

# **Diagnose in verteilten automotiven Systemen**

**Heinrich Balzer, Software Quality Lab, Universität Paderborn**

*Warburger Straße 100, D-33098 Paderborn*

*HBalzer@s-lab.uni-paderborn.de*

**Benno Stein, Bauhaus Universität Weimar**

*Bauhausstraße 11, D-99423 Weimar*

*Benno.Stein@medien.uni-weimar.de*

**Oliver Niggemann, dSPACE GmbH**

*Technologiepark 25, D-33100 Paderborn*

*ONiggemann@dspace.de*

## **Zusammenfassung**

Der vorliegende Beitrag zielt auf die Entwicklung und Evaluierung neuer Diagnoseverfahren für mechatronische Systeme ab. Unser aktueller Anwendungsschwerpunkt sind verteilte automotiv Systeme: Hier ist die Diagnose angesichts der steigenden Komplexität der Steuergerätenetzwerke eine große Herausforderung. Wir stellen einen neuartigen Diagnoseansatz vor und diskutieren mögliche Anwendungsszenarien wie die Einbindung von Diagnosefunktionalität in den Entwurfsprozess.

Unser Ansatz nutzt die Tatsache, dass das Wissen über Kausalitäten in verteilten Systemen, das für die Diagnose benötigt wird, implizit in Verhaltensmodellen enthalten ist. Solche Modelle werden in der Automobilindustrie im Entwurfsprozess erstellt. Durch die Simulation des Modells eines zu untersuchenden (Teil-) Systems in unterschiedlichen Fehlermodi werden Ursache-Wirkungs-Ketten identifiziert, die aufgrund der hohen Komplexität der Systeme oft nicht für Experten erkennbar sind. Um die Zuordnung von Fehlern zu gegebenen Symptomen zu erreichen, werden die Simulationsergebnisse durch Methoden des Data Mining generalisiert. Diese Vorgehensweise nennen wir Modellkompilation. Die Ergebnisse einer ersten Fallstudie mit Modellen aus der Automobilindustrie zeigen die Leistungsfähigkeit des Ansatzes.

## **Schlüsselwörter**

Automotive Diagnose, Modellkompilation, Simulation, Diagnostic Data Mining

## 1 Einleitung

Moderne Kraftfahrzeuge enthalten eine große Zahl mechatronischer Systeme wie z.B. ABS, Brake-by-wire oder das Automatikgetriebe. Ursache hierfür ist der Wunsch nach mehr Sicherheit und mehr Komfort im Kraftfahrzeug, was zu einer immer komplexeren Fahrzeugelektronik und einer steigenden Zahl miteinander vernetzter Steuergeräte führt. Neue Tendenzen gehen dahin, Funktionen, wie z.B. die adaptive Fahrgeschwindigkeitsregelung (ACC) oder das „lane keeping“ (Einhalten der Fahrspur), auf mehrere Steuergeräte zu verteilen.

Tritt ein Fehler in einem mechatronischen System im Kraftfahrzeug auf, so kann er nicht direkt, sondern nur über seine Auswirkungen (Symptome) beobachtet werden. Symptome stellen Abweichung des beobachteten vom erwarteten Verhalten dar und werden in den Softwaremodulen der Steuergeräte erkannt. Bild 1 skizziert den Aufbau eines Steuergerätes aus Softwaresicht. Sensorwerte und Busignale werden durch unterschiedliche Hardware- und Softwareschichten an Applikationssoftwaremodule gesendet. Dort werden die Daten verarbeitet und an Aktuatoren oder Busse weitergeleitet. Die Symptomererkennung kann nur entfernt vom möglichen Fehlerentstehungsort, in den Applikationssoftwaremodulen, implementiert werden, d.h., dass alle Symptome aus den unterschiedlichen Schichten erst dort erkannt werden können. Dieser Sachverhalt macht es schwierig, zwischen Fehlern aus unterschiedlichen Schichten zu differenzieren.

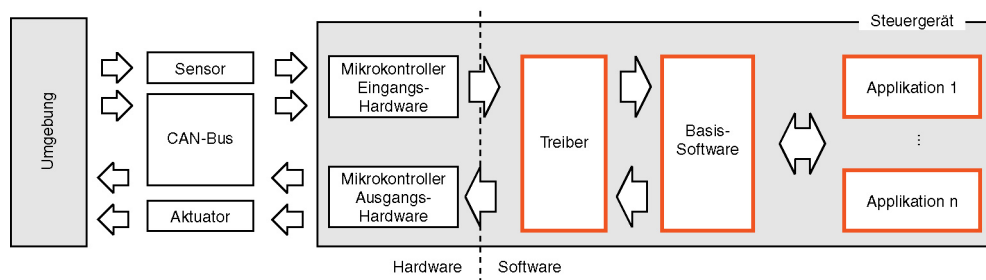


Bild 1: Die Hardware- und Softwareschichten eines Steuergerätes.

Da in einem verteilten System fehlerhafte Werte über Sensoren oder Kommunikationsbusse an Steuergeräte gesendet, von diesen verarbeitet und an andere Steuergeräte weitergeleitet werden, pflanzen sich Symptome in dem Steuergeräte-Netzwerk fort und können Fehlermeldungen sowie eine große Anzahl von Symptomen verursachen. Deshalb ist die Diagnose, also die Lokalisierung der eigentlichen Ursache, des „first principle“, in solchen verteilten automotiven Systemen eine große Herausforderung. Eine Untersuchung der Audi AG ergab beispielsweise, dass ein Steckfehler am Gateway 38 Fehlermeldungen in fünf Steuergeräten verursachte [Rei06].

Um diesen Herausforderungen zu begegnen, werden innovative Ansätze für die Diagnose benötigt. Deshalb ist in den letzten Jahren in der Automobilbranche ein Umdenken – weg von klassischen Diagnoseansätzen, die oft eine direkte Relation zwischen Symptomen und Fehlerursachen unterstellen und die Komplexität des Systemkontextes meist außer Acht lassen, hin zu neuen, mächtigeren Ansätzen – zu beobachten.

Wichtige und auch in der Automobilindustrie eingesetzte Diagnoseansätze sind die modellbasierte Diagnose [SP04, SKI02], die heuristische Diagnose und die wahrscheinlichkeitsbasierte Diagnose [Har06]. Puppe gibt eine Einführung in diese Ansätze [Pup90].

Ziel einer Fehlerdiagnose ist *(i)* die Erkennung von Symptomen, *(ii)* die Identifizierung von Fehlern und *(iii)* das Einleiten von entsprechenden Gegenmaßnahmen.

Unser Beitrag behandelt die Erkennung von Symptomen und die Identifizierung von Fehlern. Grundlegend ist ein neues Diagnoseverfahren für automotiv Systeme, das im folgenden Kapitel vorgestellt wird: die Modellkompilation [Ste01, Hus01, Ste03]. Das Kapitel gibt eine Einführung in die grundlegende Idee, zeigt die Anwendbarkeit des Ansatzes in einer Fallstudie und diskutiert mögliche Anwendungsszenarien für kompilierte Diagnosemodelle.

## 2 Diagnose durch Modellkompilation

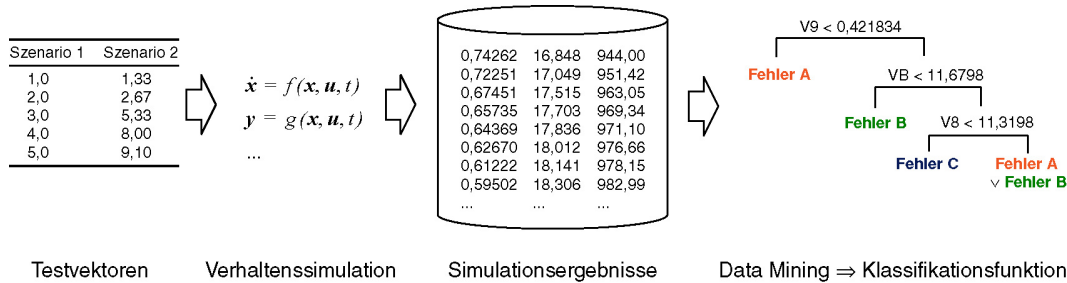
Idee des Modellkompilationsansatzes ist es, eine Klassifikationsfunktion zu lernen, die aus beobachteten Werten des Systemverhaltens auf ein Element einer Fehlermenge<sup>1</sup> abbildet. Dafür ist Wissen über den Kausalzusammenhang von Symptomen und Fehlern notwendig. Dieses Wissen ist implizit in Verhaltensmodellen (siehe Definition 1) mit hohem Detaillierungsgrad enthalten, die im Entwurfsprozess für mechatronische Systeme erstellt werden, und kann durch die Simulation der Modelle im Fehlerfall und im Nicht-Fehlerfall sowie das Protokollieren und Speichern von Variablenwerten generiert werden. Typischerweise wird hierbei das Modell in einer geschlossenen Regelschleife mit einem Streckenmodell simuliert.

Durch Anwendung spezieller Differenzoperatoren auf Simulationsergebnisse im Fehlerfall und im Nicht-Fehlerfall lassen sich die Symptome berechnen. Um die Ergebnisse zu generalisieren und allgemein anwendbar zu machen, werden Methoden des Data Mining und des maschinellen Lernens eingesetzt, mit deren Hilfe eine Abbildung von den Simulationsergebnissen auf die Fehlermenge approxi-

---

<sup>1</sup> Die Fehlermenge enthält auch das Element „Kein Fehler“.

miert wird. Die resultierende Klassifikationsfunktion lässt sich als ein kompliziertes Diagnosemodell verstehen. Bild 2 illustriert prinzipielle Schritte der Diagnose durch Modellkompilation.



*Bild 2: Überblick über die Modellkompilation. Anhand von Testvektoren werden Verhaltensmodelle im Fehler- und Nicht-Fehlerfall simuliert. Hieraus wird eine Klassifikationsfunktion gelernt.*

Das Prinzip der Modellkompilation ist hervorragend zur Diagnose in verteilten automotiven Systemen geeignet:

- In der Automobilindustrie existiert eine große Anzahl an ausführbaren Verhaltensmodellen. Diese können unmittelbar für die Simulation eingesetzt werden.
- Die Verhaltensmodelle werden in ihrer „bestimmungsgemäßen Inferenzrichtung“ analysiert. Sie müssen weder in spezielle abduktive Diagnosemodelle umgewandelt werden, noch müssen komplizierte Rückwärtssimulationen durchgeführt werden.
- Die gelernte Klassifikationsfunktion ist nahtlos in existierende Diagnoseansätze integrierbar. Zum Beispiel werden in der Werkstattdiagnose häufig Fehlerbäume eingesetzt, die durch die Klassifikationsfunktion, die je nach eingesetzter Lernmethode durch einen Entscheidungsbaum operationalisiert ist, ersetzt werden können.
- Die Auswertung eines kompilierten Diagnosemodells ist nicht rechenaufwendig. Andere Standarddiagnoseansätze wie die GDE erfordern die Ausführung und Analyse eines Verhaltensmodells zur Laufzeit [KW89].

## 2.1 Formale Beschreibung

Voraussetzung zur Anwendung der Modellkompilation ist, dass das zu diagnostizierende System durch ein ausführbares Verhaltensmodell repräsentiert wird. In diesem Abschnitt wird ein Verhaltensmodell formal definiert, um dann darauf aufbauend das Prinzip der Modellkompilation formal zu fassen.

**Definition 1:** Ein Verhaltensmodell  $M$  ist ein Tupel  $(F_U, F_Z, F_Y, V, \Delta, \Lambda)$ , dessen Elemente wie folgt definiert sind:

- $F_U, F_Z$  und  $F_Y$  sind die Mengen der Eingangsvariablen, der Zustands- und algebraischen Variablen und der Ausgangsvariablen.  $V$  enthält die Wertebereiche aller Variablen.
- $\Delta$  ist die globale Zustandsübergangsfunktion, die eine Menge von Zustandsvariablen  $F_X \subseteq F_Z$ , sowie den dazugehörigen Zustandsraum definiert.
- $\Lambda$  ist die Ausgangsfunktion und bildet von  $F_U \cup F_Z$  nach  $F_Y$  ab.

Verhaltensmodelle verteilter automotiver Systeme können unterschiedliche Zeitbasen besitzen: kontinuierlich, diskret oder diskret-ereignisorientiert. Enthält das Verhaltensmodell  $M$  eines Systems die Verhaltensmodelle  $M_1, \dots, M_k$  von Subsystemen mit unterschiedlichen Zeitbasen, so heißt das Modell  $M$  „hybrid“. Zwei Verhaltensmodelle,  $M_i, M_j$ , von Subsystemen sind gekoppelt, wenn Ausgangsvariablen von  $M_i$  mit Eingangsvariablen von  $M_j$  unifiziert sind.

Um ein Verhaltensmodell im Fehlerfall zu simulieren, muss es um Fehlerverhalten im Sinne der GDE+ erweitert werden [KW89, Str89, SD89]. Dafür wird eine zusätzliche Menge an Zustandsvariablen  $F_D$ , die die Fehlerzustände der Komponenten definieren, und die dazugehörige Zustandsübergangsfunktion  $\Delta^F$  formuliert. Für jeden zu simulierenden Fehler muss ein entsprechendes Fehlverhaltensmodell  $M^F$  bereitgestellt werden.

Sei  $\sigma$  der Simulationsoperator; in unserem Fall also das Simulationsprogramm. Weiter seien  $\sigma(M, u(t))$  und  $\sigma(M^F, u(t))$  die Simulationsergebnisse des fehlerfreien Modells  $M$  und eines fehlerhaften Modells  $M^F$  für ein Eingangssignal  $u(t)$ . Durch Anwendung eines Differenzoperators<sup>2</sup>  $\ominus$  auf die fehlerlosen und fehlerhaften Werte, kann eine Datenbank  $C_\Delta$  mit Vektoren von Symptomen erstellt werden:

$$\sigma(M, u(t)) \ominus \sigma(M^F, u(t)) = C_\Delta \quad (1)$$

In einem zweiten Schritt wird hieraus mit Methoden des maschinellen Lernens und des Data Mining eine Klassifikationsfunktion erstellt. Diese Funktion ist eine Abbildung von der Symptommenge auf die Fehlermenge:

$$C_\Delta \rightarrow F_D \quad (2)$$

Alternativ zu dem Vorgehen in zwei Schritten können die Methoden des maschinellen Lernens auch direkt auf die Simulationsergebnisse angewandt werden:

---

<sup>2</sup> Der Differenzoperator ist für jede physikalische Größe spezifisch definiert. Er berücksichtigt u.a. die Verteilung von Abweichungen und deren technischen Möglichkeiten der Beobachtbarkeit [Ste01].

$$C \rightarrow F_D \quad \text{mit } C = \sigma(\mathbf{M}, u(t)) \cup \sigma(\mathbf{M}^F, u(t)) \quad (3)$$

Bei dieser Alternative muss das Lernverfahren zusätzlich implizit die Differenzoperation berechnen. Folglich ist der Ansatz weniger leistungsfähig. Andererseits erlaubt diese Alternative den direkten Einsatz der erlernten Klassifikationsfunktion, ohne  $\mathbf{M}$  simulieren und den Differenzoperator anwenden zu müssen. In der Fallstudie, die im nächsten Abschnitt vorgestellt wird, wurde diese zweite Alternative angewandt.

## 2.2 Fallstudie

In den mechatronischen Systemen eines Kraftfahrzeugs können Fehler an unterschiedlichen Orten auftreten; zum Beispiel können Kurzschlüsse in der Elektrik vorliegen, in Bussen Nachrichten verfälscht werden oder verloren gehen, oder Fehler in der Hardware oder Software des Steuergerätes auftreten. Wir haben zur Demonstration der Leistungsfähigkeit des Modellkompilationsansatzes eine Fallstudie durchgeführt, in der Fehler am Gaspedalsensor eines Benzinmotorsteuergerätes zu diagnostizieren waren.

### 2.2.1 Verhaltensmodell

Für die Simulation des Verhaltens von Motor und Motorsteuergerät wird ein sehr genaues, differential-algebraisches Verhaltensmodell benötigt, welches das physikalische Verhalten realistisch nachbildet. Da der Motor und sein Steuergerät nicht unabhängig im Kraftfahrzeug existieren, sondern mit weiteren Komponenten interagieren, sind auch deren Verhaltensmodelle in die Simulation einzubeziehen.

In unserer Fallstudie wurde das „ASM – Gasoline Engine Simulation Package“ der Firma dSPACE verwendet [dSP05]. Es handelt sich dabei um ein vollständiges Modell eines 2,9-Liter-Motors mit 6 Zylindern, das die Modelle des Motorsteuergerätes, des Antriebs, der Fahrdynamik und der Umwelt integriert. Das Modell der Umwelt enthält u.a. das Fahrerverhalten und die Straßenverhältnisse. Die Modelle der einzelnen Komponenten bilden ein hybrides Fahrzeugmodell  $\mathbf{M}$ , das in Matlab<sup>®</sup>/Simulink<sup>®</sup> implementiert und durch mehr als 100 Zustände bestimmt ist.

Das Modell enthält einen Gaspedalsensor, dessen Sensorwerte an das Motorsteuergerät weitergeleitet werden. Dort werden die Daten benutzt, um über einen Aktuator die Position der Drosselklappe, die ein Subsystem des Motors ist, zu regeln. Die Drosselklappenposition beeinflusst andere Werte, wie z. B. das Drehmoment (siehe Bild 3).

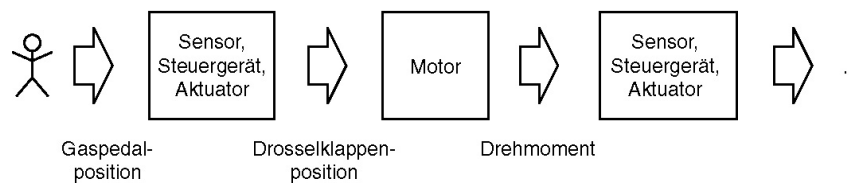


Bild 3: Signalfluss in der Fallstudie.

Im Folgenden wird das hybride Fahrzeugmodell  $M$  als die Kopplung von drei Modellen aufgefasst: dem Umgebungsmodell  $M_1$ , dem Modell der Motorsteuerung  $M_2$  und dem Modell der Software und Hardware für die Sensoren und Aktuatoren  $M_3$ , das die Signalwerte zwischen  $M_1$  und  $M_2$  weiterleitet. Für jeden Sensorfehler wird das entsprechende Fehlverhaltensmodell gebildet, indem in  $M$  das Modell  $M_3$  durch ein entsprechendes Fehlermodell  $M_3^F$  ersetzt wird. Jedes der Modelle  $M_3^F$  verändert die Sensordaten, so dass das Motorsteuergerät fehlerhafte Werte erhält. Bild 4 zeigt das Zusammenspiel der drei Modelle.

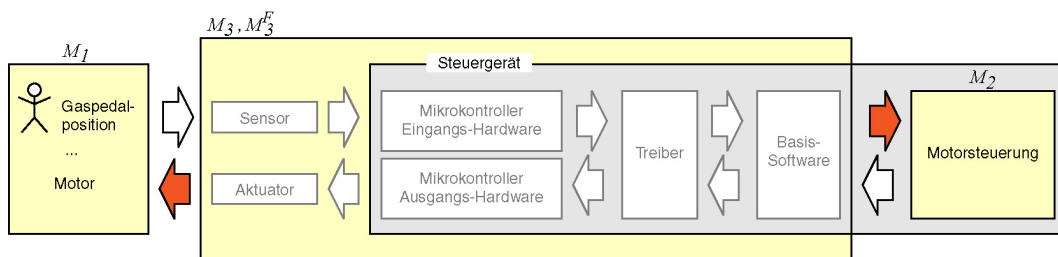


Bild 4: Zusammenspiel der drei Modelle  $M_1$  (Umgebung),  $M_2$  (Motorsteuerung) und  $M_3$  (Verbindung zwischen Motorsteuerung und Umgebung, die den Motor und die Benutzereingabe über das Gaspedal, enthält). Im Fehlerfall speist  $M_3^F$  fehlerhafte Sensorwerte in die anderen Modelle ein (dunkle Pfeile).

## 2.2.2 Betrachtete Fehlertypen

In unserer Fallstudie wurden die folgenden typischen Sensorfehler modelliert und simuliert:

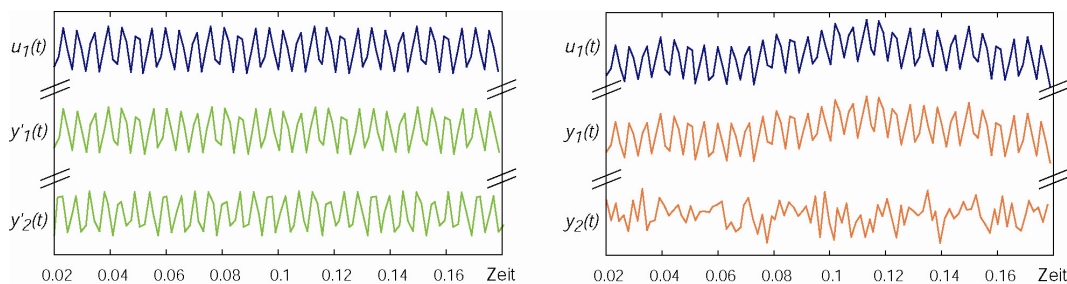
1. Der Signalwert ist auf 90% des korrekten Wertes reduziert (-10% Offset).
2. Der Signalwert ist auf 110% des korrekten Wertes erhöht (+10% Offset).
3. Der Signalwert ist mit Rauschen überlagert.
4. Das Signal fällt aus.

Entsprechend dieser vier Fehlerfälle manipuliert das jeweilige Modell  $M_3^F$  die von dem Gaspedalsensor gelieferten Signale, so dass die Motorsteuerung die fehlerhaften Signalwerte erhält, verarbeitet und an den Motor weiterleitet.

### 2.2.3 Simulation

Vor Beginn der Simulation werden  $p$  unterschiedliche Vektoren von Eingangsfunktionen definiert<sup>3</sup>. Jeder dieser Vektoren stellt ein Fahrscenario in einem festgelegten Zeitintervall dar. Mit einem Zufallsgenerator werden mehrere Zeitpunkte bestimmt, an denen der Fehler in einem Szenario auftreten soll. Jeder Simulationslauf ist durch den simulierten Fehlertyp, die fehlerhafte Komponente, das Szenario sowie den Zeitpunkt, an dem der Fehler auftritt, charakterisiert.

Die jeweiligen Modelle für das fehlerfreie und für das fehlerhafte Verhalten werden in den unterschiedlichen Szenarien simuliert. Während eines Simulationslaufs werden alle relevanten Daten protokolliert und gemeinsam mit den charakteristischen Informationen des Simulationslaufs in der Simulationsdatenbank  $C$  gespeichert. Dabei werden nur Daten gespeichert, die auch in einem realen System durch Sensoren ermittelt werden könnten. Bild 5 zeigt einen kleinen Ausschnitt der Ergebnisse einer Verhaltenssimulation im Nicht-Fehlerfall (links) und im Fehlerfall (rechts). Es wird deutlich, dass sich der Fehler im Gaspedalsensor auch auf andere Werte auswirkt.



*Bild 5: Ausschnitte der Simulationsergebnisse des fehlerlosen Modells  $M$  (linkes Diagramm) und eines fehlerhaften Modells  $M^F$  (rechtes Diagramm), bei dem der Sensorwert mit Rauschen überlagert ist.  $u_1(t)$ ,  $y_1(t)$  und  $y_2(t)$  bezeichnen den Gaspedalsensor, den Drosselklappensensor und das Drehmoment.*

<sup>3</sup> Die entsprechenden Eingangsvariablen beschreiben das Fahrerverhalten wie Bremsen, Gas geben oder Schalten.



## 2.2.4 Lernen der Klassifikationsfunktion

Durch die Simulationsläufe werden Ursache-Wirkungsketten quantifiziert und zu festen Zeitpunkten in die Simulationsdatenbank  $C$  geschrieben. Um von den konkreten Simulationsszenarien zu abstrahieren und zu generalisieren, werden Methoden des maschinellen Lernens bzw. des Data Mining angewandt, mit denen die Funktion aus Gleichung (3) gelernt wird.

Für diese Fallstudie haben wir zwei Lernalgorithmen eingesetzt: lineare Regression [Har99, WW81] und Entscheidungsbäume [BHO+84, Rip96]. Die daraus resultierenden Diagnosemodelle erzielen gute bis sehr gute Ergebnisse. Tabelle 1 stellt die Ergebnisse für beide Verfahren und die vier Fehlertypen dar. Außer bei dem Fehlertyp „ausgefallenes Signal“ erzielt der Entscheidungsbaum deutlich bessere Ergebnisse als die lineare Regression.

*Tabelle 1: Fehlerraten (prozentuale Anzahl falsch diagnostizierter Fälle) der Klassifikationsfunktionen für die vier Fehlertypen und die beiden Lernalgorithmen.*

	-10% Offset	+10% Offset	verraushtes Signal	ausgefallenes Signal
Lineare Regression	14,9%	16,5%	3,6%	0,1%
Entscheidungsbaum	1,1%	0,5%	0,3%	0,2%

## 2.3 Anwendungsszenarien

Die mit dem Modellkompilationsansatz erlernten Klassifikationsfunktionen sind in unterschiedlichen Szenarien anwendbar. In diesem Abschnitt stellen wir drei Anwendungsszenarien vor.

*Offline-Diagnose.* Die Offline-Diagnose findet vor allem in KfZ-Werkstätten statt. Ausgehend von elektronischen Fehlfunktionen oder Fehlerwarnungen an den Fahrer wird die Diagnose in der Werkstatt auf Basis der Daten aus dem Kraftfahrzeug, wie z.B. protokollierte Werte oder Fehlerspeichereinträge in den Steuergeräten, durchgeführt. Wichtige Ziele der Werkstattdiagnose sind die Reduzierung der In-Ordnung-Ausbauten<sup>4</sup> und der No-Trouble-Found-Rate<sup>5</sup> [Piq06, Rei06], was letztendlich nur durch leistungsfähigere Diagnosealgorithmen erreicht werden kann. Der Modellkompilationsansatz besitzt das Potenzial hierfür, weil er auf den „tiefen“ Verhaltensmodellen mechatronischer Systeme eines Kraftfahrzeugs beruht, und kann deshalb zur Generierung einer leistungsfähigeren Diagnosefunk-

<sup>4</sup> In-Ordnung-Ausbauten bezeichnen Komponenten, die in der Werkstatt ausgebaut und durch eine neue Komponente ersetzt werden, obwohl sie fehlerfrei sind.

<sup>5</sup> Die No-Trouble-Found-Rate ist die prozentuale Anzahl der Situationen, in denen trotz aufgetretener Symptome der Fehler nicht identifiziert werden konnte.

tion verwendet werden, die in der Werkstatt vorhandene Diagnosesoftware ergänzt und die Aufgabe der Fehleridentifizierung übernimmt.

*Online-Diagnose.* Viele Fehler sollten bereits zum Zeitpunkt des Auftretens im Kraftfahrzeug identifiziert werden, damit entweder entsprechende Gegenmaßnahmen eingeleitet werden können oder der Fahrer eine Warnung erhält. Zusätzlich wird ein Fehlereintrag in den Fehlerspeicher des entsprechenden Steuergerätes geschrieben. Für die Online-Diagnose müssen Algorithmen implementiert werden, die zur Betriebszeit des Kraftfahrzeugs auf den Steuergeräten ausgeführt werden. Neben der gesetzlich vorgeschriebenen On-Board-Diagnose<sup>6</sup> (OBD) wird z.B. das Fehlermanagement von Fahrerassistenzsystemen wie AFS (advanced frontlighting system) und ACC (adaptive cruise control) online im Fahrzeug durchgeführt. Um online Symptome zu erkennen und um auf die Fehler zu schließen, wäre es denkbar, die mit dem Modellkompilationsansatz erzeugten Diagnosefunktionen in einem Steuergerät auszuführen. So könnten einige Fehler zuverlässiger identifiziert werden, was zur Erhöhung der Sicherheit der mechatronischen Systeme eines Kraftfahrzeugs beiträgt.

*Entwurfsprozess.* Im Entwurfsprozess für verteilte automotiv Systeme werden ausführbare Verhaltensmodelle erstellt. Auf Basis dieser Modelle wird der Programmcode für die Steuergeräte generiert. Dieselben Modelle können auch verwendet werden, um nach dem Prinzip der Modellkompilation schon früh im Entwicklungsprozess eine leistungsfähige Diagnosefunktion zu erstellen und zu testen<sup>7</sup>. Solche Diagnosefunktionen lassen sich für die Online-Diagnose in den erstellten Funktionscode integrieren oder für die Offline-Diagnose in Werkstatttestersoftware einsetzen.

### 3 Zusammenfassung und Ausblick

In diesem Beitrag wurde das Prinzip der Modellkompilation für die Diagnose in verteilten automotiven Systemen vorgestellt. Durch die maschinelle Auswertung von Daten, die aus der Simulation realer Modelle für das fehlerfreie und das fehlerhafte Verhalten gewonnen wurde, lässt sich ein kompiliertes Diagnosemodell extrahieren. Im Kern handelt es sich hierbei um eine Klassifikationsfunktion, mit der von Symptomen auf Fehler geschlossen wird.

In einer Fallstudie wurde dieses Prinzip auf verschiedene Fehler eines Gaspedalsensors angewandt. Dafür wurde ein hybrides Modell für ein Motorsteuergerät

---

<sup>6</sup> Die OBD realisiert eine Überwachung der abgasrelevanten Komponenten zur Einhaltung von Emissionsgrenzwerten, um Grenzwertüberschreitungen sofort anzeigen zu können.

<sup>7</sup> Voraussetzung hierfür ist die Möglichkeit, mit den Modellen fehlerhaftes Verhalten zu simulieren.

und die entsprechende Umgebung (Motor, Antrieb, Fahrzeugdynamik und die Umwelt) aus der Automobilindustrie verwendet. Es wurden zwei Lernverfahren, lineare Regression und Entscheidungsbäume, eingesetzt, die zu guten (85%) und zu exzellenten Fehlererkennungsraten (99%) führten. Die Robustheit, der hohe Grad an Automatisierbarkeit und nicht zuletzt die Ergebnisse der Fallstudie zeigen, dass der Modellkompilationsansatz eine ernsthafte Alternative zu den bislang eingesetzten Diagnoseverfahren in automotiven Systemen darstellt.

Gegenstand unserer weiteren Forschung sind die Analyse von komplexeren Fehlerszenarien und Mehrfachfehlern, die Anwendung mächtigerer Techniken zur Datenfilterung, wie Vektorquantisierung und Clusteranalyse, sowie optimierte Lernverfahren. Die Wahl des maschinellen Lernverfahrens ist der Schlüssel für den Erfolg der Modellkompilation. Hier sollen weitere Verfahren, wie Assoziationsregeln oder Bayessche Netze untersucht werden.

## Literatur

- [BHO+84] Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J.: Classification and Regression Trees. Wadsworth, 1984.
- [dSP05] dSPACE: ASM – Gasoline Engine Simulation Package. Flyer erhältlich unter [http://www.dspace.de/ww/de/gmb/home/products/sw/automotive\\_simulation\\_models/asm\\_gasoline\\_engine\\_simulation.cfm](http://www.dspace.de/ww/de/gmb/home/products/sw/automotive_simulation_models/asm_gasoline_engine_simulation.cfm), Paderborn, Deutschland, 2005.
- [Har06] Harr, T.: Implementierung von wahrscheinlichkeitsbasierten Onboard-Diagnosealgorithmen. 3. Internationale Fachkonferenz: Diagnosesysteme im Automobil, Stuttgart, Deutschland, 2006.
- [Har99] Hartung, J.: Statistik. Oldenbourg, 1999.
- [Hus01] Husemeyer, U.: Heuristische Diagnose mit Assoziationsregeln. Dissertation, Fachbereich Mathematik und Informatik, Universität Paderborn, 2001.
- [KW89] de Kleer, J; Williams, B. C.: Diagnosis with Behavioral Modes. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI 89), S. 1324-1330, Detroit, Michigan, USA, 1989.
- [Piq06] Piques, J.-D.: Model Based Diagnosis: Application to Electronic Throttle Control. 3. Internationale Fachkonferenz: Diagnosesysteme im Automobil, Stuttgart, Deutschland, 2006.
- [Pup90] Puppe, F.: Problemlösungsmethoden in Expertensystemen. Studienreihe Informatik, Springer, Berlin, Heidelberg, New York, 1990.
- [Rei06] Reich, A.: Diagnose – ein wesentlicher Faktor zur Beherrschung der Fahrzeugelektronik. 7. Euroforum-Jahrestagung: Software im Automobil, Stuttgart, Deutschland, 2006.
- [Rip96] Ripley, B. D.: Pattern Recognition and Neural Networks. Cambridge University Press, 1996.
- [SD89] Struss, P.; Dressler, O.: "Physical Negation" – Integrating Fault Models into the General Diagnostic Engine. In Proceedings of the Fifteenth International

- Joint Conference on Artificial Intelligence (IJCAI 89), Volume 2, S. 1318-1323, 1989.
- [SKI02] Schwarte, A; Kimmich, F.; Isermann, R.: Modellbasierte Fehlererkennung und -diagnose für Dieselmotoren. Motortechnische Zeitschrift (MTZ), 63. Jahrgang, Heft 7-8, Juli/August, 2002.
- [SP04] Struss, P.; Price, C.: Model-based systems in the automotive industry. AI Magazine, 24(4), S. 17-34, 2004.
- [Ste01] Stein, B.: Model Construction in Analysis and Synthesis Tasks. Habilitation, Fachbereich Mathematik und Informatik, Universität Paderborn, 2001.
- [Ste03] Stein, B.: Model Compilation and Diagnosability of Technical Systems. In Proceedings of the 3<sup>rd</sup> IASTED International Conference on Artificial Intelligence and Applications (AIA 03), Benalmdena, Spanien, ed., Hanza, M.H., S. 191-197. ACTA Press, 2003.
- [Str89] Struss, P.: Model-Based Diagnosis – Progress and Problems. In Proceedings of the International GI-Convention, Volume 3, S. 320-331, 1989.
- [WW81] Wonnacott, T.; Wonnacott, R.: Regression: A second course in statistics. John Wiles & Sons, New York, 1981.

## Autoren

**Heinrich Balzer** ist seit Abschluss seines Studiums der Wirtschaftsinformatik an der Universität Paderborn im Oktober 2005 wissenschaftlicher Mitarbeiter im s-lab (Software Quality Lab) der Universität Paderborn. Sein Arbeitsschwerpunkt ist die Diagnose in verteilten automotiven Systemen.

**Benno Stein** ist Leiter der Arbeitsgruppe Web-Technology & Information Systems an der Bauhaus-Universität Weimar. Schwerpunkt seiner Forschung ist die Modellierung und Lösung wissensintensiver Aufgaben der Informationsverarbeitung.

**Oliver Niggemann** ist Produktmanager bei der dSPACE GmbH in Paderborn. Er ist dort für Themen wie automotive Software-Architekturen und Werkzeuge für den System-Entwurf verantwortlich.