

# **Funktionaler Softwaretest für aktive Fahrerassistenzsysteme mittels parametrierter Szenario-Simulation**

Florian Schmidt, Eric Sax

electronics solutions  
MBtech Group  
Kolumbusstr. 2  
71063 Sindelfingen  
florian.schmidt@mbtech-group.com  
eric.sax@mbtech-group.com

**Abstract:** Im vorliegenden Beitrag wird das Testen der Embedded Software von umfelderfassenden aktiven Fahrerassistenzsystemen mit Hilfe einer bilderzeugenden Sensorstimulation für Hardware-in-the-Loop Testsysteme vorgestellt. Dies ermöglicht quantitative Aussagen zur Erreichen von Qualitätszielen für funktionale Tests mit hoher Testtiefe und -breite. Generierung und Aufbau der relevanten Test-Szenarien wird erläutert.

## **1 Software im Automobil**

Moderne Fahrzeuge beinhalten eine Fülle an Software-Komponenten. Diese sind üblicherweise als Embedded Software in Steuergeräten oder vernetzt in Steuergeräteverbänden implementiert (vgl. Abb. 1). Steuergeräte (SG), wie beispielsweise das Motor-SG, die Dachbedieneinheit, das Kombi-Instrument oder multifunktionale „Sensor-Aktor-Module“ kombinieren also Hardware (Sensoren, Aktoren) mit Funktionen, die in Form von Software realisiert sind (vgl. Abb. 2). Dabei sind häufig Echtzeitbedingungen zu erfüllen, um die korrekte Funktion (z.B. eines Airbagsteuergeräts) sicher zu stellen. Aktuelle Oberklassefahrzeuge haben über 80 verschiedene über Busleitungen vernetzte Steuergeräte. Aber nicht nur die Anzahl, auch die Komplexität ihrer Funktionen und damit der ablaufenden Software steigert sich kontinuierlich. Gründe dafür sind wachsende Kundenerwartungen, erhöhte Anforderungen z.B. an Sicherheitsstandards, und auch neue technische Möglichkeiten. [Gr05, Ch09]

Embedded Software im Automotive Bereich steht unter dem besonderen Druck, besonders hohe Anforderungen an die Zuverlässigkeit während eines ganzen Autolebens lang, und das unter allen Temperatur- und Klimabedingungen und Nutzungsverhalten.

Die ständige Interaktion der Software mit den anderen Verkehrsteilnehmern und besonders mit Menschen in ihrer Umgebung legen den benötigten hohen Qualitäts- und Sicherheitsstandard fest. Dazu kommen vielfältige verschiedene rechtliche Rahmenbedingungen, die Notwendigkeit einer einfachen Wartbarkeit, das Zusammenspiel von Hard- und Software sowie vieler Zulieferer für die einzelnen Elektronik-Komponenten sowie die Notwendigkeit von Ausfallsicherheit und Notlaufkonzepten selbst im Fehlerfall.

### Gesamtvernetzung BR 204.0

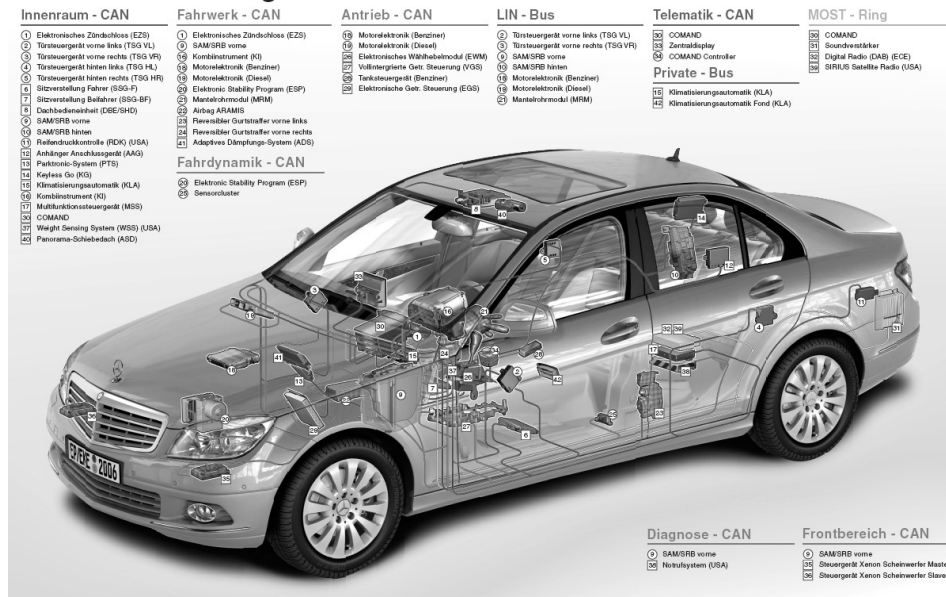


Abb. 1: Steuergeräte und Datenbusse der Daimler C-Klasse (Baureihe 204).

Automotive Software ist also durch hohe Komplexität, komfort- und sicherheitsrelevante fahrzeugspezifische Funktionen und daraus resultierend höchste Anforderungen an die Softwarequalität gekennzeichnet. [Li02]

### 1.1 Aktive Fahrerassistenzsysteme

Steuergeräte beinhalten Funktionen zur Entlastung des Fahrers, sei es durch Informationen, Warnungen oder aktive Unterstützung. Erst seit wenigen Jahren gibt es die so genannten Fahrerassistenzsysteme (FAS). Aktive Fahrerassistenzsysteme (engl. Advanced Driver Assistance Systems, ADAS) unterstützen dabei teilweise sogar durch aktive autonome Eingriffe. Beispiele hierfür sind Einpark- und Spurhalteassistenten, Fußgänger- und Verkehrszeichenerkennung. Diese Systeme greifen im Notfall selbständig durch Lenken oder (Not-) Bremsen direkt in die Längs- oder Querregelung des Fahrzeuges ein.

Die Umfelderkennung durch Sensoren wie Radar, Video oder Ultraschall ist dabei der übliche Weg, um den Algorithmen Informationen zur Situationsanalyse und -bewertung zu verschaffen. Als Sensordatenfusion bezeichnet man dabei das Vorgehen, immer mehrere Datenquellen redundant zu nutzen, um die Richtigkeit einer Entscheidung genauer abzusichern. Erst wenn Radar-Echos und Videobild unabhängig zum gleichen Analyse-Ergebnis führen, darf das Fahrzeug darauf aktiv reagieren.

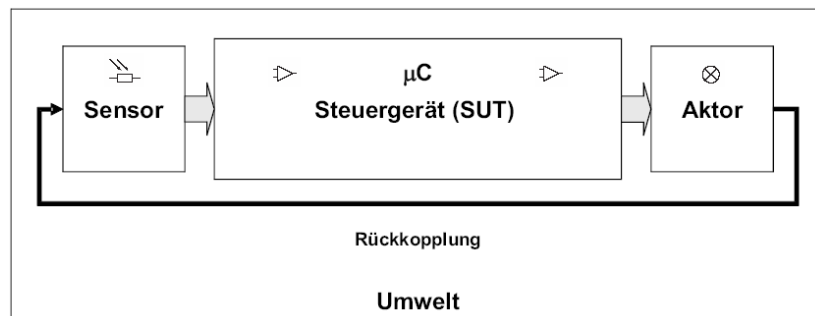


Abb. 2: Technischer Aufbau eines Steuergeräts als rückgekoppeltes System [Mü07].

Die Sicherheit aller Verkehrsteilnehmer ist das höchste Ziel bei der Entwicklung derart aktiv eingreifenden Funktionen. An die Software dieser Systeme werden aufgrund ihrer möglichen autonomen Aktionen besonders hohe Anforderungen gestellt. Neben einer sorgfältigen und professionellen Softwareentwicklung ist daher auch das Testen der Softwarequalität ein wichtiger Schritt auf dem Weg zum Serieneinsatz eines mit diesen Helfern ausgestatteten Fahrzeugs.

## 1.2 Heutige Vorgehensweise beim Softwaretest von ADAS

Derzeit wird Software und deren Funktionalität in Fahrerassistenzsystemen vielfach durch Testfahrten [ZS08] und einfache Sensorstimulationen getestet.

Testfahrten bieten den Vorteil, die reale Umgebung des Einsatzbereichs unter vielfältigsten Bedingungen als „Testdaten“ einzubinden. Damit einher geht der Nachteil, durch viele gefahrenen Kilometer zwar die Anzahl der geprüften Situationen („Use Cases“) zu erhöhen, jedoch lässt sich das Testen damit nicht allumfassend abdecken. Systematik, Automatisierbarkeit, Reproduzierbarkeit und Beeinflussbarkeit der Umgebung sind kaum gegeben, zudem sind aus Sicherheitsgründen viele Testfälle nicht durchführbar. Bei Kamerasystemen kann zum Beispiel das Einspielen von zuvor während realer Fahrten aufgezeichneter Sensordaten (z.B. Radar- oder Videodaten) in ein Testobjekt im Labor das Testspektrum erweitern und zu umfassenderen und teilweise genaueren Testergebnissen führen.

So können ohne Gefahr und mit beliebigen Wiederholungen vorher aufgenommene und nach Situationen (z.B. Tunnelfahrt, Landstraße, Fußgängerüberweg) klassifizierte Video-Dateien erneut wiedergegeben und damit abgefahren werden.

Jedoch zeigen sich auch hier zwei Nachteile: Zum einen bleibt eine Reaktion der Funktion (z.B. Notbremsung) ohne Reaktion auf die Sensoren (das Fahrzeug im Video fährt weiter). Zum anderen ist eine Sensordatenfusion nur dann möglich, wenn alle Sensordaten gleichzeitig in der Realität aufgenommen wurden – oder z.B. manuell in einem Videoobjekt markiert wurden, um daraus künstlich Radarechos zu errechnen.

Eine Aussage über den Reifegrad und die Erreichung der Qualitätsziele ist mit beiden Vorgehen nur eingeschränkt und höchstens qualitativ möglich. Auch wenn beide Verfahren ihre Berechtigung haben, so ist doch eine weitere Optimierung der Testverfahren wünschenswert.

### 1.3 Herausforderung

Um die Software von klassischen Steuergeräten durchgehend während des gesamten Entwicklungsprozesses zu testen, sind mehrere Standards etabliert. Das „V-Modell“ der Entwicklung kennt Modell-, Software-, Komponenten-, Integrations- und Gesamtfahrzeug-Tests. [Sa08]

Für die ersten vier Testobjekte (auch System-under-Test, SuT) wird üblicherweise die „X-in-the-Loop“ Technologie angewendet, bei der die Umgebung des Testobjekts simuliert und als Input stimuliert wird.

Die Reaktion des SuT geht wiederum direkt in die Umgebungssimulation ein. Diese „Loop“ kann auf professionellen Testsystemen in Echtzeit (Taktzyklen bis zu 200  $\mu$ s) ablaufen. Die Testfälle werden nach etablierten Testprozessen und Testspezifikationstechniken erstellt und mit Testautomatisierungssoftware wie PROVEtech:TA [MB09] der MBtech Group implementiert. Für klassische Steuergeräte entstehen dabei jeweils leicht mehrere tausend Testfälle um die Softwarequalität sowohl bzgl. Testbreite und -tiefe abzusichern.

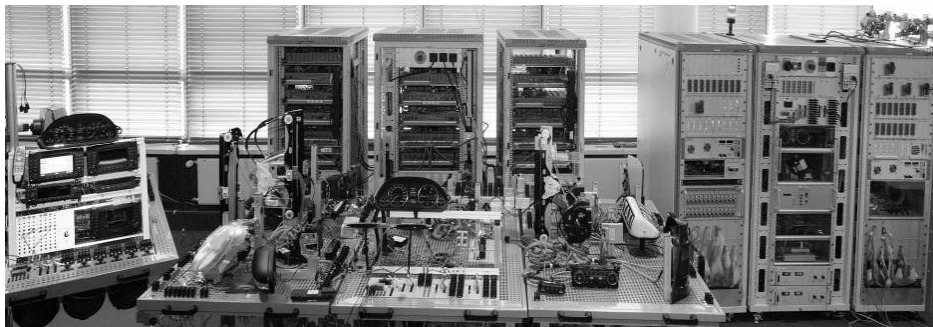


Abb. 3: HiL-Integrations-Testsystem für die gesamte Elektronik der Mercedes A-Klasse.

Der Vorteil der Model- (MiL), Software- (SiL) oder Hardware-in-the-Loop- (HiL, Abb. 3) Testsysteme liegt in der Reproduzierbarkeit der Tests, in der einfachen Möglichkeit automatisch Variationen von z.B. Parametern oder Randbedingungen durchzuführen wie auch im automatisierbaren Ablauf und damit Dauerbetrieb (24/7) der Tests. Diese

Vorteile führen zu einer besonders ausgeprägten Testtiefe und –breite. Daher ist es äußerst wünschenswert, auch aktive Fahrerassistenzsysteme (FAS) auf HiL-Testsystemen zu testen.

## **2 Lösungsansatz**

Um die Softwarequalität von aktiven FAS auch automatisiert im Laborbetrieb testen zu können, wurde ein Testsystem entwickelt, das direkt in bestehende Testumgebungen der sonstigen Fahrzeugelektronik integrierbar sowie für die verschiedenen Assistenz-Funktionalitäten anpassbar ist. PROVEtech:VL, die Visual-Loop-Komponente der PROVEtech Tool Suite, bietet offene Schnittstellen und erweiterbare Lösungen zum quantitativen, durchgängigen Testen. Ziel der Technologie-Entwicklung war der funktionale Test kamera-basierter Systeme mit dem Fokus auf einer Simulation der Testfälle im "bending knee"-Bereich der Systemperformance, also der für das FAS schwierigen Situationen.

### **2.1 Testsystem: PROVEtech:VL**

Das Testsystem um PROVEtech:VL (Abb. 4) besteht aus den Elementen Datenhaltung, Bediensystem, Testequipment und System-under-Test (SuT). Das SuT kann ein einzelnes Fahrerassistenzsystem-Steuergerät oder ein Steuergeräteverbund sein. Für den Fall der Sensordatenfusion ist die synchronisierte und konsistente Stimulation mehrerer Sensoren und Sensortypen mit identischen Umgebungsinformationen vorgesehen.

Diese Umgebungsinformationen werden aus verschiedenen Quellen importiert. Beispiele dafür sind Open Street Map (die für jeden frei nutzbare Wiki-Weltkarte) für Straßen- und Geoinformationsdaten sowie die Höhendaten der NASA SRTM<sup>1</sup>, um mit geringem Aufwand reale Strecken nachzubilden. Des Weiteren lassen sich beliebige 3D-Objekte wie Fahrzeuge, Häuser, Bäume, Personen etc. einfügen. Die Datenhaltung erfolgt in einer speziellen Datenbank. Die verwendeten Datenformate für Austausch-Dateien basieren auf offenen Standards wie OpenDRIVE [VI09].

Aus diesen Strecken-, Gelände-, und Objektdaten werden Testfälle zusammengestellt, die in Testszenarien kombiniert werden. Die Testszenarien werden dann durch einen hoch spezialisierten Rechner graphisch aufbereitet und über beliebige visuelle Schnittstellen (z.B. analoges Video, DVI oder Beamer, Leinwand und Kamera) in das SuT übertragen.

---

<sup>1</sup> Shuttle Radar Topography Mission, <http://www2.jpl.nasa.gov/srtm>

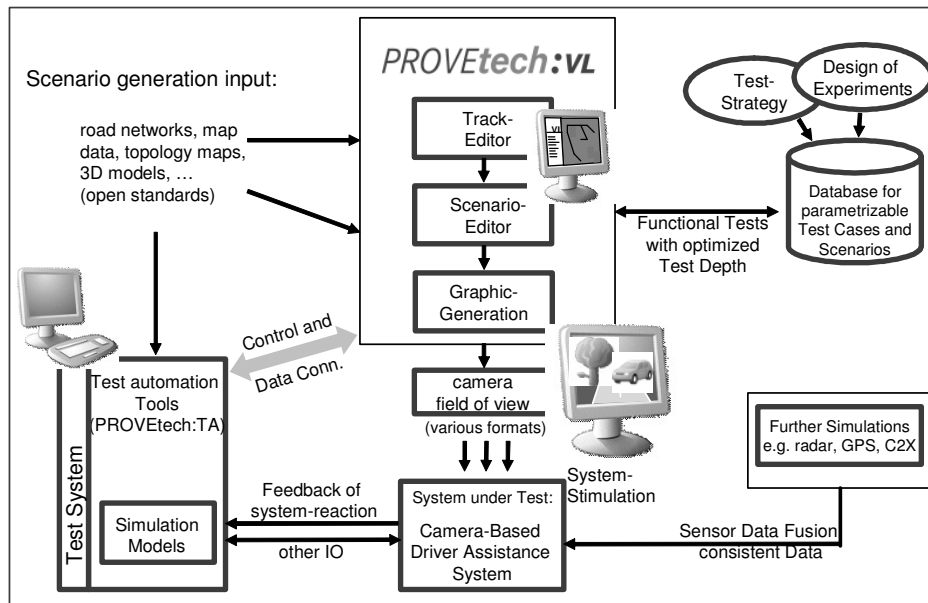


Abb. 4: Schematischer Überblick PROVEtech:VL.

Die Reaktionen des SuT werden in das Testequipment rückgekoppelt und fließen direkt in die Berechnung der Umgebungsmodelle ein. Diese Umgebungsmodelle liefern zusätzlichen Input für die Darstellung der Grafik, beispielsweise wird damit die exakte aktuelle Position der Fahrzeug-Kamera in Echtzeit berechnet, um die Kamera jederzeit mit aktuellen Bildern zu versorgen. Der „Visual Loop“ wird also geschlossen zwischen dem Fahrerassistenzsystem und dem HiL-Testsystem mit dem Zusatz der Sensordatensimulation für den Kamera-Sensor, also die ergänzenden grafischen Informationen.

Hierbei wird besonderer Wert auf eine fotorealistische und damit für das SuT absolut echt wirkende Darstellung gelegt. Grundlegende Funktionstests prüfen, ob eine weiße Linie auf schwarzem Untergrund vom SuT tatsächlich als Fahrspurmarkierung erkannt wird. Möchte man jedoch die Testtiefe erhöhen, so ist dem Steuergerät eine realitätsgetreue Nachbildung der Wirklichkeit zu liefern. Nur damit ist das Verhalten in Grenzregionen der zu testenden Funktion überprüfbar. Beispielsweise muss ein Software-Algorithmus bei blendender Sonne wie auch bei Dämmerung und Regen zuverlässig funktionieren. Die Realität kennt keine Standard-Laborbedingungen, daher muss sich auch in schwierigen Situationen, wie sie im Serieneinsatz der Funktion ständig vorkommen werden, eine Fehlfunktion absolut ausschließen lassen.

Um den hohen Grad des Fotorealismus zu gewährleisten wird eine State-of-the-Art Graphic Engine aus dem „Serious Games“-Bereich eingesetzt. Ähnliche Technologien werden auch in Virtual-Reality und in Spielssystemen verwendet.

Randbedingungen dabei sind die Notwendigkeit der Echtzeit<sup>2</sup>-Darstellung sowie das Preisbewusstsein der Anwender. Wichtiges Element der Darstellung ist also ein Wetter- und Partikelsystem, Verschmutzungselemente sowie vielfältige Parameter, um alle Werte anzupassen. Beispiele dafür sind Spurbreiten, Verschmutzungsgrade, Regenintensität, Windstärke, Tages- und Jahreszeit, Verkehrsdichte und Fahrstil der Verkehrsteilnehmer.

Die PROVEtech:VL-Technologie nutzt also kombiniert die Vorteile der bestehenden Testverfahren für FAS (Vielzahl an realen Test-Situationen) und die Vorteile der HiL-Technologie (Echtzeit-Feedback der Reaktionen, Reproduzierbarkeit, Variabilität, 24/7-Betrieb) unter Laborbedingungen. Die einfache Erstellung neuer Testfälle über eine grafische Benutzeroberfläche sowie die Nutzung der Sensordatenfusion liefern zusätzliche Mehrwerte.

## 2.2 Parametrierbare Szenarien

Nach dem Testsystem nun zu den Testprozessen: Die Vorgehensweisen innerhalb des Testprozesses hat MBtech mit dem Testprozessmodell PROVEtech:TP5 [Bä08] definiert. Die fünf Säulen des Modells beinhalten die Phasen der Teststrategie, Testplanung, Testspezifikation, Testimplementierung sowie Testdurchführung und –auswertung. Das gewünschte Ziel der Teststrategie hat direkten Einfluss auf die Auswahl und Zusammenstellung der Testfälle. Ein mögliches Ziel kann dabei z.B. die Absicherung der Funktionalität und Robustheit nach ISO/IEC 9126 sein. Einflussfaktoren wie begrenzte Ressourcen können dabei Nebenbedingungen darstellen.

Aus vorhandenen Testfall-Katalogen, durch die Analyse von Studien<sup>3</sup> und durch den Einsatz von Expertenwissen wird eine Datenbank mit Situationen (oder „Use Cases“) gefüllt, die für Tests zur Verfügung stehen.

Die Auswahl relevanter Situationen daraus ist abhängig vom Funktionsumfang des SuT sowie von dem gewünschten Teststrategie-Ziel. So werden einzelne Situationen, z.B. Use Case „Tunnelfahrt“, zum Test verschiedener Assistenzfunktionen verwendet. All diese Situationen sind mit möglichen Parametern versehen, z.B. Tageszeit, Wetter, Spurbreite und erlaubte Geschwindigkeit. Die Verknüpfung eines Use Case mit einem bestimmten Parametersatz und einem Test-Ziel wird als Testfall bezeichnet.

Eine komplette Durchführung aller möglichen Testfälle würde zwar zu einer extrem guten Testabdeckung sowie Testtiefe und –breite führen, ist jedoch weder zeitlich machbar noch in aller Konsequenz sinnvoll. Doch selbst wenn man offensichtliche Zusammenhänge - z.B. Sonnenstand und Farbe des Himmels oder Tageszeit und Verkehrsdichte oder Jahreszeit und Straßenverschmutzung durch Laub - kombiniert und nicht jeden möglichen Parameterwert betrachtet (theoretisch ist jeder Testfall für

---

<sup>2</sup> Echtzeit bedeutet hier, dass bei jedem neu aufgenommenen Frame der Kamera die vorhergehenden Reaktionen des Steuergeräts bereits in das Umgebungsmodell und damit in die aktuelle Darstellung der Fahrzeugumgebung eingeflossen sein müssen. Aktuelle Kameras verwenden meist 25 oder 30 fps.

<sup>3</sup> z.B. Verkehrs-, Fahrzeugnutzungs- oder Unfallstatistiken

sekundliche Variationen der Tageszeit über das gesamte Jahr hinweg durchführbar), so verbleiben dennoch extreme Testfallmengen.

Ein weiterer Aspekt ist der fortlaufende Zusammenhang von Use Cases. Der Use Case „Tunneleinfahrt“ ist nicht getrennt von den Use Cases „Tunnelfahrt“ und „Tunnelausfahrt“ zu betrachten. Nach dem Tunnel ließe sich der Use Case „Blendung durch tiefstehende Sonne“ kombiniert mit „Kind läuft auf die Straße“ testen. Schließlich können auch einfache Fälle wie eine Geradeausfahrt ohne Störung auf unterschiedlichsten Straßentypen getestet werden. Die Möglichkeiten der Kombinatorik sind hier schier grenzenlos und nicht in endlicher Zeit durchführbar. Gerade die Fahrten in Echtzeit über eine Vielzahl ähnlicher Streckenabschnitte benötigen viel Zeit.

Es ist also sinnvoll, neben einer Äquivalenzklassenbildung von Parametern die Auswahl von Testfällen noch weiter zu optimieren. Als Ansatz wird hier gewählt, dass man ein Szenario (also die später zu visualisierende Strecke samt Umgebung und Ereignissen) aus einzelnen Testfällen aufbaut. Damit sind bereits verschiedene Use Cases innerhalb kurzer Streckenabschnitte kombinierbar.

Durch die optimierte Generierung mehrerer Szenarien erfolgt eine Abfolge an parametrisierten Testläufen, die die Anforderung an die Teststrategie unter Beachtung der Nebenbedingungen (Ressourcenbeschränkung!) sowie semantischer Zusammenhänge erfüllt. Nach Ansätzen des Design of Experiment (DoE) [KI07] werden aus dieser mehrdimensionalen Testfall- und Parametermatrix die relevanten Testfälle ausgewählt. Nachdem im Vorfeld die Einflüsse von Parameteränderungen und -kombinationen auf die Reaktion des SuT nicht vollständig bekannt sind, ist das DoE sehr gewissenhaft durchzuführen, um zwar die Anzahl der Tests zu reduzieren, jedoch dabei die Testtiefe und -breite nicht einzuschränken.

### **3 Zusammenfassung und Ausblick**

Durch die geschickte Generierung von Test-Szenarien werden mit Hilfe der PROVEtech:VL-Technologie automatisiert reproduzierbare und einfach zu variierende Tests der Software von Fahrerassistenzsystemen mit besonders hoher Testtiefe und -breite durchgeführt.

Das Testen von Steuergeräten auf gewünschte Qualitätsziele hin wird damit in der Fahrzeugdomäne der aktiven Sicherheitssysteme auf ein bisher nicht gekanntes, hohes Niveau gebracht. Steuergeräte mit fusionierten Sensordaten sowie mit hohen Sicherheitsanforderungen lassen sich nun bereits im Labor zusätzlich zu weiteren Testverfahren absichern.

Der nächste Schritt in der PROVEtech:VL-Entwicklung sieht vor, das Zusammenspiel zwischen dem Algorithmus zur optimierten Generierung von Testszenarien und den semantischen Zusammenhängen innerhalb der Datenbank zu automatisieren.



## Literaturverzeichnis

- [Bä08] Bäro, T.: Analyse und Bewertung des Testprozesses von Automobilsteuergeräten. Shaker Verlag, Aachen, 2008.
- [Ch09] Charette, R.: This Car Runs on Code. IEEE Spectrum Online, <http://www.spectrum.ieee.org>, Feb. 2009.
- [Gr05] Grimm, K.: Software-Technologie im Automobil. In (Liggesmeyer, P.; Rombach, D., Hrsg.): Software Engineering eingebetteter Systeme. Spektrum Akademischer Verlag, Heidelberg, Berlin, 2005.
- [KI07] Klein, B.: Versuchsplanung - DoE: Einführung in die Taguchi/Shainin-Methodik. Oldenbourg Wissenschaftsverlag, München, 2007.
- [Li02] Liggesmeyer, P.: Software-Qualität. Spektrum Akademischer Verlag, Heidelberg, Berlin, 2002.
- [MB09] MB-technology GmbH: PROVEtech:TA - Das umfassende Werkzeug zur Testautomation. <http://www.mbtech-group.com>, 26.04.2009.
- [Mü07] Müller, Ch.: Durchgängige Verwendung von automatisierten Steuergeräte-Verbundtests in der Fahrzeugentwicklung. Shaker, Aachen, 2007.
- [Sa08] Sax, E.: Bedeutung des Testens in der Automobilindustrie. In (Sax, E., Hrsg.): Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie. Hanser Verlag, München, 2008.
- [VI09] VIRES Simulationstechnologie GmbH: OpenDRIVE – managing the road ahead. <http://www.opendrive.org>, 26.04.2009.
- [ZS08] Zlocki, A.; Schröder, U.: Test und Bewertung von Bildverarbeitungssystemen. 3. Optische Technologien in der Fahrzeugtechnik, Leonberg, 03.-04.06.2008