of particular testing is introduced in detail in (Amersbach and Winner 2017).

Shrinking the parameter space with particular testing

Among other benefits (cf. Amersbach and Winner 2017), particular testing can significantly shrink the parameter space, mainly based on three effects:

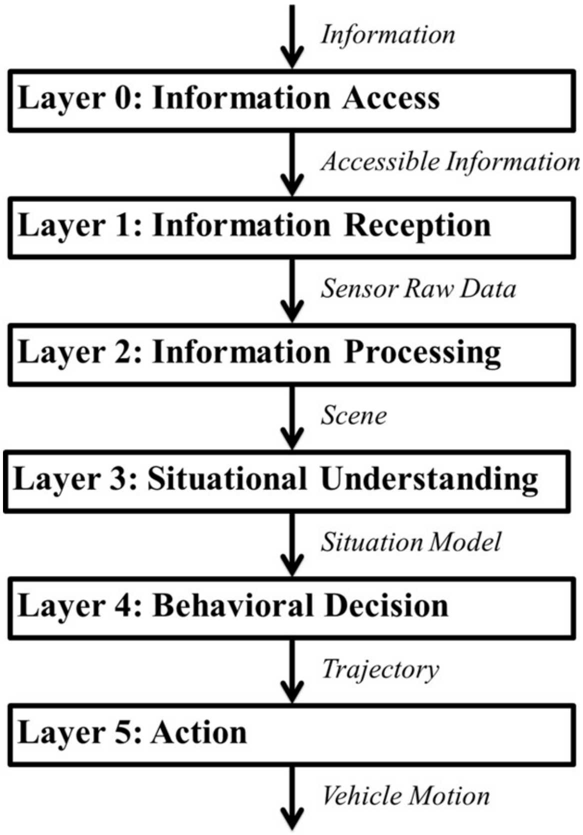
Information
Layer 0: Information Access
Accessible Information
Layer 1: Information Reception
Sensor Raw Data
Layer 2: Information Processing
Scene
Layer 3: Situational Understanding
Situation Model
Layer 4: Behavioral Decision
Trajectory
Layer 5: Action
Vehicle Motion

Figure 2. Decomposition layers.

- The parameter space for one single layer is smaller than the parameter space for the complete HAD function: Most of the parameters only have an influence on some of the layers; for example, the majority of the parameters representing the environment, like the sun's position or weather conditions, only affect functional layers 1 and 2 (environment perception). Therefore, the influence parameters for each functional layer are subsets of the total parameter space. However, this only has an effect if finely discretized parameters are not relevant for all layers. This effect reduces the size of the test suite by around 50% for higher test coverages (i.e., $t \geq 6$) for the exemplary scenarios analyzed in this study. However, for lower test coverages (i.e., $t \leq 3$), the size of the test suite will be increased. Nevertheless, because a particular test most likely requires less effort compared to a system test, the total test effort can still be reduced.
- Less complex subsystems require a smaller test coverage: Because the maximum FTFI number of all failures within a system depends on the complexity of the system, it is assumed that the maximum FTFI number of a subsystem is lower than the maximum FTFI number of the complete system and therefore the required coverage for particular testing is smaller compared to testing of the complete system. However, the required test coverage for a new system cannot be predicted and empirical analysis of the FTFI numbers for HAD functions and their subfunctions is not possible yet, because not enough data are available. This effect would lead to a reduction of the parameter space by one order of magnitude if the required coverage for particular testing were reduced by at least one (i.e., $t_{\text{part.}} \leq t_{\text{system}} - 1$) compared to testing of the complete system (compare Figure 1).
- The test of the perception layers can be aggregated for a set of similar scenarios: When analyzing the influence parameters, it becomes evident that the majority of parameters with a high number of possible values only have an influence on the perception of the HAD function that is represented in functional layers 0–2.

An obvious example is the position of the sun, which only influences layers 1 and 2. Additionally, its possible parameter space is rather large, considering that the azimuth angle (relative to the OUT) can obtain values between 0° and 360°, whereas the elevation angle is spread between around $-10°$ and 90°, depending on the topology and location. Here one could argue that only sun positions within the field of view of cameras and LIDAR have to be considered, which might be true when only evaluating direct blinding effects. However, to include perception errors caused by sun reflections from static or dynamic objects in the scene, all possible sun positions in combination with object attributes and positions have to be included in the test suite and an adequate discretization of the angular position has to be chosen.

Aggregating the parameters with an influence on the perception layers in one equivalency class scenario for a set of similar scenarios (e.g., scenarios on a 2-lane Autobahn) that contains the parameter space for the complete scenario set could thus further reduce the size of the test suite. Whereas the layers 0–2 are tested using the equivalency class scenario, which contains all possible combinations of all parameters with and influence on those layers, layers 3–5 are tested in the original scenario set. In the exemplary scenario set analyzed in this study, the size of the test suite was reduced by over 50% for small test coverages by this approach. However, for 9-wise or higher coverage, the size of the test suite was increased (see Figure 3). This can be explained by the fact that the equivalency class scenario is more complex than the scenarios from the set and therefore contains more influence parameters with a high number of possible values.

Application on an exemplary set of scenarios

For this study, the functional decomposition method is applied to an exemplary set of 9 scenarios to evaluate its potential to overcome the parameter space explosion. The "Swiss scenario" (see Figure 4) is a representation of a scenario that led to a real-world accident of a Tesla Model S in Switzerland in May 2016. The other 8 logical scenarios (free driving, following, lane change, cut in, cut out, cut through, passing an obstacle, and approaching the end of a traffic jam) are used in the project PEGASUS. The choice of the parameters and their allocated discretization steps is slightly arbitrary because it is purely based on assumptions made by the authors due to the lack of empirical data. However, choosing different parameters and discretization steps would

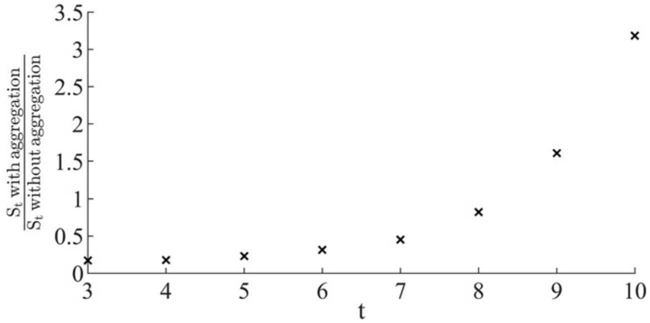

Figure 3. Reduction of the test suite size by aggregation of parameters in a equvivalency class scenario.

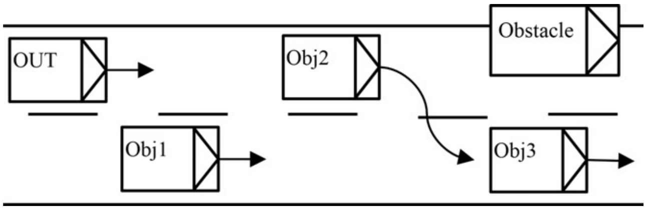

Figure 4. "Swiss Scenario".

change the absolute values but would not significantly affect the relative comparison between particular testing and testing of the complete system. The allocated parameters can be found in Table 1 in the Appendix.

Results

The combination of the 3 effects discussed above leads to a reduction of the test suite by a factor of 20 for 10-wise coverage and by a factor of 130 for 3-wise coverage, as illustrated in Figure 5. This means that the size of the required test suite is reduced by 95–99% by particular testing compared to scenario-based testing of the complete system.

In the analyzed exemplary parameter space, the absolute size of the test suites could be reduced from $6.8{\cdot}10^6$ to $5{\cdot}10^4$ concrete scenarios for 3-wise coverage and from $6.7{\cdot}10^{14}$ to $3.1{\cdot}10^{13}$ concrete scenarios for 10-wise coverage.

Discussion

In this article, the required size of test suites for scenario-based approval of HAD is analyzed for an exemplary set of scenarios. The discretization steps of the influence parameters and the required test coverage mainly influence the size of the test suites. On the one hand, this leads to a discretization challenge that cannot be solved within this study; on the other hand, the required test coverage could only be defined by analyzing empirical data, which are not yet available for HAD. Nevertheless, based on studies of data from other domains, the possible range for the required test coverage is narrowed down and discretization steps for the influence parameters are assumed. Based on those assumptions, the potential to shrink the parameter space by functional decomposition and particular testing of HAD functions is analyzed. Though the absolute values for the test suite sizes strongly depend on the assumptions made throughout their creation, the relative findings are not affected because the same assumed data are used to compare both methods, particular testing and testing of the complete system. However, the findings have yet to be validated with a practical implementation of the functional decomposition approach.

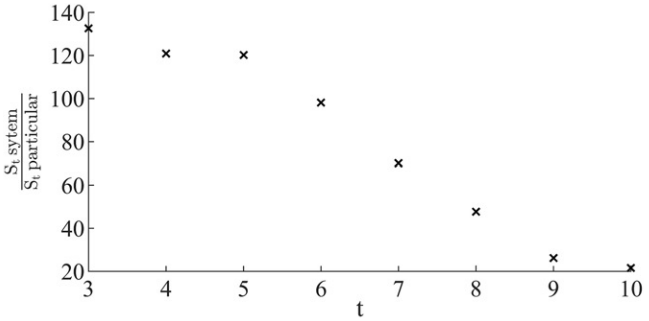

Figure 5. Potential reduction of the parameter space by functional decomposition.

It is concluded that the functional decomposition approach potentially reduces the size of the required test suite by a factor of 20, …, 130, depending on the required test coverage. This does not solve the challenge of parameter space explosion on its own, because the resulting parameter space is still enormous. Nevertheless, combined with other reduction methods, it is a valuable contribution to overcome the parameter space explosion. If the allocation of influence factors to functional layers is done correctly, the reduction of the test suite sizes does not reduce the fault detection potential, because the reduced parameter space dimensions do not influence the respective functional layers. However, it still has to be proven that testing the functional layers independently can fully replace the testing of the complete system.

Data availability

All data analyzed during this study are included in the Appendix (see online supplement) to this published article.

Funding

This study is part of the project PEGASUS and was funded by the Federal Ministry for Economic Affairs and Energy (BMWi, Grant No. 19A15012F) based on a decision by the German Bundestag.

ORCID

Christian Amersbach http://orcid.org/0000-0002-5153-7401
Hermann Winner http://orcid.org/0000-0002-9824-3195

References

Althoff M, Dolan JM. 2014. Online verification of automated road vehicles using reachability analysis. IEEE Trans Robot. 30(4): 903–918.

Amersbach C, Winner H. 2017. Functional decomposition: An approach to reduce the approval effort for highly automated driving. Paper presented at: 8. Tagung Fahrerassistenz.

Bagschik G, Menzel T, Maurer M. 2018. Ontology based scene creation for the development of automated vehicles. Paper presented at: 2018 IEEE Intelligent Vehicles Symposium (IV).

Bagschik G, Stolte T, Maurer M. 2017. Safety analysis based on systems theory applied to an unmanned protective vehicle. Procedia Eng. 179:61–71.

Graab B, Donner E, Chiellino U, Hoppe M. 2008. Analyse von Verkehrsunfällen hinsichtlich unterschiedlicher Fahrerpopulationen und daraus ableitbarer Ergebnisse für die Entwicklung adaptiver Fahrerassistenzsysteme. In: TU München & TÜV Süd Akademie GmbH, eds. Conference: active safety through driver assistance. München. https://mediatum.ub.tum.de/doc/1145118/.

Grindal M, Offutt J, Andler SF. 2005. Combination testing strategies: a survey. Softw Test Verif Reliab. 15(3):167–199.

Gründl M. 2005. Fehler und Fehlverhalten als Ursache von Verkehrsunfällen und Konsequenzen für das Unfallvermeidungspotenzial und die Gestaltung von Fahrerassistenzsystemen. [PhD Dissertation]. Universität Regensburg.

International Organization for Standardization. 2019. ISO/PAS 21448: 2019 Road vehicles–Safety of the intended functionality: International Organization for Standardization. https://www.iso.org/standard/70939.html.

Junietz P, Schneider J, Winner H. 2017. Metrik zur Bewertung der Kritikalität von Verkehrssituationen und - szenarien. Workshop Fahrerassistenz und automatisiertes Fahren.

Kalra N, Paddock SM. 2016. Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability? RAND Corporation. Research Report. doi:10.7249/RR1478.

Kuhn DR, Wallace DR, Gallo AM. 2004. Software fault interactions and implications for software testing. IEEE Trans Softw Eng. 30(6): 418–421.

Menzel T, Bagschik G, Maurer M. 2018. Scenarios for development, test and validation of automated vehicles. Paper presented at: 2018 IEEE Intelligent Vehicles Symposium (IV), p. 1821–1827.

NHTSA. 2019. Voluntary safety self-assessment. https://www.nhtsa.gov/automated-driving-systems/voluntary-safety-self-assessment.

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

Reason J. 1990. The contribution of latent human failures to the breakdown of complex systems. Philos Trans R Soc Lond B Biol Sci. 327(1241):475–484.

Schuldt F. 2017. Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen [Towards testing of automated driving functions in virtual driving environments]. [PhD Dissertation]. TU Braunschweig.

Society of Automotive Engineers. 2014. J3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. SAE Standard J3016_201401, SAE International.

Sommerville I. Software engineering. New York: Addison-Wesley; 2006.

Wachenfeld W, Winner H. 2016. The release of autonomous vehicles. In: Winner H, Maurer M, Gerdes JC, Lenz B, editors. Autonomous driving: technical, legal and social aspects. Berlin: Springer. p. 425–449.

Waymo. 2017. On the road to fully self-driving: Waymo safety report. Waymo LLC. https://waymo.com/safety/.