

# **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

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.

In einem ersten Schritt wird auf den hohen Stellenwert von Software im Automobil und besonders den aufgrund der sicherheitskritischen Funktionen hohen Stellenwert der Softwarequalität eingegangen. Aktive kamerabasierte Fahrerassistenzsysteme (FAS) werden vorgestellt und bisherige Testmethoden, u.a. Testfahrten und einfache Videoeinspeisungen, beschrieben. Ziel der Absicherungstechnologie für FAS ist jedoch, wie bei jedem anderen Steuergerät, der Hardware-in-the-Loop-Test.

Daher wird im Folgenden eine Technologie zum "in the Loop"-Testen von FAS vorgestellt, d.h. es gibt eine Rückkopplung zwischen der Funktionsreaktion und der Sensorstimulation in Echtzeit. Dies ist für funktionale Tests bei aktiven, d.h. (teil-) autonom agierenden Funktionen unumgänglich. Schwerpunkt der Technologie ist eine möglichst realistische Darstellung der Fahrzeugumgebung, um beliebige Funktionen gerade an ihren Grenzbereichen in aller Tiefe prüfen zu können.

Abschließend wird das Verfahren vorgestellt, mit dem die beschriebene Technologie eingesetzt wird. Hier werden aus Use Cases parametrierbare Test-Szenarien generiert. Aufgrund der Fülle an Parametern und Variationsmöglichkeiten ist vollständiges Testen nicht möglich. Ein Algorithmus erzeugt aus einer Testfall-Datenbank nach Ansätzen des Design of Experiment (DoE) und je nach Randbedingungen die relevantesten Testfälle in einer sinnvollen Reihenfolge. Dies erlaubt, je nach Test-Strategie und zur Verfügung stehenden Ressourcen eine automatische Generierung der nötigen Tests zur Funktionsabsicherung.