

# Szenario-Optimierung für die Absicherung von automatisierten und autonomen Fahrsystemen

Florian Hauer  
Technische Universität München  
florian.hauer@tum.de

Bernd Holzmüller  
ITK Engineering GmbH  
bernd.holzmueller@itk-engineering.de

**Abstract:** Die Verifikation und Validierung automatisierter und autonomer Fahrsysteme, besonders das Finden geeigneter Testsznarien für die virtuelle Absicherung, stellen eine enorme Herausforderung dar.

Diese Arbeit stellt eine Testmethodik vor, die metaheuristische Suche adaptiert, um Szenarien zu optimieren. Hierfür muss ein passender Suchraum und eine geeignete Gütefunktion aufgestellt werden. Ausgehend von einer abstrakten Beschreibung der Funktionalität und Anwendungsfälle des Systems werden parametrisierte Szenarien abgeleitet. Deren Parameter spannen einen Suchraum auf, aus dem die passenden Szenarien zu identifizieren sind. Hierfür werden suchebasierte Verfahren verwendet, die sich an einer Gütefunktion orientieren und somit die Szenarien identifizieren, in denen das System das schlechteste Verhalten zeigt. Bei geeigneter Ableitung einer solchen Gütefunktion kann die Basis geschaffen werden für eine Argumentation über Vollständigkeit der Tests und über die Qualität des Systemverhaltens. Außerdem wird ein zielgerichtetes Testen mit automatischer Testfallauswertung ermöglicht.

## 1 Motivation

Das Streben nach autonomen Fahrsystemen resultiert in immer komplexeren und umfangreicheren Systemen. Prototypen unterschiedlicher Automatisierungsgrade und verschiedener Unternehmen sind in Form von Testfahrzeugen im Realverkehr unterwegs und erzielen vielversprechende Ergebnisse.

Dennoch gibt es auf dem Weg zur Serienreife noch einige Hürden. Besonders die Absicherung solcher Systeme stellt aufgrund der Systemkomplexität und der potentiell unendlichen Szenarien eine große Herausforderung dar [KW17]. Auch wenn Realtestfahrten weiterhin sehr wichtig für Verifikation und Validierung sein werden, reicht die reale Absicherung allein nicht aus [WW16][HWKK15][ZP]. Stattdessen soll verstärkt virtuell in einem szenariobasierten closed-loop Aufbau getestet werden [USH<sup>+</sup>17]. Die Identifikation und Wahl von passenden Testsznarien ist extrem schwierig und eine der wesentlichen Problemstellungen.

Da keine konkrete und präzise Beschreibung existiert, was ein autonomes Fahrzeug im Detail können muss [LHL<sup>+</sup>16], kommt es zu vielen Herausforderungen bei der Testfallerzeugung, -auswahl und -auswertung [BNSB16][SZS<sup>+</sup>15]. Aktuelle Testmethodiken reichen nicht aus [BDF<sup>+</sup>14][HSS16] und Verbesserungen sind notwendig [HPT10][HWLZ16]

(siehe Abschnitt 3). Spätestens bei höherautomatisierten Fahrsystemen werden Teile der Funktionalität von maschinell erlernten Funktionen bereitgestellt, welche mit inhärenten Schwierigkeiten bei der Whitebox-Verifikation verbunden sind [BGH17]. Somit liegt der Fokus vor allem auf Blackbox-Verfahren, wofür suchebasierte Verfahren häufig als vielversprechende Technologie sowohl in Theorie und Praxis vorgeschlagen werden [SNF14]. Existierende Arbeiten zur Verwendung von suchebasierten Verfahren für den Test von Fahrfunktionen fokussieren sich auf die technologischen Aspekte und nehmen den für die Suche vorausgesetzten Suchraum und die passende Gütefunktion als gegeben an oder erstellen diese ad-hoc (siehe Abschnitt 3). Das lässt methodische Aspekte unberücksichtigt. Tatsächlich sind die Ableitung des Suchraums und der Qualitätsfunktion aber sehr schwierig, was bei nicht korrekter Durchführung in falschen Schlussfolgerungen über die Testergebnisse und die Sicherheit des Systems resultieren kann. Es ist außerordentlich wichtig, dass vielversprechende Technologien wie suchebasierte Verfahren in einer strukturierten Methodik angewandt werden.

In dieser Arbeit wird eine solche Methodik für den Test von automatisierten und autonomen Fahrsysteme präsentiert. Dabei werden suchebasierte Verfahren adaptiert, indem die Mittel zur Ableitung von Suchräumen und Zielfunktionen gegeben werden. Ausgehend von abstrakten, funktionalen Beschreibungen wird beides, also Suchraum und Zielfunktion, erstellt. Sie dienen als Startpunkt für die heuristische Suche, um „gute“ Testfälle zu erzeugen. So werden konstruktiv funktionalitätsspezifische Extremszenarien identifiziert, in denen das System das schlechteste Verhalten zeigt.

Abschnitt 2 beschreibt den Grundgedanken des szenariobasierten Testens. In Abschnitt 3 wird ein Überblick über existierende Arbeiten gegeben, bevor anschließend in Abschnitt 4 die heuristische Suche beschrieben wird. Nachdem in Abschnitt 5 die Methodik vorgestellt wurde, wird deren Anwendung für die Systemfreigabe in Abschnitt 6 diskutiert. In Abschnitt 7 wird die Arbeit zusammengefasst.

## 2 Szenariobasiertes Testen

### 2.1 Szenarien

Das bereits verbreitete Konzept der funktionalen, logischen und konkreten Szenarien [BMRM17] wird für diese Arbeit aufgegriffen (siehe auch Abbildung 1). Funktionale Szenarien sind textuelle Beschreibungen von abstrakten Anwendungsfällen, in denen das System seine Funktionalität bereitzustellen hat [BMRM17]. Deren Detaillierung sind die logischen Szenarien, welche Betriebsszenarien durch Entitäten und deren Beziehungen mithilfe von Parameterbereichen darstellen [BMRM17]. Dies wird in dieser Arbeit wie folgt formalisiert: Ein logisches Szenario wird als Tripel  $(D, V, I)$  definiert. Dabei enthält  $D$  fixierte Informationen, welche den Rahmen der Szenarien darstellen. Die Variablen  $v_i$  in  $V$  ( $|V| = n$ ) stellen Parameter dar, von denen jeder im zugehörigen Intervall  $I_i \in I$  variiert werden kann. Wird jedem  $v_i, i = 1, \dots, n$  ein Wert zugeordnet, so erhält man eine Belegung für dieses logische Szenario. Eine solche Belegung entspricht einem konkreten Szenario. Diese sind die finalen, ausführbaren Szenarien [BMRM17]. Mithilfe der Intervalle spannen die logischen Szenarien einen potentiell unendlichen Raum

$A = I_1 \times I_2 \times \dots \times I_n \subset \mathbb{R}^n$  an möglichen konkreten Szenarien auf.

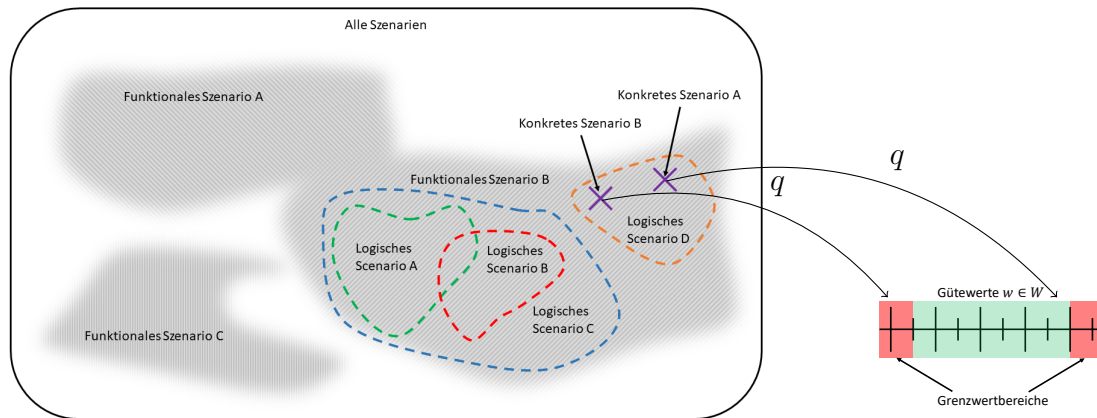


Abbildung 1: Relationen zwischen funktionalen, logischen und konkreten Szenarien sowie die Zuordnung einer Güte zu einem konkreten Szenario

## 2.2 Limit Testing

Man betrachte eine Funktion  $q : A \rightarrow W$ , welche jedem konkreten Szenario eine Güte  $w \in W$  zuordnet. Diese ist üblicherweise von dem Verhalten des Systems unter Test (SUT) abhängig, was bedeutet, dass das SUT in einem konkreten Szenario beobachtet wird und die Beobachtung (z.B. Simulationsergebnis) als Grundlage der Güteberechnung dienen. Das Ziel von Limit Testing ist es, die Szenarien abzutesten, für die  $q$  Grenzwerte zurückliefert (siehe auch linker Pfeil in Abbildung 1).

## 2.3 „Gute“ Testfälle

Ein Testfall besteht generell aus einer Eingabe, einer erwarteten Ausgabe sowie Umgebungsbedingungen. Testszenarien (hier: konkrete Szenarien) bilden die Eingabe und die Umgebungsbedingungen. Die erwartete Ausgabe kann für die hier vorliegenden, kontinuierlichen Systeme nicht durch einen einzelnen Wert dargestellt werden, mit dem auf Gleichheit verglichen wird, sondern wird mithilfe von Bereichsangaben angegeben.

In dem hier betrachteten Kontext handelt es sich dabei um einen sicheren Operationsbereich (siehe Abbildung 2) [KW16]. In diesem darf das System frei handeln und solange es diesen Bereich nicht verlässt, wird es als sicher bezeichnet. Vor allem bei (möglicherweise nichtdeterministischen) Systemen, welche z.B. auf maschinellem Lernen aufbauen, spielt dies eine entscheidende Rolle, da im Vornherein nicht für jede Situation ein bestimmtes Verhalten exakt beschrieben werden kann. Im Sinne von Limit Testing wird ein *guter* Testfall wie folgt definiert (siehe [Pre15]):

*Ein guter Testfall kann potentiell fehlerhaftes Systemverhalten aufzeigen. Dies bedeutet, dass sich in einem guten Testszenario ein korrektes System den Grenzen*

des sicheren Operationsbereiches annähert und ein fehlerhaftes System diese überschreitet.

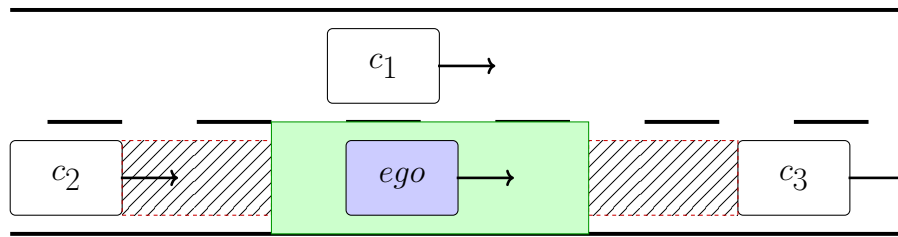


Abbildung 2: Beispiel für einen sicheren Operationsbereich (grün), eingeschränkt durch notwendige Abstände nach vorne und hinten (rot) sowie einem Fahrzeug auf der angrenzenden Spur ( $c_1$ )

### 3 Existierende Arbeiten

Verschiedene Ansätze existieren in dieser Domäne. Ein sehr praktikabler Ansatz ist die Extraktion von konkreten Szenarien aus Sensordaten von Realfahrten [MKK15][ZKK<sup>+</sup>15] [LSK13][WKRL17][ZP] oder aus der Simulation [SBW<sup>+</sup>16] [ZKK<sup>+</sup>16]. Dabei stammt die Umgebung in den Szenarien aus der Realität, wird manuell erstellt oder automatisiert erzeugt [KKDS17][CLP<sup>+</sup>15]. Die resultierende Menge an Testszenarien kann dann anhand von Kritikalitätsheuristiken und -metriken gefiltert werden [JSW17][WJWW16]. Es kann allerdings sein, dass unter den aufgezeichneten Szenarien keine besonders *guten* Testszenarien enthalten sind oder dass die enthaltenen Szenarien nicht zielgerichtet eine spezielle Funktionalität des Gesamtsystems testen. Darüber hinaus impliziert eine hohe Kritikalität nicht unbedingt, dass das jeweilige Szenario ein guter Testfall ist, da sich das System korrekt oder inkorrekt, sowohl in nicht-kritischen als auch in kritischen Szenarien, verhalten kann. Hinzu kommt, dass bei den meisten der genannten Ansätze die entstehende Menge an Szenarien nur als Open-Loop-Szenarien verwendet werden können, da eine Umwandlung in Closed-Loop-Szenarien - abhängig vom Anwendungsfall - nur sehr schwer oder überhaupt nicht möglich ist.

Ein Datenbankansatz [PZK<sup>+</sup>17][PZBE17] verwendet Aufzeichnungen von Realfahrten und deren Zuordnung zu zuvor spezifizierten, logischen Szenarien [RFUE16][RSZ<sup>+</sup>17]. Durch eine ausreichend große Menge solcher Aufzeichnungen wird eine aussagekräftige Verteilung und Beschreibung des Realverkehrs gewonnen. Entsprechend dieser Verteilungen werden dann für den Test von neuen Systemen Parameterbelegungen gewählt, um so die häufigsten Szenarien ausreichend abzudecken. Es kommt zu den bereits genannten Problemen für Realfahrtaufzeichnungen, z.B. dass zwar viele Standardszenarien, aber wenige schwierige, das System besonders fordernde Szenarien getestet werden.

Um die Anzahl an Szenarien möglicherweise zu reduzieren, werden beobachterbasierte Ansätze vorgeschlagen [OBW<sup>+</sup>18][Tat16]. Diese erfordern eine spezifizierte Vorbedingung, welche den Beobachter aktiviert, und ein erwartetes Verhalten als Nachbedingung.

Konkrete Szenarien werden dann so ausgewählt, dass eine hohe Abdeckung im Suchraum erzielt wird. Tatsächlich ist es jedoch überaus schwierig - wenn nicht unmöglich - diese Vorbedingungen so zu spezifizieren, dass keine Fälle vom Beobachter übersehen werden, aber auch nicht zu viele Fehlidentifikationen getätigt werden. Außerdem kann die Abdeckung eventuell nur schwer für eine große Anzahl an Szenarien erzielt werden. Eine hohe Abdeckung im Suchraum bedeutet außerdem nicht zwangsläufig, dass „gute“ Testsznarien gefunden werden. Darüber hinaus ist die Herkunft und methodische Erstellung der Bedingungen unklar.

Im Bereich der suchebasierten Verfahren in der Domäne der Fahrsysteme präsentiert die initiale Forschung die Anwendung von suchebasierten Verfahren für den funktionalen Test eines Parkassistenten [BW03] und eines Bremsassistenten [BW05][BW08] (darauf aufbauend: [HDN17]). In der Folge von technologischen Weiterentwicklungen wurde die Anwendung für verschiedene Steuergeräte gezeigt, basierend auf optimierten Eingangssignalen [PCG05][VLW<sup>+</sup>13][HPSG17][HSB<sup>+</sup>16][MNB17][DHJ<sup>+</sup>17]. Kürzlich wurde ein weiterer technologischer Schritt veröffentlicht, welcher maschinelles Lernen verwendet, um die Szenariooptimierung für einen Notbremsassistenten zu beschleunigen [ANBS16][NBS<sup>+</sup>18][BSH17]. Diese Arbeiten fokussieren sich auf den technologischen Aspekt der Optimierung und beschreiben lediglich für die präsentierten Fallstudien, wie dieser Ansatz angewandt werden kann. Die zugrundeliegende Methodik für die Erstellung des Suchproblems und der Zielfunktion wird nicht beschrieben. Ebenfalls bleibt der Aspekt der Vollständigkeit als auch die Einbindung eines automatischen Orakels unberücksichtigt.

#### 4 Suche-basierte Verfahren

Suchebasierte Verfahren sind mit den gegebenen Inputfaktoren – Suchraum und Gütefunktion – ein passendes Werkzeug, um das Element im Suchraum zu identifizieren, für das die Gütefunktion den (global) besten Wert zurückliefert. Im Kontext der Szenarien stellen  $A \subset \mathbb{R}^n$  den Suchraum und  $q$  die Gütefunktion dar. Als Ausgangspunkt dienen manuell oder zufällig ausgewählte oder bereits zuvor identifizierte, konkrete Szenarien. Nach der Simulation eines Szenarios werden die Simulationsergebnisse für die Berechnung der Güte des Szenarios bzw. des Systemverhaltens verwendet. Basierend auf den Gütewerten wählt der Optimierer anhand einer Heuristik neue konkrete Szenarien aus. Dies wird fortgesetzt, bis keine Szenarien mit besserer Güte mehr gefunden werden, eine maximale Iterationszahl erreicht oder die zur Verfügung gestellte Rechenzeit aufgebraucht ist.

Eine Vielzahl an Verfahren und Heuristiken stehen zur Verfügung. Für diese Arbeit sind vor allem die globalen Optimierungsverfahren von Bedeutung, da - so gut wie technisch möglich - das Szenario gefunden werden soll, welches das globale Optimum darstellt. Abhängig von der Anzahl an Optimierungszielen können herkömmliche, durch die Natur und die Physik inspirierte Einzel- oder Mehrzielverfahren verwendet werden, z.B. die weitverbreiteten [FP15] genetischen Algorithmen [Mit95] (Mehrzielversion: NSGA-II [DPAM02]) und Partikelschwarmalgorithmen [KE95]. Welche Heuristik für den gegebenen Suchraum und die gegebene Gütefunktion das globale Optimum am schnellsten und zuverlässigsten findet, kann meist a-priori nicht bestimmt werden. Abhängig von den im Testprozess vorhandenen Ressourcen und der Dauer eines Simulationslaufes können

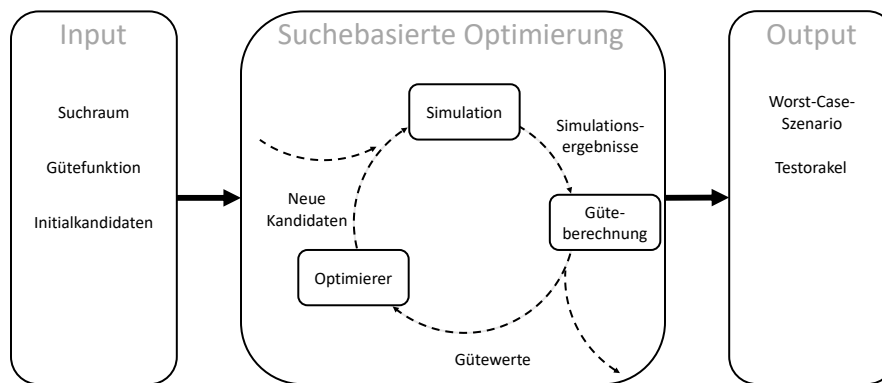


Abbildung 3: Anwendung von suchebasierten Verfahren auf Suchraum und Gütefunktion, um Testszenarien zu erhalten

unterschiedliche Arten von Verfahren gewählt werden. Für sehr begrenzte Ressourcen und besonders teure Simulationsläufe, z.B. da deutlich langsamer als in Echtzeit simuliert wird, können sequentielle Verfahren verwendet werden, die möglichst viele Informationen über den Suchraum sammeln. Diese sind nicht immer parallelisierbar. Falls ausreichend Ressourcen vorhanden sind und einzelne Simulationsläufe eher schnell ausgeführt werden können, bieten sich Verfahren an, die stark parallelisierbar sind.

## 5 Methodik

Existierende Arbeiten zeigen, dass suchebasierte Verfahren als Technologie in konkreten Anwendungsfällen einsetzbar sind. Die Vorstellung folgender Methodik soll nicht nur motivieren, diese Technologien als Ergänzung oder anstelle von anderen Ansätzen zu verwenden, sondern auch eine Herangehensweise bieten, um diese Technologien zielgerichtet einzusetzen, sodass eine Basis zur Argumentation über funktionale Korrektheit entsteht.

Da das vollständige Testen aller möglichen Szenarien nicht praktikabel ist, schlägt die hier vorgestellte Methodik vor, die bewährten Prinzipien der Äquivalenzklassen- und Grenzwertanalyse anzuwenden, indem zunächst die möglichen Szenarien nach Anwendungsfällen (z.B. Spurwechsel) in logische Szenarien gruppiert werden (Äquivalenzklassen) und anschließend für jedes logische Szenario diejenigen konkreten Szenarien identifiziert werden, für die das System mithilfe der definierten Gütefunktion am schlechtesten bewertet wird (Grenzwertbetrachtung). Diese Szenarien werden als Worst-Case-Szenarien bezeichnet. Falls das System den sicheren Operationsbereich auch in den Worst-Case-Szenarien nicht verlässt, so wird angenommen, dass dies auch in den anderen konkreten Szenarien dieses Suchraums der Fall ist. Das System wird also als sicher bezeichnet. Ein passender Suchraum und eine passende Gütefunktion sind nun so abzuleiten, dass diese Argumentation so gut wie technisch möglich unterstützt wird.

Als Ausgangspunkt dienen abstrakte Informationen über das SUT, welche idealerweise in Form einer Anforderungsspezifikation oder funktionalen Spezifikation vorliegt (siehe Abbildung 4). Zusammen mit Informationen aus Sicherheitsanalysen können funktionale Szenarien abgeleitet werden.

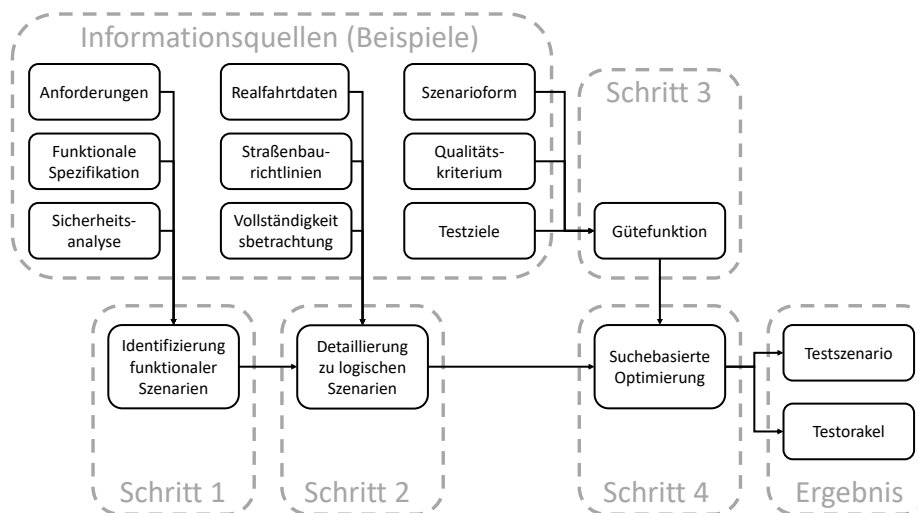


Abbildung 4: Überblick über die einzelnen Schritte der Methodik

Diese werden anschließend zu logischen Szenarien detailliert. Da diese den Suchraum darstellen, sind bei der Detaillierung verschiedene Aspekte zu beachten. Die fixierten Informationen zusammen mit den Variablen sollten ausreichen, damit Aussagen über das Szenario und das Systemverhalten getroffen werden können, gleichzeitig sollten diese jedoch nicht zu einschränkend gewählt werden, um keine möglichen konkreten Szenarien zu unterschlagen. Die Anzahl der Variablen sollte durch passend gewählte fixierte Informationen konstant und möglichst gering gehalten werden, um damit auch den Suchraum konstant und möglichst einfach zu halten, was die nachfolgende Suche nach dem Optimum deutlich vereinfachen kann. Für eine geeignete Wahl der Intervallgrenzen kann z.B. eine Datenanalyse von Realfahrten durchgeführt werden. Zusätzlich können (inter)nationale Richtlinien als Orientierung dienen. Beispielsweise enthält das „European Agreement on Main International Traffic Arteries“ Informationen über das Aussehen von Autobahnen. Sollten sich logische Szenarien so sehr überschneiden, dass der Suchraum des einen im Suchraum des anderen (fast) vollständig enthalten ist, kann entsprechend versucht werden, die Anzahl an logischen Szenarien zu reduzieren.

Nachdem der sichere Operationsbereich durch passende Abstände oder Metriken beschrieben ist, kann eine Gütefunktion für das System aufgestellt werden. Diese kann aus mehreren Teilen zusammengesetzt werden, wobei einige Teilfunktionen mithilfe von entsprechenden Güteberechnungen sicherstellen, dass das Szenario durch die Optimierung in die gewünschte Form gebracht wird. Unter allen Szenarien, welche diese Form aufweisen, wird daraufhin das Szenario gesucht, in dem sich das System am nächsten an die Grenzen des sicheren Operationsbereichs annähert (korrektes System) oder diese Grenzen am weitesten überschreitet (fehlerhaftes System). Im aufgestellten Suchraum kann so durch Optimierung anhand der Gütefunktion nach dem Worst-Case-Szenario gesucht werden. Verlässt das System in diesem konkreten Szenario den sicheren Operationsbereich nicht, so wird angenommen, dass das System für alle konkreten Szenarien, die durch das logische Szenario dargestellt werden, sicher ist.

## 6 Systemvergleich und -freigabe

Durch methodisch korrekte Verwendung von suchebasierten Verfahren erhält man eine Liste an Gütewerten für das schlechteste Systemverhalten für unterschiedliche logischen Szenarien. Neben der Fehlerdetektion erlaubt dies auch den Vergleich von Systemen, da die Güte von bestimmten Systemfunktionen messbar gemacht wird. Beim Regressionstest bietet dies die Grundlage für eine Schlussfolgerung, ob das System ausschließlich durch Fehlerbehebung besser geworden ist oder ob an anderer Stelle ein Problem verursacht wurde. Das ist vor allem interessant, wenn erwünschte Anforderungen zu konkurrierenden Systemeigenschaften führen und ein guter Kompromiss erzielt werden muss.

Wie oben beschrieben sollen Systeme nicht den sicheren Operationsbereich verlassen, welcher durch Distanzen in Raum und Zeit beschrieben werden kann. Da eine Worst-Case-Betrachtung durchgeführt wird, wäre das beste System eines, welches immer den größtmöglichen Abstand einhält. Vor allem in dichtem Verkehr würde dies zu einem extrem zurückhaltendem Verhalten führen, was im schlimmsten Fall dazu führt, dass z.B. ein Spurwechsel unmöglich wird, da keine Lücke zwischen den anderen Autos groß genug ist. Es muss also auch überprüft werden, ob das System den vorhandenen Spielraum sinnvoll nutzt, wofür ebenfalls die vorgestellte Methodik verwendet werden kann. Es wird z.B. nach der größten Lücke auf der benachbarten Spur gesucht, bei der sich das System gegen einen Spurwechsel entscheidet.

Neben Systemversionen können aber auch verschiedene Systemvarianten verglichen werden, bei denen für die gleiche Funktionalität unterschiedliche Lösungsansätze eingesetzt werden. Für die Freigabe eines Systems können gewünschte Ziele für die einzelnen Funktionalitäten gesetzt und über die Güte überprüft werden. Da das Worst-Case-Szenario abhängig vom System ist, wird in dieser Arbeit die Verwendung einer Liste an logischen Szenarien mit den jeweiligen Gütefunktionen anstatt einer klassischen Liste von konkreten Szenarien vorgeschlagen. Tatsächlich kann es sein, dass ein konkretes Szenario, welches bei einem System ein Fehlverhalten identifiziert hat, für ein geringfügig anderes System völlig unbrauchbar ist.

## 7 Zusammenfassung

Zu Beginn wurde beschrieben, dass die Adaption von suchebasierten Verfahren für das Testen von automatisierten und autonomen Fahrsystemen schwierig ist. Diese Techniken sind inhärent abhängig von einer passenden Gütefunktion und einem passenden Suchraum, deren beider Herkunft unklar ist. Zusätzlich soll eine Argumentation über die funktionale Korrektheit und über die Vollständigkeit des Testvorgangs unterstützt werden. Diese Arbeit beschreibt eine Methodik zur Verwendung solcher Verfahren, indem gezeigt wird, wie Systemverhalten bewertbar gemacht werden kann und worauf beim Erstellen des Suchproblems und beim Ableiten einer Gütefunktion zu achten ist. Es liegt in der Natur des Testens, dass keine Garantien für die Abwesenheit von Fehlern gegeben werden können, jedoch bieten die präsentierten Worst-Case-Szenarien eine Basis für eine solche Argumentation.



## Literatur

- [ANBS16] Raja Ben Abdesslem, Shiva Nejati, Lionel C Briand und Thomas Stifter. Testing advanced driver assistance systems using multi-objective search and neural networks. In *Automated Software Engineering (ASE), 2016 31st IEEE/ACM International Conference on*, Seiten 63–74. IEEE, 2016.
- [BDF<sup>+</sup>14] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller und Hermann Winner. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4):6–22, 2014.
- [BGH17] Simon Burton, Lydia Gauerhof und Christian Heinzemann. Making the Case for Safety of Machine Learning in Highly Automated Driving. In *International Conference on Computer Safety, Reliability, and Security*, Seiten 5–16. Springer, 2017.
- [BMRM17] Gerrit Bagschik, Till Menzel, Andreas Reschka und Markus Maurer. Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen. In *11. Workshop Fahrerassistenzsysteme. Hrsg. von Uni-DAS e. V.*, Seiten 125–135, 2017.
- [BNSB16] Lionel Briand, Shiva Nejati, Mehrdad Sabetzadeh und Domenico Bianculli. Testing the untestable: model testing of complex software-intensive systems. In *Proceedings of the 38th international conference on software engineering companion*, Seiten 789–792. ACM, 2016.
- [BSH17] Halil Beglerovic, Michael Stolz und Martin Horn. Testing of autonomous vehicles using surrogate models and stochastic optimization. In *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, Seiten 1–6. IEEE, 2017.
- [BW03] Oliver Bühler und Joachim Wegener. Evolutionary functional testing of an automated parking system. In *Proceedings of the International Conference on Computer, Communication and Control Technologies (CCCT'03) and the 9th. International Conference on Information Systems Analysis and Synthesis (ISAS'03), Florida, USA*, 2003.
- [BW05] Oliver Bühler und Joachim Wegener. Evolutionary functional testing of a vehicle brake assistant system. In *6th Metaheuristics International Conference, Vienna, Austria*, 2005.
- [BW08] Oliver Bühler und Joachim Wegener. Evolutionary functional testing. *Computers & Operations Research*, 35(10):3144–3160, 2008.
- [CLP<sup>+</sup>15] Carlos Campos, João Miguel Leitão, João Paulo Pereira, António Ribas und António Fernando Coelho. Procedural generation of topologic road networks for driving simulation. In *Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on*, Seiten 1–6. IEEE, 2015.

- [DHJ<sup>+</sup>17] Jyotirmoy Deshmukh, Marko Horvat, Xiaoqing Jin, Rupak Majumdar und Vinayak S Prabhu. Testing cyber-physical systems through bayesian optimization. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):170, 2017.
- [DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal und TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [FP15] Robert Feldt und Simon Poulding. Broadening the search in search-based software testing: It need not be evolutionary. In *Search-Based Software Testing (SBST), 2015 IEEE/ACM 8th International Workshop on*, Seiten 1–7. IEEE, 2015.
- [HDN17] Thomas Hierlinger, Tobias Dirndorfer und Tobias Neuhauser. A Method for the Simulation-Based Parameter Optimization of Autonomous Emergency Braking Systems. In *25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, 2017.
- [HPSG17] Florian Hauer, Alexander Pretschner, Maximilian Schmitt und Markus Groetsch. Industrial Evaluation of Search-Based Test Generation Techniques for Control Systems. In *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Seiten 5–8. IEEE, 2017.
- [HPT10] Falke Hendriks, Riné Pelders und Martijn Tideman. Future testing of active safety systems. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 3(2010-01-2334):170–175, 2010.
- [HSB<sup>+</sup>16] Dominik Holling, Alvin Stanescu, Kristian Beckers, Alexander Pretschner und Matthias Gemmar. Failure Models for Testing Continuous Controllers. In *Software Reliability Engineering (ISSRE), 2016 IEEE 27th International Symposium on*, Seiten 365–375. IEEE, 2016.
- [HSS16] Philipp Helle, Wladimir Schamai und Carsten Strobel. Testing of Autonomous Systems—Challenges and Current State-of-the-Art. In *INCOSE International Symposium*, Jgg. 26, Seiten 571–584. Wiley Online Library, 2016.
- [HWKK15] Thomas Helmer, Lei Wang, Klaus Kompass und Ronald Kates. Safety performance assessment of assisted and automated driving by virtual experiments: Stochastic microscopic traffic simulation as knowledge synthesis. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, Seiten 2019–2023. IEEE, 2015.
- [HWLZ16] WuLing Huang, Kunfeng Wang, Yisheng Lv und FengHua Zhu. Autonomous vehicles testing methods review. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, Seiten 163–168. IEEE, 2016.
- [JSW17] Philipp Junietz, Jan Schneider und Hermann Winner. Metrik zur Bewertung der Kritikalität von Verkehrssituationen und -szenarien. In *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, 2017.

- [KE95] James Kennedy und Russell Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Jgg. 4, Seiten 1942–1948. IEEE, 1995.
- [KKDS17] Baekgyu Kim, Yusuke Kashiba, Siyuan Dai und Shinichi Shiraishi. Testing Autonomous Vehicle Software in the Virtual Prototyping Environment. *IEEE Embedded Systems Letters*, 9(1):5–8, 2017.
- [KW16] Philip Koopman und Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.
- [KW17] Philip Koopman und Michael Wagner. Autonomous Vehicle Safety: An Interdisciplinary Challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017.
- [LHL<sup>+</sup>16] Li Li, Wu-Ling Huang, Yuehu Liu, Nan-Ning Zheng und Fei-Yue Wang. Intelligence Testing for Autonomous Vehicles: A New Approach. *IEEE Transactions on Intelligent Vehicles*, 1(2):158–166, 2016.
- [LSK13] Ulrich Lages, Martin Spencer und Roman Katz. Automatic scenario generation based on laserscanner reference data and advanced offline processing. In *Intelligent Vehicles Symposium Workshops (IV Workshops), 2013 IEEE*, Seiten 146–148. IEEE, 2013.
- [Mit95] Melanie Mitchell. Genetic algorithms: An overview. *Complexity*, 1(1):31–39, 1995.
- [MKK15] Pascal Minnerup, Tobias Kessler und Alois Knoll. Collecting Simulation Scenarios by Analyzing Physical Test Drives. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, Seiten 2915–2920. IEEE, 2015.
- [MNB17] Reza Matinnejad, Shiva Nejati und Lionel Briand. Automated testing of hybrid Simulink/Stateflow controllers: industrial case studies. In *Proceedings of 11th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering (ESEC/FSE 2017)*, 2017.
- [NBS<sup>+</sup>18] Shiva Nejati, Lionel Briand, Thomas Stifter et al. Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms. In *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018)*. ACM, 2018.
- [OBW<sup>+</sup>18] Stefan Otten, Johannes Bach, Christoph Wohlfahrt, Christian King, Jan Lier, Hermann Schmid, Stefan Schmerler und Eric Sax. Automated Assessment and Evaluation of Digital Test Drives. In *Advanced Microsystems for Automotive Applications 2017*, Seiten 189–199. Springer, 2018.
- [PCG05] Hartmut Pohlheim, Mirko Conrad und Arne Griep. Evolutionary safety testing of embedded control software by automatically generating compact test data sequences. Bericht, SAE Technical Paper, 2005.

- [Pre15] Alexander Pretschner. Defect-Based Testing. In: Dependable Software Systems Engineering, 2015.
- [PZBE17] Andreas Pütz, Adrian Zlocki, Julian Bock und Lutz Eckstein. System validation of highly automated vehicles with a database of relevant traffic scenarios. *12th ITS European Congress*, 2017.
- [PZK<sup>+</sup>17] Andreas Pütz, Adrian Zlocki, Jörg Küfen, Julian Bock und Lutz Eckstein. Database Approach for the Sign-Off Process of Highly Automated Vehicles. In *25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, 2017.
- [RFUE16] Christian Roesener, Felix Fahrenkrog, Axel Uhlig und Lutz Eckstein. A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Seiten 1360–1365. IEEE, 2016.
- [RSZ<sup>+</sup>17] Christian Roesener, Jan Sauerbier, Adrian Zlocki, Felix Fahrenkrog, Lei Wang, András Várhelyi, Erwin de Gelder, Joris Dufils, Sandra Breunig, Pablo Mejuto et al. A Comprehensive Evaluation Approach for Highly Automated Driving. In *25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, 2017.
- [SBW<sup>+</sup>16] Christoph Sippl, Florian Bock, David Wittmann, Harald Altinger und Reinhard German. From simulation data to test cases for fully automated driving and ADAS. In *IFIP International Conference on Testing Software and Systems*, Seiten 191–206. Springer, 2016.
- [SNF14] Michal Sroka, Roman Nagy und Dominik Fisch. Genetic Algorithms in Test Design Automation. *Applied Mechanics & Materials*, (693), 2014.
- [SZS<sup>+</sup>15] Jan Erik Stellet, Marc René Zofka, Jan Schumacher, Thomas Schamm, Frank Niewels und J Marius Zöllner. Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, Seiten 1455–1462. IEEE, 2015.
- [Tat16] Mugur Tatar. Test and Validation of Advanced Driver Assistance Systems Automated Search for Critical Scenarios. *ATZelektronik worldwide*, 11(1):54–57, 2016.
- [USH<sup>+</sup>17] Simon Ulbrich, Fabian Schuldt, Kai Homeier, Michaela Steinhoff, Till Menzel, Jens Krause und Markus Maurer. Testing and validating tactical lane change behavior planning for automated driving. In *Automated Driving*, Seiten 451–471. Springer, 2017.
- [VLW<sup>+</sup>13] Tanja EJ Vos, Felix F Lindlar, Benjamin Wilmes, Andreas Windisch, Arthur I Baars, Peter M Kruse, Hamilton Gross und Joachim Wegener. Evolutionary functional black-box testing in an industrial setting. *Software Quality Journal*, 21(2):259–288, 2013.

- [WJWW16] Walther Wachenfeld, Philipp Junietz, Raphael Wenzel und Hermann Winner. The worst-time-to-collision metric for situation identification. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, Seiten 729–734. IEEE, 2016.
- [WKRL17] Christian Wolschke, Thomas Kuhn, Dieter Rombach und Peter Liggesmeyer. Observation Based Creation of Minimal Test Suites for Autonomous Vehicles. In *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Seiten 294–301. IEEE, 2017.
- [WW16] Walther Wachenfeld und Hermann Winner. The release of autonomous vehicles. In *Autonomous Driving*, Seiten 425–449. Springer, 2016.
- [ZKK<sup>+</sup>15] Marc René Zofka, Florian Kuhnt, Ralf Kohlhaas, Christoph Rist, Thomas Schamm und J Marius Zöllner. Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems. In *Information Fusion (Fusion), 2015 18th International Conference on*, Seiten 1422–1428. IEEE, 2015.
- [ZKK<sup>+</sup>16] Marc René Zofka, Sebastian Klemm, Florian Kuhnt, Thomas Schamm und J Marius Zöllner. Testing and validating high level components for automated driving: simulation framework for traffic scenarios. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, Seiten 144–150. IEEE, 2016.
- [ZP] Ding Zhao und Huei Peng. From the Lab to the Street: Solving the Challenge of Accelerating Automated Vehicle Testing. online at [https://mcity.umich.edu/wp-content/uploads/2017/05/Mcity-White-Paper\\_Accelerated-AV-Testing.pdf](https://mcity.umich.edu/wp-content/uploads/2017/05/Mcity-White-Paper_Accelerated-AV-Testing.pdf), retrieved 29th June 2017.