Automatische Steuergerätetests auf Basis von Hardware-in-the-Loop Simulation

Automated ECU-tests based on Hardware-in-the-Loop Simulation

Dipl. Ing. (FH) **J. Zehentbauer**, AUDI AG, Ingolstadt, Dipl.-Ing. **M. Plöger**, Dipl.-Ing. (FH) **U. Louis**, dSPACE GmbH, Paderborn

Zusammenfassung

Die stark anwachsende Anzahl von Steuergeräten im Kraftfahrzeug verdeutlicht zum einen den rasanten technischen Fortschritt und spiegelt zum anderen die gestiegene Erwartungshaltung des Kunden an das Premiumfahrzeugsegment wider.
Immer stärker zeigt sich die Elektronik als bestimmender Wettbewerbsfaktor. Die damit verbundene Zunahme der Steuergeräte und die steigende Variantenvielfalt lassen sich nur noch durch effiziente Entwicklungswerkzeuge beherrschen. Nur mit ihrer Hilfe lassen sich zukünftig die hochgesteckten Ziele bzgl. Time-to-Market, Qualität und Kosten auf Seiten der Steuergeräte- und Fahrzeughersteller erreichen.

Als eines dieser Werkzeuge hat sich die Hardware-in-the-Loop Simulation etabliert.
Basierend auf einem flexiblen Software- und Hardwarekonzept ist es das ideale Tool für die Entwicklung und den Test moderner Steuergeräte oder -netzwerke. Doch erst durch eine Automatisierung dieser Tests lässt sich die Prüfqualität hinsichtlich Prüftiefe und -breite signifikant erhöhen.
In Zukunft bietet die Testautomatisierung enormes Potential durch eine durchgängige Wiederverwendung von Testszenarien über die einzelnen Stufen des Entwicklungs-prozesses hinweg.

Summary

The rapidly increasing number of ECUs in vehicles reflects the immense speed of technical progress and also the customers rising expectations with regard to top-of-the-range vehicles.
Electronics are increasingly becoming a decisive factor in competition. The greater number of ECUs that this involves, and the greater diversity of versions, can be mastered only by means of more efficient development tools. Without these, it will not be possible to

achieve the ambitious objectives that ECU and vehicle manufacturers have set themselves with regard to time-to-market, quality and cost.

Hardware-in-the-loop simulation has established a place for itself as one of these tools.
Based on a flexible software and hardware concept, it is the ideal tool for developing and testing modern ECUs and networks. However, the quality of tests with respect to testing depth and width cannot be significantly improved until they are automated.
Test automation has enormous potential for the future, since it allows the seamless reuse of test scenarios throughout the individual stages of the development process.

1. Ausgangssituation

Innovationen im Kfz werden heute größtenteils von Elektronik- und hier wiederum von Softwareinnovationen getrieben. Dies führte vor allem in den letzten Jahren zu einem enormen Anstieg in der Zahl elektronischer Steuergeräte im Fahrzeug. In den Fahrzeugen heutiger Generation kommen so 40 und mehr Steuergeräte zum Einsatz.
Gleichzeitig stieg aber auch die Codegröße pro Steuergerät. Das hat dazu geführt, dass der Softwareumfang im Automobil insgesamt exponentiell angestiegen ist.
Unter diesen gegebenen Umständen können die hochgesteckten Ziele hinsichtlich Time-to-Market, Qualität und Kosten nur durch moderne Prozesse und Softwarewerkzeuge erreicht werden. Dabei ist die Hardware-in-the-Loop-Simulation (HIL - Test von Steuergeräten mit einer Simulation der Steuergeräteumgebung im geschlossenen Regelkreis, z.B. Fahrzeug- oder Motorsimulation) in den verschiedenen Phasen des Steuergeräteentwicklungszyklus (V-Modell) zu einem zentralen Bestandteil geworden, deren Nutzen hinlänglich bekannt ist.

2. HIL-Simulation: Ein aktueller Stand

Die Einsatzgebiete der HIL-Simulation wurden in den letzten Jahren stark erweitert. Sie kommt heute unter anderem bei Entwicklungen wie Kurbelwellen-Starter-Generator, 42V-Bordnetz, elektromagnetischem Ventiltrieb, Diesel-Common-Rail-Systeme, Fahrdynamik-regelung oder x-by-Wire Technologien zum Einsatz oder ermöglicht diese überhaupt erst. Hieraus erwachsen stets neue technische Anforderungen an die HIL Simulatoren. So müssen z.B. Komponenten für die Bedienung moderner Bussysteme wie LIN oder zeitgesteuerte Bussysteme (TTP, Flexray) in die Simulatoren integriert werden.
Aus den verschiedenen Phasen (Software-Entwicklung und -Integration, Abnahme- und Release, Steuergeräteverbundtest), in denen die HIL-Simulation im Entwicklungszyklus eingesetzt wird, ergeben sich unterschiedliche Anforderungen an die Testsysteme.
Entsprechend diesen unterschiedlichen Anforderungen hat dSPACE drei unterschiedliche Klassen des dSPACE-Simulators entwickelt (Bild 1).

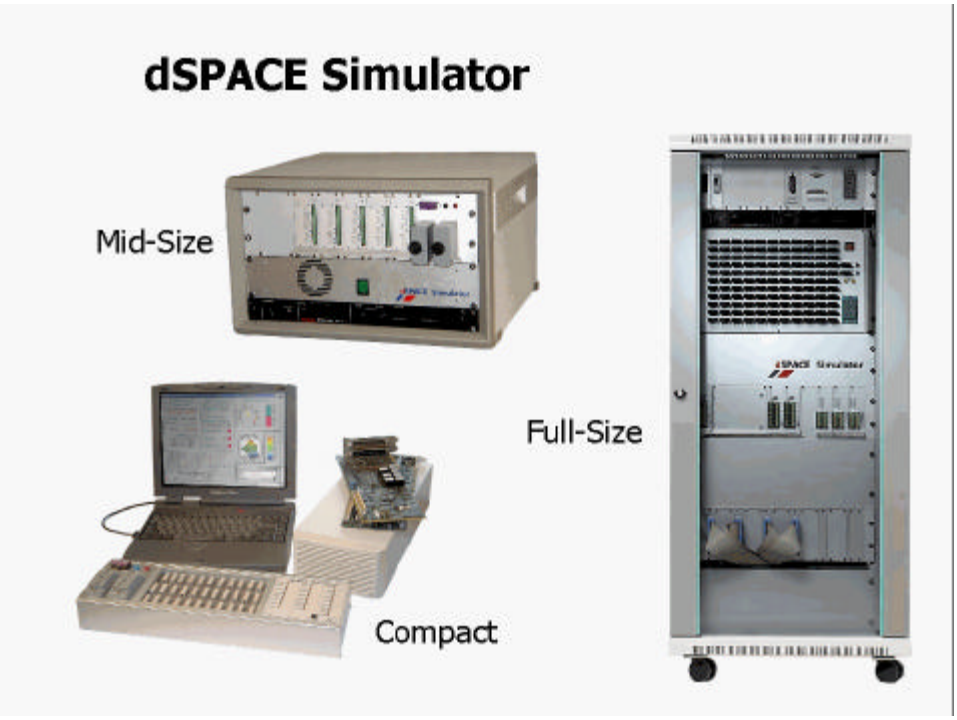

Bild 1: dSPACE Simulator Klassen

Figure 1: dSPACE simulator classes

Als einen entscheidenden Vorteil für den Anwender erweist es sich, dass die technologische Basis aller dSPACE HIL-Simulatoren identisch ist: Die Softwarekomponenten (Bedienoberflächen, Automatisierbarkeit, Simulationsmodelle in MATLAB /Simulink) sowie die wesentlichen Hardwarekomponenten sind identisch. Das spart, bei einem Wechsel zwischen den verschiedenen Simulatorklassen, Zeit und Kosten.

Für den Test untereinander vernetzter Steuergeräte lassen sich dann natürlich auch die virtuellen Fahrzeugkomponenten, wie z.B. Motor, Getriebe und Fahrwerk eng untereinander verkoppeln. Das kann zum Beispiel auch dadurch erreicht werden, dass existierende einzelne Simulatoren untereinander sehr einfach zu einem Multiprozessorsystem zusammengeschlossen werden können. Dieser Übergang vom Einzelsimulator zum Verbund (Bild 2) wird besonders einfach, durch die effiziente und hochwertige Unterstützung mittels grafischer Programmierung in MATLAB/Simulink.

Bild 2: Simulatorverbund zum Steuergerätetest

Figure 2: Simulationnetwork for ECU-testing

3. Anforderungen an ein Tool zur automatischen Prüfung

Der bekannte Nutzen der Hardware-in-the-Loop-Simulation beim Testen von Steuergeräten oder –verbünden kann durch die Automatisierung dieser Tests signifikant gesteigert werden. Dieses gilt zum Beispiel für die Prüfqualität, die erst durch Testautomatisierung (TA) auf das erforderliche Maß angehoben werden kann.
Betrachtet man die Entwicklung der Software-Tools im HIL-Bereich, so wurden schon von jeher Möglichkeiten geboten, um automatische Tests durchführen zu können. Lagen die Schwerpunkte in der Vergangenheit auf Unterstützungsfunktionen wie Makros, Treiber-Libraries oder GUI-Buildern, so ist heute der Trend festzustellen, komplett durchgängige Systeme bereitzustellen.

Im folgenden wird eine durchgängige Lösung vorgestellt, die im Rahmen eines Engineering - Projektes in Zusammenarbeit von dSPACE und AUDI entstand. Mit diesem Projekt sollten in erster Linie zwei Ziele erreicht werden: Eine Erhöhung der Prüfqualität auf der einen Seite und eine wirkungsvolle Entlastung der Nutzer auf der anderen Seite.

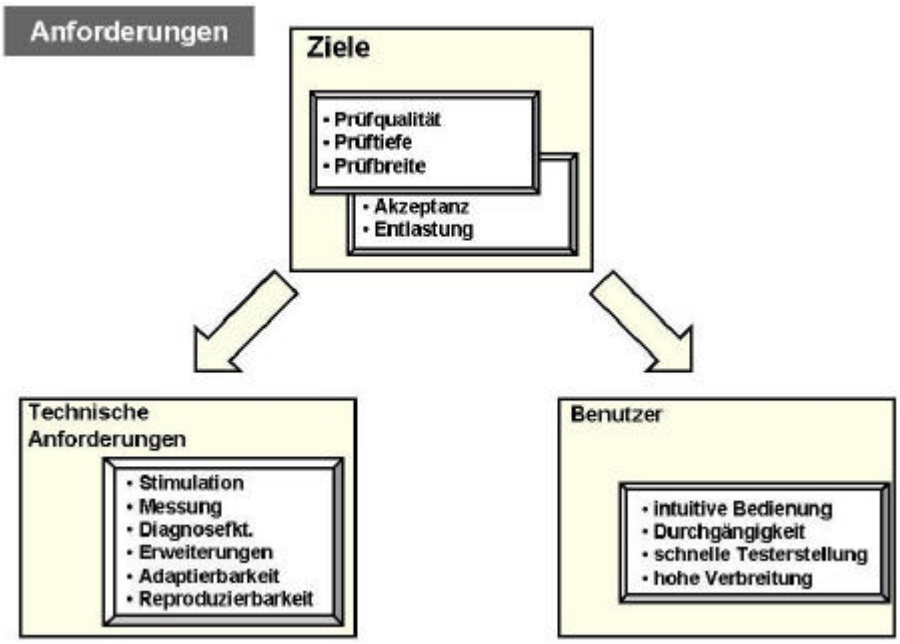

Bild 3: Anforderungen und Ziele

Figure 3: Demands and targets

Neben den sich daraus ergebenden technischen Forderungen ist insbesondere die Schnittstelle zum Nutzer von entscheidender Bedeutung (Bild 3). Deshalb lag der Entwicklungsschwerpunkt darin, ein einfach und intuitiv zu bedienendes Tool bereitzustellen. Dieses ermöglicht es dem Anwender nach kürzester Einarbeitung Prüfabläufe zu definieren, auszuführen und auszuwerten und somit eine ausführbare Prüfspezifikation zu erstellen. Die eigentliche Testgenerierung sollte im wesentlichen ohne jegliche Programmierkenntnisse erfolgen können, wenngleich die Option für die Integration von manuellen nutzerspezifischen Programmteilen vorhanden ist.

Entwicklungsziele:

- Steigerung der Prüfqualität durch höhere Prüftiefe und Prüfbreite
- Entlastung von Routinetests und Akzeptanz bei den Endanwendern

Technische Anforderungen:

- ein durchgängiges Tool für alle wesentlichen Phasen des Testablaufes
- einheitliches Layout für Spezifikation und Dokumentation
- Vorgaben und Ergebnissen im gleichen Dokument (wg. Vergleichbarkeit und Auswertung)
- einfache Auswertung von Zwischenergebnissen / Ergebnissen
- schnelle Übersicht hinsichtlich: Testergebnis i.O. / n.i.O.
- klare Ablagestruktur der Ergebnisdateien
- keine zusätzlichen Softwarevoraussetzungen (wie z.B.: Compiler)
- mögliche Trennung von Testerstellung / Testauswertung und Testausführung
- flexibel und änderungsfreundlich (schnelle Änderung einzelner Rahmenparameter)
- Integrationsfähigkeit von handcodierten Programmteilen

Benutzeranforderungen:

- intuitive Spezifikation von Prüfabläufen (Sicherstellung durch Mechanismen wie: „Drop-Down-Menüs“, „Copy-Paste“ usw.)
- interaktive Testdurchführung (Ergebnisse und Zwischenergebnisse können jederzeit mitverfolgt werden)
- kurze Einarbeitungszeit für den Endanwender (keine Programmierkenntnisse erforderlich)

4. Implementierung der Testautomatisierung

4.1 Schnittstelle zum Benutzer

In der Konzeptphase für die Testautomatisierung wurde zunächst der bisherige Prozess der manuellen Softwareprüfungen analysiert. Dabei zeigte sich, dass einzelne Funktionstests in vielen Fällen ähnlich ablaufen. Dies gilt auch weitestgehend über Steuergerätegrenzen hinweg. Wichtig war dieser Punkt insbesondere deshalb, weil die Testautomatisierung im späteren Verlauf auch auf andere Projekte übertragen werden sollte.

Die Prüfungen selbst folgen meist einem einheitlichen Schema. Nach dem Anfahren eines bestimmten interessierenden Arbeitspunktes werden Eingangsgrößen definiert verändert. Für die zu überwachenden Ausgangsgrößen werden Sollwerte festgelegt, und es findet ein Vergleich von Soll und Ist statt. Daraus wird eine i.O. / n.i.O. – Betrachtung abgeleitet, und die Ergebnisse werden in der Testdokumentation abgelegt.

Charakteristisch ist ein sequentieller Ablauf, der sich sehr anschaulich in eine zeilen- und tabellenförmige Struktur überführen lässt. Dies legte den Gedanken nahe die eigentlichen Einzelprüfabläufe als Prüftabellen darzustellen. Darüber hinaus konnten auch die geforderten Verwaltungsfunktionen für die Zusammenführung mehrerer Einzeltests in Tabellen realisiert werden.

Daneben sollte die Benutzerschnittstelle bei der Auswertung unterstützen, die Möglichkeit für einfache Änderungen bieten und intuitiv bedienbar sein. Aus diesen Gründen kristallisierte sich Excel als zentrales Userinterface heraus. Mit zu dieser Entscheidung trug der hohe Bekanntheitsgrad, die weite Verbreitung und die in Excel enthaltene Programmiersprache VBA (Visual Basic for Applications) bei.

Der Anwender kann sich den Prüfablauf in Form eines Excel-Blattes beliebig zusammenstellen. Die Abarbeitung erfolgt Zelle für Zelle von links nach rechts, bzw. zeilenweise von oben nach unten. Für die Auswertung stehen die Standardfunktionen aus der Tabellenkalkulation zur Verfügung.

Die Funktion einer Zelle wird über die Definition der Spalte im Spaltenkopf festgelegt. Dabei können die am häufigsten benötigten Funktionen über „Drop-Down“ Menüs ausgewählt werden. Für Sonderfunktionen wurde zudem die Option offen gehalten, eigene Programmteile einzufügen. Dabei ist lediglich der Pfad anzugeben; das entsprechende Codestück wird dann automatisch eingebunden.

Für die Überprüfung von Diagnosefunktionen, wurden eigens Spaltendefinitionen vorgesehen. Somit können Funktionen wie Fehlerspeicher lesen, löschen, Messwerteblöcke lesen usw. ausgewählt werden. Eine zusätzliche Excel-Toolbar ermöglicht dem Nutzer die Bedienung der TA-spezifischen Funktionen, wie das Erzeugen von ausführbaren Scripten und das Starten des Prüfablaufs. Die eigentliche Testgenerierung ist dabei nicht komplizierter als die Erstellung eines Diagramms und kommt ohne jegliche Programmierkenntnisse aus. Dies führte zu einer raschen Durchdringung und einer hohen Akzeptanz bei den Anwendern.
Des weiteren konnten auch die geforderten Verwaltungsfunktionen in Excel abgebildet werden. Über der Schicht aus beliebig vielen, einzelnen Prüftabellen liegt eine weitere Tabelle, die Sequenztabelle. Diese hat die Funktion eine ganze Sequenz von nacheinander abzuarbeitenden Einzelprüfungen zu verwalten. Darüber hinaus erfolgt auch die Aktivierung von einzelnen Tests und der Start einer Sequenz von hier aus.

4.2 Technisch Umsetzung

Aus den in Excel beschriebenen Testabläufen werden die Tabellendaten mittels Visual Basic an einen in Python realisierten Codegenerator übergeben. Dieser generiert ausführbare Scripte in Python. Bei Python handelt es sich um eine objektorientierte Interpretersprache. Excel fungiert somit zusammen mit VBA als User-Interface und Übergabeschnittstelle. Die erzeugten Scripte sind handgeschriebenem Code gleichwertig und können auch manuell erweitert, oder verändert werden.
Die Echtzeithardware ist über eine Pythonschnittstelle angebunden. Über eine entsprechende Library können Variablen der Echtzeithardware manipuliert, gelesen und visualisiert werden.
Die Messwerte werden zusammen mit einem Ablaufprotokoll in eine Dateikopie der Excel-Prüftabellen zurück geschrieben, so dass das Erscheinungsbild von Testspezifikation und Testdokumentation einheitlich ist. Um zeitliche Abhängigkeiten und Bezüge realisieren zu können, wurde auf der Echtzeithardware eine unabhängige Uhr implementiert. Diese arbeitet völlig getrennt von der PC-Systemzeit und dient zur Synchronisation während des Testablaufes.
Die verschiedenen Steuergeräte im Fahrzeug weisen im VW-Konzern eine standardisierte Diagnoseschnittstelle auf. Über diese kann der Kundendienst auf den Speicher eines Steuergerätes zugreifen. Dadurch können z.B. schnell Fehler diagnostiziert,

Parametrierungen vorgenommen, oder variantenabhängige Codierungen durchgeführt werden. Diese Diagnosefunktionalitäten nehmen einen großen Stellenwert innerhalb der Steuergerätesoftware ein. Deshalb bildet die Überprüfung dieser Eigendiagnosen auch einen Schwerpunkt beim Softwaretest des Steuergerätes.

Die Testautomatisierung macht sich dabei die angesprochene Schnittstelle der Steuergeräte zum Werkstatttester zunutze, indem sie auf diesem Weg auf das Steuergerät zugreift. Dazu wurde von dSPACE eine Python-Library entwickelt, welche die Kommunikation mit dem Steuergerät ermöglicht (Bild 4).

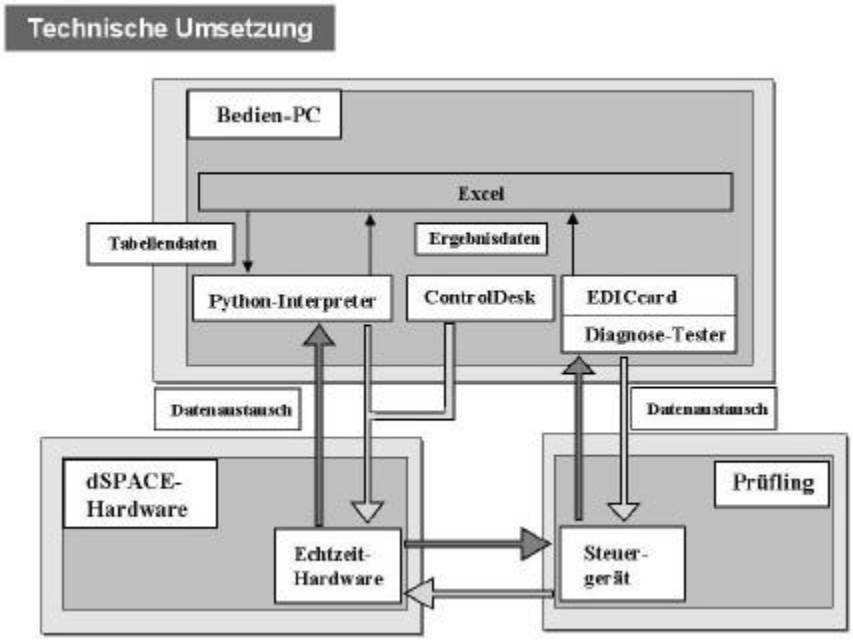

Bild 4: Umsetzung

Figure 4: Implementation

Bei der Erzeugung des Codes für die Echtzeithardware wird eine Mapping-Datei angelegt. Diese bildet die Schnittstelle für übergeordnete Softwareschichten wie etwa Control Desk (zentrale grafische Bedienoberfläche der Firma dSPACE). Auch die Testautomatisierung bedient sich dieser Schnittstelle. Der Nutzer kann so über einen gewöhnlichen „Drag&Drop-Mechanismus“ eine Verbindung zwischen Parametern der Prüftabellen und der Echtzeithardware herstellen.

5. Betrachtung des Nutzens

5.1 Vorteile von HIL / Automatisierung von Testabläufen

Da die Testautomatisierung auf die Technik der Hardware-in-the-Loop Simulation aufsetzt, gelten prinzipiell auch deren Vorteile. Diese seien der Vollständigkeit halber erwähnt, nähere Angaben finden sich unter /1/. So bietet die HIL-Technik die Möglichkeit Hardware, im Falle von AUDI in erster Linie elektronische Steuergeräte, außerhalb des Fahrzeugs in einer realistisch nachgebildeten Umgebung zu testen.
Neben der Reduktion von Testfahrzeugen und damit einhergehenden Zeit- und Kostenersparnis können Versuche in Extrembereichen durchgeführt werden. Gerade bei diesem Punkt stößt man in der Praxis ohne HIL auf Grenzen. Schwierig wird dieses Unterfangen deshalb, weil die Steuergeräte umfangreiche Plausibilitätschecks der Eingangssignale durchführen und dies z.T. durch Adaptionseffekte überlagert wird.
Im Gegensatz zum Fahrzeug lassen sich am Simulator eindeutig reproduzierbar, definierte Arbeitspunkte anfahren, um dann einzelne Funktionen detailliert zu testen.

5.2 Auswirkungen auf die Prüfabdeckung

Im heutigen Softwareentwicklungsprozess setzen viele Hersteller auf eine modulare Softwareplattform. Ein Grund hierfür liegt in der besseren Wiederverwendbarkeit von Softwarekomponenten, wodurch sich Vorteile hinsichtlich Qualität, Entwicklungszeit und –kosten ergeben. Durch eine Standardisierung von Softwarebausteinen wie Treiber-software und Betriebssysteme lassen sich in zukünftigen Projekten große Teile der Soft-ware wiederverwenden */2/*.
Doch gerade im Zusammenspiel mit neu entwickelten Softwarekomponenten können auch neue Fehler in bereits getesteten Softwareteilen auftreten. Dies schafft die Notwendigkeit die Funktion, auch von bereits im Einsatz befindlichen Codeteilen, stets neu zu verifizieren. Insbesondere für dieses Einsatzgebiet sind automatische Prüfroutinen prädestiniert.

Nur so können mit geringem Aufwand Tests dieser Softwarekomponenten erfolgen. Im Laufe der Zeit entsteht so mittels der Testautomatisierung eine Bibliothek an Standardtests für Standardsoftwareteile. Der Entwickler wird so von Routinetests entlastet und kann sich ganz auf die Überprüfung von neuen Funktionen konzentrieren.

Erst dadurch kann eine hohe Prüfbreite gewährleistet werden. Eine Erhöhung der Prüftiefe bedingt geradezu den Einsatz von Testautomatisierung auf modellbasierten HIL-Systemen. Nur automatisch ablaufende Tests am HIL-Simulator machen gezielte Parametervariationen oder Zufallsanalysen wie Monte-Carlo erst möglich, ohne hohe Personalkapazitäten zu binden

5.3 Anwendungsbeispiele

Im Rahmen des hier vorgestellten Pilotprojektes war der Einsatz als erstes beim Schalttafeleinsatz (Kombiinstrument) vorgesehen. Hierbei werden alle elektrischen Schnittstellen (wie CAN, digitale I/O, analoge I/O) und darüber hinaus interne Größen wie Diagnose oder Speicher (RAM, ROM) betrachtet. Bislang sind mehr als 150 Einzeltests für verschiedenste Teilfunktionen, die von der Eigendiagnose über die Prüfung von Bordcomputerberechnungen bis hin zur Wartungsintervall-Verlängerung reichen, realisiert worden. Alle aneinandergereiht ergeben diese eine Prüfdauer von über 200 h und füllen somit eine komplette Arbeitswoche.

Nach den ersten positiven Erfahrungen kam die Testautomatisierung auch in anderen Projekten zum Einsatz. Heute wird auch das Airbagsteuergerät, die Luftfederung, verschiedene Getriebesteuergeräte und die automatische Distanzregelung per Automat getestet (Bild 5).

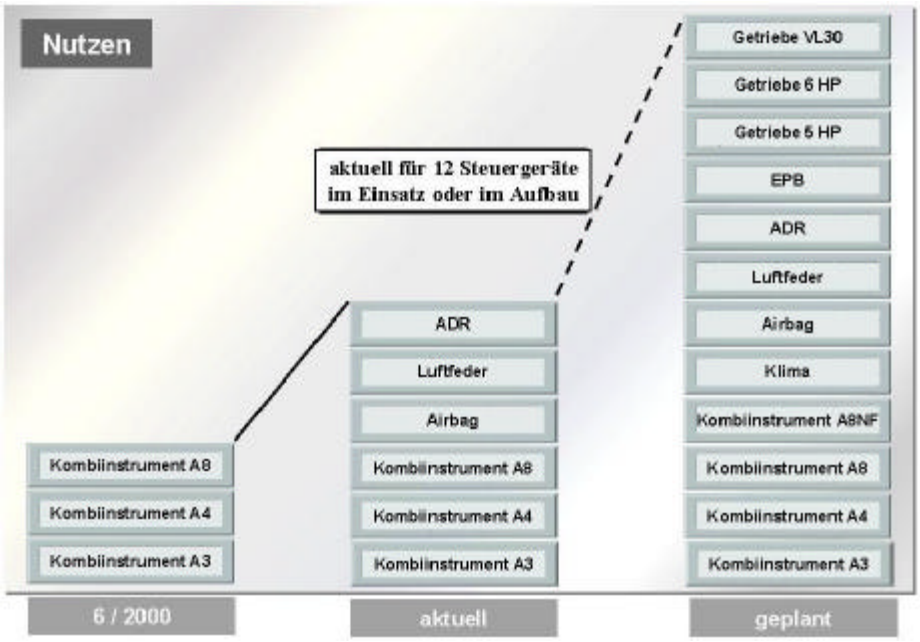

Bild 5: Anwendungen

Figure 5: Applications

Die Schwerpunkte hierbei liegen in den Eigendiagnosen der Steuergeräte und in einzelnen Softwarealgorithmen. Darüber hinaus befindet sich die Testautomatisierung für weitere Steuergeräte gerade im Aufbau.

6 Ausblick

In Zukunft werden die bei HIL zur Beschreibung der Steuergeräteumgebung eingesetzten Tools wie MATLAB/Simulink und Stateflow auch in frühen Phasen des Entwicklungsprozesses Verbreitung finden. Statt der heute üblichen grafischen Ablaufdiagramme oder in Textform abgefassten Beschreibungen werden „ausführbare Lastenhefte“ für Steuergeräte entstehen.

Mit diesen wird es möglich sein die Spezifikationen Software-in-the-Loop, also in einer Strecke und Regler umfassenden Simulation, ohne jegliche Hardware zu testen.

Heute im HIL-Bereich verwendete Testtools sind nicht zwingend an die Hardwareplattform gebunden, sondern können auch in der Phase der Softwarespezifikation und Softwareimplementierung für erste Tests eingesetzt werden. So können Testvektoren bereits in

diesem Entwicklungsstadium definiert werden. Im weiteren Entwicklungsverlauf können diese dann über die einzelnen Softwarestände hinweg bis hin zur Seriensoftware wiederverwendet werden.

Im Gegensatz zu heute, werden automatische Prüfungen am Hardware-in-the-Loop-Simulator nicht mehr dazu dienen, in Prosa beschriebene Funktionen zu prüfen.

Viel mehr wird die fehlerfreie Umsetzung von bereits offline getesteten Algorithmen und Reglern im Steuergerät überprüft. Auch das Modell der Regelstrecke (Steuergeräteumgebung) ist so in der HIL-Phase wiederverwendbar (Bild 6).

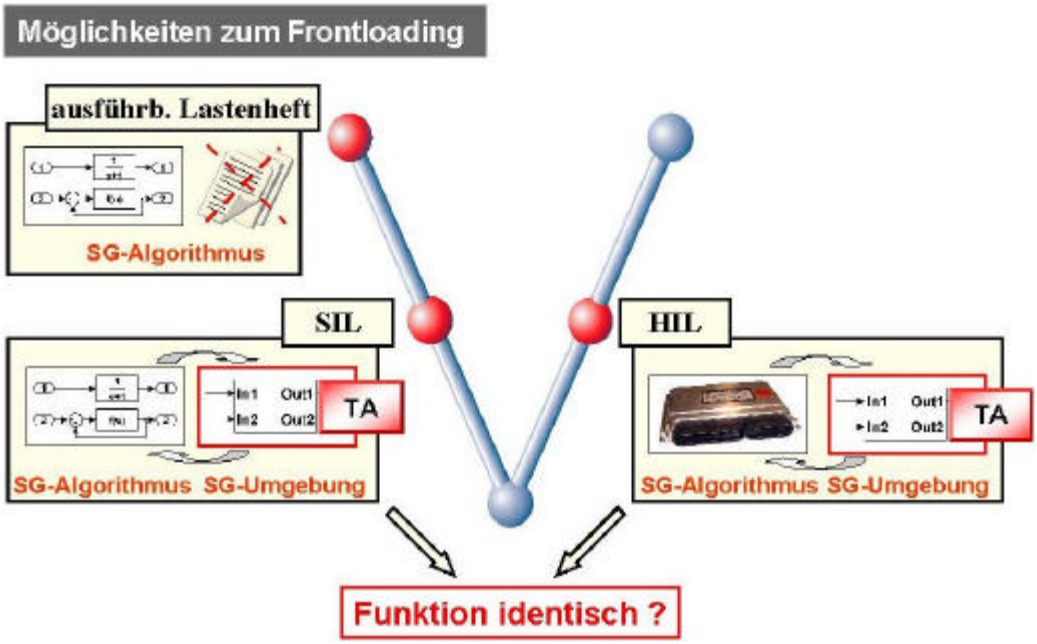

Bild 6: Frontloading durch Einsatz durchgängiger Toolketten

Figure 6: Frontloading using a throughout consistent toolchain

Die Variantenvielfalt an oft herstellerspezifischen Diagnosetools stellt derzeit noch ein Hindernis in der Vereinheitlichung der Test- und Simulationssysteme dar. Eine Bestrebung um hier eine Vereinheitlichung zu erreichen ist die ASAM-Initiative /3/ (Arbeitskreis zur Standardisierung von Automatisierungs- und Meßsystemen). Diese definiert Schnittstellen und Standards, welche die Hersteller von Mess- und Applikationssystemen unabhängig von der konkreten Implementierung der Steuergerätesoftware macht. Somit wäre auch die detaillierte Betrachtung von

Softwaremodulen z.B. an Hand von Übergabe- und Ergebniswerten, die im ASAP2-File abgelegt sind, möglich.
Für die Automatisierung basierend auf dSPACE-Simulatoren wird derzeit eine neue Generation von Tools entwickelt, welche eine "Kondensierung" des umfangreichen TA-Know-Hows aus dem vorgestellten sowie verschiedenen weiteren Engineering Projekten darstellen wird. Dabei garantiert eine enge Zusammenarbeit mit verschiedenen Kunden aus der Automobilindustrie an den Kundenanforderungen orientierte Konzepte. Dieses Tool wird neben HIL die gesamte Toolkette für den Steuergeräteentwicklungszyklus abdecken. Verschiedene höherwertige Benutzergänge werden für die Erfüllung unterschiedlicher Kundenanforderungen sorgen.

7 Literatur

/1/ H. Hanselmann
Hardware-in-the-Loop Simulation: Ziele, Wege, Resultate
Haus der Technik 2/98

/2/ H. Deiss, B. Aumann, T. Schober
Time to Market in der Softwareentwicklung
-ReUse und Standardisierung bei Getriebesteuerungen –
VDI Berichte Nr. 1547, 2000

/3/ Informationen zu ASAM: www.asam.de

/4/ H.Schütte, M.Plöger, K.Diekstall, P. Wältermann, T. Michalsky
Testsysteme im Steuergeräte-Entwicklungsprozess
Automotive Electronics 2001, Sonderausgabe von ATZ und MTZ und Automotive Engineering partners