

# Besonderheiten beim Test automobiler Steuerungs- und Regelungssysteme

Dr. Eckard Bringmann, eckard.bringmann@piketec.com

PikeTec GmbH, Waldenserstr. 2-4, 10551 Berlin

## Zusammenfassung

Es werden die charakteristischen Eigenschaften der Softwareentwicklung im Automobilbereich im Hinblick auf den Test und die Auswirkungen der Einführung modellbasierter Entwicklungsmethoden besprochen. Außerdem wird das Testwerkzeug "TPT" vorgestellt, das speziell für den Test automobiler Steuerungs- und Regelungssysteme entwickelt wurde.

## Keywords

Modellbasiertes Testen, Automobile Steuerungs- und Regelungssysteme, Testautomatisierung, Test des kontinuierlichen Verhaltens, Closed-loop Tests

## Motivation

In den letzten 15 Jahren hat sich die Entwicklung automobiler Steuergeräte von einer reinen Elektrotechnik- und Maschinenbau-Disziplin zu einer Kombination von Software- und Systemengineering gewandelt. Die Auswirkungen dieser Änderung auf Entwicklungsprozesse, Methoden und Werkzeuge waren sehr weitreichend und sind auch heute noch nicht abgeschlossen.

Gegenwärtig gibt es eine neue Tendenz in der Automobilindustrie hin zur modellbasierten Entwicklung. Software wird nicht mehr ausschließlich in klassischen Programmiersprachen wie C oder Assembler geschrieben, sondern mit MATLAB/Simulink, Statemate, Lab-View oder ähnlichen Werkzeugen modelliert. Diese Techniken bieten für die spezifische Domäne der automobilen Steuerungs- und Regelungssysteme viele Vorteile, die ihre wachsende Popularität erklären. Die Modellierungstechniken haben eine implizite Zeitsemantik und bieten Mechanismen für den Umgang mit kontinuierlichen Signalen und Signalmanipulationen; die modellbasierte Entwicklung ermöglicht Simulationen des Systems in sehr frühen Entwicklungsphasen; Fehler in der Spezifikation oder dem Design lassen sich damit frühzeitig und kostengünstig finden.

## Testen automobiler Software

Mit der wachsenden Komplexität und der Einführung von modellbasierten Entwicklungsprozessen haben auch die Anforderungen an die Qualitätssicherung zugenommen.

Vor ungefähr 15 Jahren umfasste der Test eingebetteter Steuergeräte im Wesentlichen vier Bereiche: (1) elektromagnetische Verträglichkeit (EMV-Tests) (2) elektrische Tests (Kurzschluss, Kriechstrom, Stromspitzen), (3) Klimatests und (4) Feldtests (auf Teststrecken oder auf der Straße). Aufgrund der noch geringen Komplexität gab es keinen Bedarf an speziellen Testmethoden. Im Zuge der Einführung modellbasierter Entwicklungstechniken ist der Test lange Zeit vernachlässigt worden. Da mit der modellbasierten Entwicklung Fehler bereits frühzeitig in den ausführbaren Modellen gefunden werden können, wurde angenommen, dass der Test der Modelle und der Tests des generierten Codes weniger wichtig ist und deshalb keine neuen Testtechniken erfordert. Dieses Ungleichgewicht zwischen der Veränderung bei Software-Konstruktionsprozessen und Testprozessen führte in den vergangenen Jahren zu erheblichen Qualitätsproblemen bei Neufahrzeugen einiger Automobilhersteller, die durch Softwarefehler verursacht waren und erhebliche Garantie- und Kulanzkosten nach sich zogen.

Mittlerweile ist die Notwendigkeit eines gründlichen Tests trotz modellbasierter Entwicklungsprozesse allgemein akzeptiert. Leider gibt es gegenwärtig kaum Testmethoden, die die spezifischen Anforderungen an den Test von Softwareentwicklungen im Automobilbereich abdecken. Klassische Testmethoden und Testwerkzeuge eignen sich hier nur zum Teil. Daher entwickeln viele Serienprojekte proprietäre Eigenlösungen mit erheblichem Zeit- und Budgetaufwand und einem geringen Wiederverwendungsgrad.

## Anforderungen an den Test automobiler Software

Automobile eingebettete Steuerungs- und Regelungssysteme interagieren mit einer realen physikalischen Umgebung und bestehen aus Software, Hardware, Aktuatoren und Sensoren (elektrisch oder mechanisch). Die Entwicklung eines solchen Systems erfordert das Co-Design der Software und Hardware-Bestandteile und damit einen iterativen Entwicklungsprozess mit vielen Zwischenreleases. Das bedeutet für den Test, dass dieselben Tests im Entwicklungszyklus ständig wiederholt werden müssen. Um den Aufwand für diese Testumfänge zu minimieren, ist eine **Automatisierung der Tests** erforderlich.

Die Testautomatisierung kann außerdem die Koordination zwischen Automobilherstellern und Zulieferern während der Entwicklung vereinfachen, wenn jedes Steuergeräterelease, das vom Zulieferer erstellt wird, eine Menge von Abnahmetestfällen bestehen muss, bevor es in ein Erprobungsfahrzeug integriert wird.

Während der Entwicklung wird ein eingebettetes Sys-

tem in mehreren Stufen integriert. Diese Stufen werden wie folgt klassifiziert:

- *Model-in-the-Loop* (MiL): System und Umgebung liegen als Modell vor,
- *Software-in-the-Loop* (SiL): System liegt als Code vor und interagiert mit einer modellierten Umgebung,
- *Processor-in-the-Loop* (PiL): System liegt als Code vor und läuft im Gegensatz zu SiL auf dem Zielprozessor,
- *Hardware-in-the-Loop* (HiL): System liegt als fertiges Steuergerät vor und kommuniziert mit einer simulierten Umgebung in Echtzeit,
- *Erprobungsträger*: System ist ins Fahrzeug integriert

Da sich die Systemfunktionalität zwischen den Integrationsstufen nicht verändert, sollten sich auch die Testfälle nicht ändern. Testfälle sollten möglichst nur einmal entwickelt werden müssen und dann so **portabel** sein, so dass sie für alle Integrationsstufen verwendet werden können. Obwohl diese Anforderung trivial erscheint, ist sie ein Schwachpunkt der gegenwärtigen Testpraxis. Existierende Testverfahren sind i.Allg. für eine bestimmte Testplattform spezialisiert und unterstützen daher keine Portabilität.

Beim Test modellbasierter Entwicklungen hängen Testfälle häufig vom Systemverhalten ab. D.h. der genaue Verlauf eines Testfalls hängt davon ab, wie sich das getestete System verhält. In diesem Sinn müssen System und Testtreiber als **closed-loop** laufen, was viele Testtechniken nicht unterstützen. Ein Beispiel für Closed-Loop-Tests: Ein Testfall für eine Motorsteuerung soll einen Sensorausfall simulieren, sobald die Motordrehzahl einen bestimmten kritischen Wert überschreitet. Der Zeitpunkt dieser Überschreitung ist nicht fix, sondern hängt von verschiedenen Faktoren und Parametern des Systems ab, so dass das System während des Tests beobachtet werden muss, um das Überschreiten zu erkennen.

Testtechniken müssen **echtzeitfähig** sein und Reaktionszeiten im  $\mu$ -Sekundenbereich unterstützen, um Tests mit reproduzierbaren Ergebnissen auch im HiL-Bereich zu garantieren. Ohne die Echtzeitfähigkeit sind manche Tests im HiL nicht durchführbar oder haben so große zeitliche Abweichungen, dass die Ergebnisse unbrauchbar sind.

Da automobile Steuerungen und Regelungen kontinuierliche Signale lesen und schreiben, müssen Testtechniken ebenfalls **kontinuierliche Signale** unterstützen. Die meisten Testtechniken sind hierfür nicht geeignet, sondern können nur diskrete Werteänderungen beobachten bzw. Signale diskret verändern.

Neben den genannten Aspekten ist auch die **Lesbarkeit von Test** als Schnittstelle zwischen Testern, Programmieren und Systemingenieuren sowie zwischen Automobilherstellern und Zulieferern besonders wichtig.

Aufgrund der Systemkomplexität ist ein **systematisches Vorgehen** bei der Testfallauswahl vonnöten, um Redundanzen und eine schlechte Testabdeckung zu vermeiden.

### Automobiler Test mit TPT

TPT steht für *Time Partition Testing* und ist ein Testwerkzeug, das speziell für den Test automobiler Steuerungs- und Regelungssysteme entwickelt wurde. Es basiert auf einer grafischen Testmodellierungssprache, mit deren Hilfe ausführbare echtzeitfähige Tests auf übersichtliche und strukturierte Weise erstellt werden können, die auch die Modellierung kontinuierlicher Signale unterstützen.

Für die Testdurchführung wird ein TPT-Testfall in plattformunabhängigen Bytecode übersetzt, der von einer speziell für TPT entwickelten virtuellen Maschine (TPT-VM) in Echtzeit ausgeführt werden kann. Die TPT-VM ist hochoptimiert und sehr kompakt in Ansi-C geschrieben und daher auf unterschiedlichste Plattformen einfach portierbar. Für die Kommunikation mit einer Integrationsplattform (MiL, SiL, PiL, HiL) muss lediglich ein Plattform-Adapter entwickelt werden, der Signale zur Testlaufzeit zwischen der Plattform und der TPT-VM austauscht. Solche Adapter stehen für eine Vielzahl von Systemen wie z.B. Matlab/Simulink, Targetlink, Tessa, PROVEtech:TA, CTB-HiL, C-Code-Module, LabView und einige proprietäre SiL-Umgebungen bereits zur Verfügung.

TPT bietet ein spezielles Konzept zur systematischen Testfallauswahl. Testfälle werden bei TPT nicht einzeln modelliert, sondern *alle* Testfälle gehören zu *einem* Testmodell. Ziel ist es, Gemeinsamkeiten und Unterschiede der Testfälle klar zu differenzieren, da die Unterschiede zwischen den Testfällen die eigentlichen testrelevanten Aspekte sind: Unterscheiden sich zwei Testfälle in einem bestimmten Aspekt, ist dieser Aspekt für den Test offenbar besonders relevant, da bei den beiden Varianten der Testfälle andere Fehlerpotentiale vermutet werden. Die TPT-Testmodellierung unterstützt deshalb die Bildung von solchen Varianten im Testmodell.

### Ausblick

Der Test automobiler Steuerungs- und Regelungssysteme wird in den kommenden Jahren weiter an Bedeutung gewinnen. Besonderheiten des Entwicklungsprozesses in der Automobilindustrie erfordern spezifische Testansätze in diesem Bereich. TPT ist ein Verfahren, das sich auf dieses Segment spezialisiert hat und das bereits heute in einer Vielzahl von Serienprojekten bei einigen Automobilherstellern und Zulieferern im Einsatz ist.