

Automatisierter Test für Softwaremodule

Methode zur Testfall-Generierung

Die Daimler AG setzt bei der Entwicklung von Softwaremodulen für automatische Fahrzeuggetriebe eine modellbasierte Testumgebung ein. Die Absicherung der Funktionalität der Module erfolgt mit einem neu entwickelten Testgenerator von QTronic, der Tausende von Testfällen automatisch erzeugt, simuliert und analysiert. Die einzelnen Komponenten der Entwicklungsumgebung, wie Modelle für Fahrzeug und Antriebsstrang, Softwaremodule und Testgenerator, sind über ein Software-in-the-Loop(SiL)-Werkzeug gekoppelt und auf handelsüblichen Laptops ablauffähig. Dadurch lassen sich die Softwaremodule wie im realen Fahrbetrieb ansteuern und testen.

1 Einleitung

Die Komplexität von Motor- und Getriebesystemen für Kraftfahrzeuge nimmt ständig zu. Ein Grund hierfür sind die wachsenden Erwartungen des Marktes hinsichtlich Motor- und Getriebeeffizienz, Agilität, Fahrspaß und Emissionen. Eine Vielzahl der Marktanforderungen lässt sich nur über die Kombination von robuster Mechanik und intelligenter Software realisieren. Ein Beispiel für diese Entwicklung ist das MCT-Siebengang-Sportgetriebe, bei dem der hydrodynamische Drehmoment-Wandler durch eine elektronisch geregelte Anfahrkupplung ersetzt wurde [1]. Für die Regelung der Anfahrkupplung mussten neue Softwaremodule entwickelt und in die bestehende Software der 7G-Tronic integriert werden. Zusätzlich zu den bisherigen Testverfahren wie Hardware-in-the-loop (HiL) Tests, Untersuchungen an Prüfständen und Fahrversuchen mit physikalischen Prototypen wurde eine Methode zum Test von Softwaremodulen mittels automatischer Testfall-Generierung eingesetzt. Dadurch wurden Testabdeckung und Testeffizienz erheblich verbessert. Ziel der Testautomatisierung ist ein umfassender Systemtest, bei dem möglichst

viele relevante Systemzustände automatisch erreicht und überprüft werden.

2 Modellbasierte Softwaretestumgebung

Die Testmethode wird für den Systemtest komplexer Softwaremodule eingesetzt, die sich aufgrund der Rückkopplungen zwischen Softwarefunktionen und Dynamik der Mechanik nur im eingebetteten Zustand testen lassen. Grundlage dazu ist eine Softwaretestumgebung, die eine Kopplung von Streckenmodellen, den zu testenden Softwaremodulen und einem Testgenerator ermöglicht. Als Testgenerator wird das Entwicklungswerkzeug TestWeaver von QTronic verwendet. Die Testmethode wird entwicklungsbegleitend eingesetzt, um Schwachstellen und Fehler der Funktionssoftware möglichst früh zu erkennen.

2.1 Arbeitsschritte der Testmethode

Im ersten Arbeitsschritt wird die Testumgebung eingerichtet. Das Streckenmodell wird aufgebaut und der Testautomat wird konfiguriert. Dazu muss zum einen festgelegt werden, welche Variablen der Testfallgenerator steuern darf (zum Beispiel

Die Autoren



Dr.-Ing. Anton Rink ist Leiter der Getriebe-logik bei der Daimler AG in Stuttgart.



Dipl.-Ing. Emmanuel Chrisofakis ist Berechnungsingenieur bei der Daimler AG in Stuttgart.



Dr. Mugur Tatar ist Geschäftsführer der QTronic GmbH in Berlin.

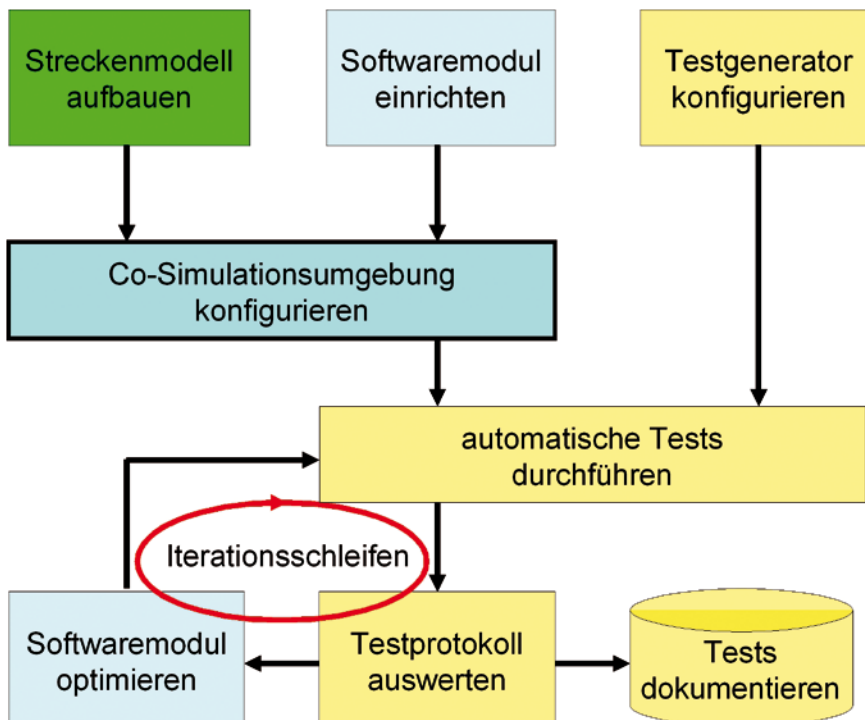


Bild 1: Arbeitsschritte der Testmethode

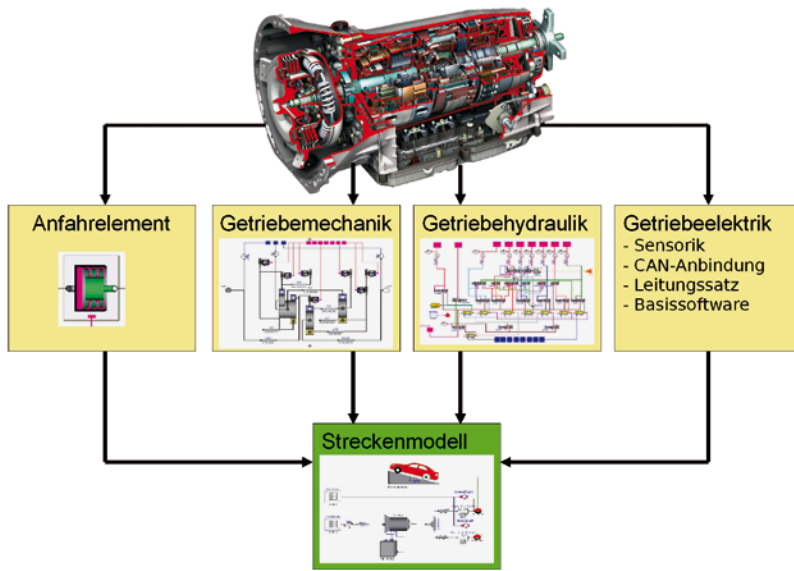


Bild 2: Streckenmodell der Testumgebung

Straßensteigung, Fahrpedal, Bremspedal etc.). Zum anderen müssen Kriterien für die gut/schlecht Bewertung des Systemverhaltens definiert werden. Außerdem sind Berichtsvorlagen für das Protokollieren des Testverlaufs zu erstellen. Der zu testende Softwarestand wird für den Einsatz in der SiL Umgebung vorbereitet. Dazu werden die Softwaremodule in eine auf einem PC ablauffähige Form (als DLL) gebracht. Nach der Konfiguration der Co-Simulationsumgebung wird der Testgenerator gestartet. Das Testwerkzeug generiert tausende von unterschiedlichen Testfällen, führt diese im SiL aus, registriert und bewertet die resultierende Systemantwort und erstellt ein Testprotokoll. Das

Testergebnis wird durch einen Entwicklungsingenieur bewertet. Nach Korrektur der gefundenen Fehler wird der Softwaretest zur Erfolgskontrolle erneut durchlaufen. Die einmal erstellte Konfiguration des Testgenerators kann dabei beliebig oft genutzt werden. Dadurch wird eine iterative Vorgehensweise unterstützt, mit der die Funktionssoftware kontinuierlich verbessert wird. Bild 1 zeigt zusammenfassend die Arbeitsschritte der Testmethode.

2.2 Streckenmodell der Testumgebung

Herzstück der Testumgebung ist das Streckenmodell des Getriebes, Bild 2, dessen Detaillierung festlegt, welche Softwarefunktionen der Getriebesteuerung mit

welcher Güte getestet werden können. Mittels der objektorientierten Modellierungssprache Modelica [4] wird das komplexe dynamische Verhalten des Getriebes, bestehend aus Radsätzen, Wellen, Schaltelementen, Ölversorgung, Hydraulikmodul und Elektronikmodul, beschrieben. Die für die Getriebesteuerung wichtigen Komponenten, wie Anfahrkuppelung, Schaltkupplungen, Bremsen und hydraulische Steuerung, sind detailliert modelliert. Mögliche Bauteilausfälle der Getriebehardware, die sich über eine Risikoanalyse ermitteln lassen, werden ebenfalls in das Streckenmodell integriert und lassen sich über Fehlerzustandsvariablen aktivieren. Weitere Teile des Streckenmodells sind Motor, Triebstrang, Bremssystem, Rad-Straße-Kontakt und Fahrzeugaufbau. Das Streckenmodell wird mit Hilfe von Messungen, die an Komponentenprüfständen und im Fahrzeug aufgezeichnet werden, kalibriert, wobei eine Modellgenauigkeit für das statische und das dynamische Verhalten von mindestens 10 % gefordert wird. Dadurch entsteht ein Modell für die Längsdynamik eines Fahrzeugantriebs, mit dem sich die relevanten Fahrsituationen und Bauteilfehler eines Getriebes simulieren lassen.

2.3 Zu testende Softwaremodule

Die zu testenden Softwaremodule der Getriebesteuerung werden als DLL in die Testumgebung integriert. Je nach Entwicklungsphase können unterschiedliche Beschreibungsformen für die Spezifikation von Struktur, Verhalten, Variab-

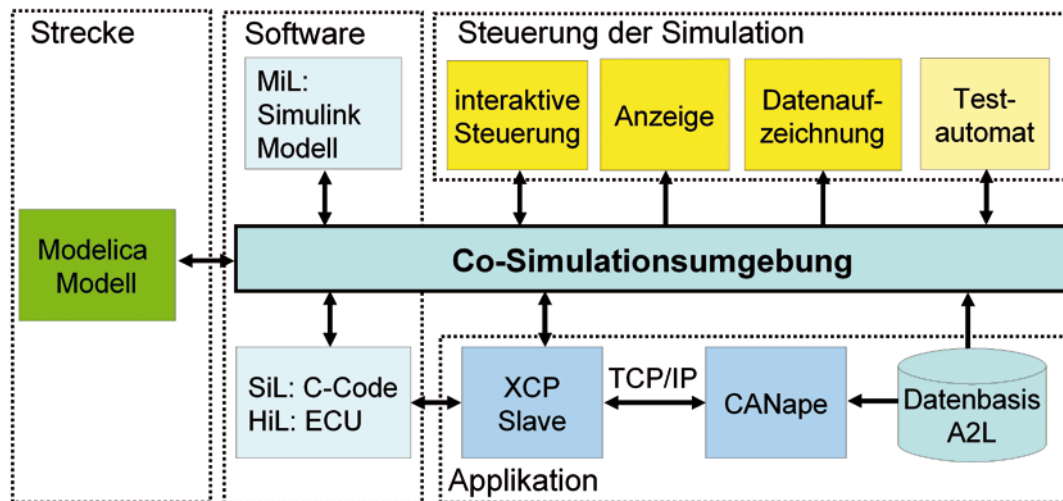


Bild 3: Co-Simulationsumgebung mit interaktiver Steuerung und Anzeige

len und Parameter der Software eingesetzt werden. In dem Anwendungsbeispiel dieses Beitrags wird der SiL-Test betrachtet; das heißt, der originale C-Code der Getriebesteuerung wird ohne Modifikationen in die Testumgebung integriert. Damit die Software mit dem Simulationsmodell kommunizieren kann, werden hardwarenahe Funktionen des Getriebesteuergeräts nachgebildet. Die so erzeugte DLL der Getriebesteuerung wird mit der gleichen Rechenzykluszeit wie im Fahrzeug aufgerufen.

2.4 Simulationsumgebung

Das Streckenmodell und die zu testenden Softwaremodule der Getriebesteuerung werden als DLL in einer SiL-Simulationsumgebung mittels Co-Simulation zyklisch, zum Beispiel im 10-ms-Takt, ausgeführt. In jedem Takt tauschen die Module Signale miteinander aus. Auf diese Weise kann das Zusammenspiel zwischen Software und Fahrzeughardware „virtuell“ auf einem PC ermittelt und geprüft werden. **Bild 3** zeigt die Struktur der gesamten Softwaretestumgebung, bestehend aus Streckenmodul, Softwaremodul, Anbindung an Applikationswerkzeug über XCP, graphische Schnittstelle zur Steuerung der Simulation oder – alternativ – Anbindung zum Testautomaten. Für die Applikation der Softwaremodule werden in der SiL-Umgebung – dank der standardisierten ASAM-Schnittstellen XCP und A2L – dieselben Werkzeuge genutzt, die auch im Fahrzeug zum Messen und zum Applizieren eingesetzt werden. Für die Analyse des C-Codes kann ein Debugger, sowie ein Werkzeug zur Messung der Code-Erreichung integriert werden.

2.5 Testgenerator und automatisierte Testauswertung

Das Entwicklungswerkzeug TestWeaver [2, 3] wird als Testgenerator eingesetzt. Während eines Tests erzeugt der Testfallgenerator selbständig tausende von Testfällen, führt diese per Simulation im SiL aus und bewertet die resultierende Systemantwort, **Bild 4**. Testfälle werden dabei nicht rein zufällig erzeugt, sondern mittels einer intelligenten Steuerung, die zwei Ziele verfolgt. Das erste Ziel ist das Erreichen einer möglichst großen Testabdeckung. Testfälle (Fahrmanöver) werden vom Testfallgenerator systema-



eCarTec

2. Internationale Messe für Elektromobilität

Datum:

19. - 21. Oktober 2010

Ort:

Neue Messe München

Themen:

Elektro-Fahrzeuge
Energie und Infrastruktur
Speichertechnologien
Antriebs- und Motorentechnik
Engineering und Zulieferung
Finanzen
Fahrzeug-Teststrecke



www.ecartec.de

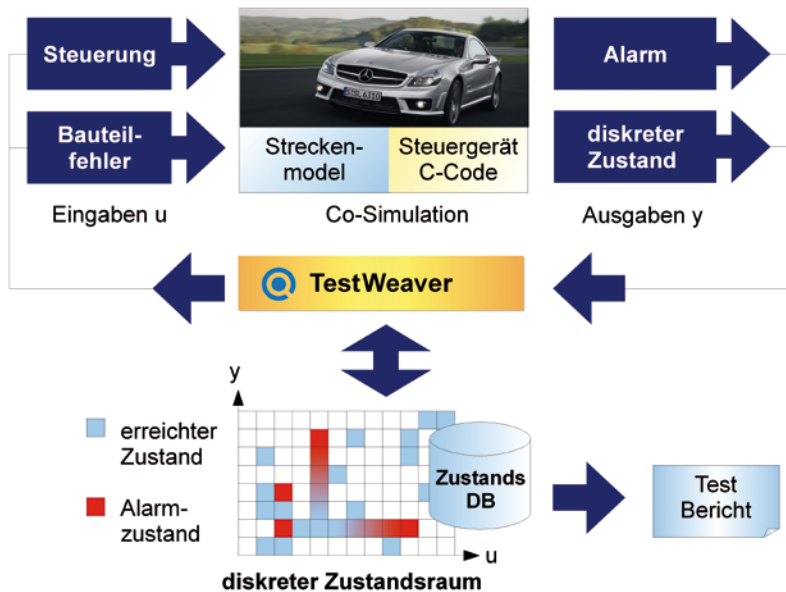


Bild 4: Arbeitsweise des Testfallgenerators

tisch so konstruiert, dass möglichst alle erreichbaren Systemzustände von mindestens einem Testfall erreicht werden. Der Raum aller Systemzustände kann dabei während der Konfiguration des Testfallgenerators vom Testingenieur mit wenig Aufwand festgelegt werden. Das zweite Ziel des Testfallgenerators ist es, möglichst viele als „schlecht“ oder „kritisch“ eingestufte Systemzustände zu erreichen. Dies entspricht dem Konstruieren von Fahrmanövern, bei denen die Getriebesteuerung ein schlechtes Verhalten zeigt oder versagt. Die Bewertungskriterien werden bei der Konfiguration des Testfallgenerators festgelegt.

3 Anwendungsbeispiel MCT-Siebengang-Sportgetriebe

Die Softwaremodule der Getriebesteuerung für das MCT-Siebengang Sportgetriebe [1], bei dem der hydrodynamische Drehmomentwandler durch eine frei ansteuerbare hydraulische Kupplung ersetzt wurde, wurden mit dieser Testmethode entwickelt. Getestet wurde der original C-Code der Funktionssoftware mit den originalen Applikationsparametern. Die Adaptionswerte der Applikationssoftware wurden vor dem Test per Automatisierungsskript in der SiL-Umgebung automatisch ermittelt. Sie garantieren, dass die Funktionssoftware wäh-

rend des Tests optimal auf das verwendete Streckenmodell abgestimmt ist. Nur dann lassen sich Qualitätskriterien wie Schaltqualität und Bauteilbelastung sinnvoll in der SiL-Umgebung testen. Für die Überprüfung des Überwachungskonzepts wurde ein Streckenmodell eingesetzt, das Hydraulik- und Sensorfehler enthält. Der Testfallgenerator wurde so konfiguriert, dass die Variablen für Anlasser, Fahrpedal, Bremspedal, Wählhebel (P-R-N-D), Schalter für Fahrprogramm, Straßensteigung, Komponentenfehler und Öltemperatur vom Testgenerator als Inputs verwendet werden. Zum Beispiel erzeugte der Testgenerator für die Überprüfung eines Softwarestands 1100 Fahrmanöver mit einer Dauer von 50 Sekunden. Während der Simulationszeit von 15 Stunden wurden 235000 unterschiedliche Zustände erreicht und ausgewertet. Die einzelnen Fahrmanöver wurden automatisch bezüglich Schwachstellen im C-Code (zum Beispiel Division durch Null) und in der Logik (beispielsweise falsche Fehlerreaktion) analysiert. Die Wirksamkeit der Fehlerkorrekturen wurde anhand eines erneuten Testlaufs überprüft. Aufgrund der systematischen Vorgehensweise des Testautomaten wurden auch Fahrmanöver simuliert, die in der bisherigen Testspezifikation nicht enthalten waren. Die starke Automatisierung des Softwaretests und die hohe Wiederverwendbarkeit der Testkonfigu-

ration führen gegenüber einem manuellen Test zu einer deutlichen Erweiterung der Testabdeckung bei erheblich reduziertem Arbeitsaufwand für den Entwicklungsingenieur.

4 Zusammenfassung

Die zunehmende Komplexität von Softwaremodulen bedingt einen immer höheren Aufwand für den Test der Software. Die in diesem Beitrag vorgestellte Methode setzt einen Testautomaten ein, der autonom tausende von Fahrmanövern erzeugt, per Simulation ausführt und bewertet. Grundlage für die Testautomatisierung ist ein Streckenmodell von Fahrzeug und Getriebe, das hinreichend genau das reale System abbildet. Aufgrund des hohen Automatisierungsgrades wird der Testaufwand für den Entwicklungsingenieur erheblich reduziert und die Testabdeckung bezüglich der erreichten Systemzustände deutlich gesteigert. Durch eine zusätzliche Parallelisierung der Tests auf mehreren Simulationsrechnern kann der Testprozess weiter beschleunigt werden.

Literaturhinweise

- [1] G. Korherr, C. Dörr, A. Rink, R. Wörner: Neues Automatikgetriebe im PKW Hochleistungssegment, in: VDI Bericht 2029 zum VDI-Kongress Getriebe in Fahrzeugen, 2008
- [2] A. Junghanns, J. Mauss, M. Tatar: Testautomatisierung nach dem Schachspielerprinzip. In: C. Gühmann (Hrsg.): Simulation und Test in der Funktions- und Softwareentwicklung für die Automobilelektronik, Expert Verlag Renningen, pp. 320 - 331, 2008
- [3] A. Junghanns, J. Mauss, M. Tatar: TestWeaver – A Tool for Simulation-based Test of Mechatronic Designs – In: Proceedings of the 6th International Modelica Conference, Bielefeld, 3.-4.3.2008
- [4] Homepage der Modelica Association, siehe <http://www.modelica.org>

Download des Beitrags unter
www.ATZonline.de

ATZ
online

ATZ
elektronik

Read the English e-magazine.
Order your test issue now:
SpringerAutomotive@abo-service.info