

Automatische Testauswertung für Steuergerätesoftware

Dr. Hans-Werner Wiesbrock*

IT Power Consultants
Gustav-Meyer-Allee 25
D 13355 Berlin
hans-werner.wiesbrock@itpower.de
Tel: +49 (0)30 46 77 69 47
Fax: +49 (0)30 46 30 76 49



* unterstützt durch das BMBF-Projekt IMMOS, Proj.-Ref. 01ISC31D

Zusammenfassung

In diesem Artikel werden die Chancen und Möglichkeiten als auch Grenzen einer automatisierten Test-Evaluation der Signale von Steuergerätesoftware aufgezeigt. Vorzugsweise werden dabei Testumgebungen betrachtet, in denen reproduzierbare Eingabesignale zum Treiben der Software verwendet werden können und die Signalausgaben der Steuergerätesoftware unverrauscht dem Tester zugänglich sind. Es werden neu entwickelte und implementierte Algorithmen vorgestellt, ihr systematischer Einsatz und die Erfahrungen mit der Tool-Unterstützung *MEval* beschrieben.

Inhaltsverzeichnis

Automatische Testauswertung für Steuergerätesoftware	1
Zusammenfassung.....	2
Inhaltsverzeichnis.....	2
Automatische Testauswertung, Einsatz und Chancen	2
Besonderheiten bei Steuergerätesoftware	4
Beispiel: Messdaten als Ausgabedaten im HiL.....	5
Beispiel: Nicht vollständig reproduzierbare Eingaben im HiL.....	5
Aufgaben eines automatisierten Vergleichs (SiL, MiL)	5
Algorithmen und Implementierung.....	6
Innovative mehrstufige Verfahren	7
Differenzmatrix-Verfahren	7
Das Standard-Verfahren	7
Standardschwellwerte	8
MEval	8
Einsatz und Möglichkeiten, Grenzen und bisherige Erfahrungen.....	9
Referenzen.....	10

Automatische Testauswertung, Einsatz und Chancen

Mikroelektronik und damit Software für Steuergeräte dringen in immer zahlreichere und auch sicherheitskritische Bereiche vor, insbesondere in der Automobilbranche. Schätzungen zeigen, dass im Jahre 2010 ca. 40 % der Wertschöpfung des Fahrzeugs durch Elektrik/Elektronik geprägt sein wird. Schon mehr als 2000 Einzelleitungen und 3800 m Kabel verbinden heute über vier Hauptbussysteme und weitere Sub-Busse etwa 60-70 Steuergeräte in einem Fahrzeug. Zwischen diesen Steuergeräten werden mehr als 250 CAN-Botschaften und über 2500 Signale ausgetauscht [1]. Der wachsende Elektronikanteil bedeutet aber nicht nur ein Mehr an Komfort, sondern stellt auch neue, hohe Anforderungen an die Qualität und Zuverlässigkeit der elektronischen Systeme sowie deren eingebettete Software. Fehlfunktionen in der Elektronik wirken sich immer häufiger fatal auf die Sicherheit von Personen und das Auftreten hoher Sachschäden aus.

Mit diesem Fortschreiten steigt also der Druck, die hohe Qualität und Sicherheit zu gewährleisten. Zugleich wächst aber auch der Kostendruck und die Suche nach Einsparungspotential. Die Entwicklung softwarebasierter, eingebetteter Steuerungs- und

Regelsysteme erfolgt seit Ende der 90er Jahre zunehmend modellbasiert [2]. Sie ist der vielversprechendste Weg, den divergierenden Aufgaben zu begegnen. Bei diesem Vorgehen wird eine ausführbare Funktionsspezifikation (Funktionsmodell) erstellt, die zunächst zusammen mit einem Strecken- / Umgebungsmodell simuliert und später direkt auf dem eingebetteten System (Steuergerät, ECU) implementiert werden kann. Die Modellierung erfolgt unter Verwendung von regelungstechnischen Blockschaltbildern und erweiterten Zustandsübergangsdiagrammen mit Werkzeugen wie MATLAB / Simulink / Stateflow. Dieser Entwicklungsprozess kann bereits frühzeitig durch geeignete Qualitätssicherungsmaßnahmen flankiert und abgesichert werden. Erst nach dem ausgiebigen Modelltest wird der Programmcode manuell oder (heute zunehmend) automatisch aus dem Modell generiert.

Ein iteratives Vorgehen nötigt zu aufwendigen Regressions- und Back-to-Back-Tests. Regressionstests stellen sicher, dass die Modifikationen im Modell oder im Code die schon korrekt umgesetzte Funktionalität der älteren Version ebenfalls erfüllen. Back-to-Back-Tests dienen dem Nachweis der Äquivalenz zwischen den verschiedenen Repräsentationsformen der Software, z.B. zwischen Modell und dem aus dem Modell generierten Programmcode. Beim fortschreitenden Regressions- oder Back-to-Back-Test kann ein und derselbe Testdatensatz verwendet werden. Somit dient ein auf Korrektheit geprüfter Ausgabedatensatz als Referenz für nachfolgende Tests. Das Problem der Testauswertung reduziert sich auf den Vergleich der aktuellen Ausgangssignalwerte mit den Referenzwerten. Eine Automatisierung solcher Vergleiche bedeutet ein hohes Einsparpotential bei gleichzeitiger Qualitätssteigerung. In herkömmlicher Softwareentwicklung wird diese Aufgabe mit weitgehend automatisierten Tests und deren Auswertung gelöst [3]. Anders jedoch sieht es bisher in der Entwicklung von Steuergerätesoftware aus. Zwar sind bereits gute Testwerkzeuge auf dem Markt vorhanden und werden auch erfolgreich eingesetzt, jedoch hakt es bisher an einer automatisierten Testauswertung. Immer noch hat der Tester durch „scharfes Hingucken“ zu beurteilen, ob der Test bestanden wurde.

In diesem Aufsatz werden neue Ansätze dazu vorgestellt sowie ein neues Werkzeug, *MEval*, welches diesen automatisierten Vergleich durchführt.

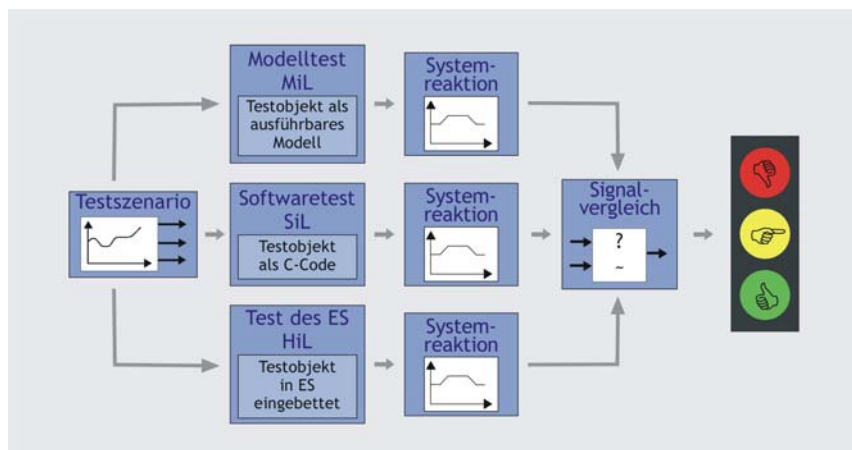


Abbildung 1 Diversitäre Tests in modellbasierter Softwareentwicklung

In Abbildung 1 werden noch einmal die verschiedenen Entwicklungssituationen aufgezeigt, in denen ein automatisierter Vergleich von Test- und Referenzergebnissen durchgeführt werden kann. Die Vorteile eines solchen Vorgehens liegen auf der Hand:

- ↳ Effektiveres und effizienteres Testen
- ↳ Größere Testabdeckung durch Automatisierung
- ↳ Höhere Qualität durch Operationalisierung
- ↳ Reproduzierbare und nachweisbare Testqualität
- ↳ Hohes Einsparpotential: Personal und Zeit

Durch den Einsatz automatisierter Testauswertung können auch umfangreichere Tests durchgeführt werden und der Tester kann sich auf die Erfassung der wichtigen und nötigen Testfälle konzentrieren. Definierte Vergleichskriterien (Arbiter im Sprachgebrauch des UML-Testing Profiles) führen zu reproduzierbaren Testauswertungen, die Beurteilung unterliegt so nicht mehr dem subjektiven Eindruck des Testers.

Besonderheiten bei Steuergerätesoftware

So viele Vorteile, aber warum wird dann eine solche Technik bisher nicht eingesetzt? Der Grund liegt in den spezifischen Problemen einer Testauswertung für Steuergerätesoftware. Im Gegensatz zu Datenbankapplikationen und vielen anderen Anwendungsbereichen ist es für eingebettete elektronische Systeme charakteristisch, dass sie kontinuierlich über Sensoren und Aktuatoren mit ihrer Außenwelt interagieren, zeitveränderliche Sensorsignale verarbeiten und zeitveränderliche Ausgaben erzeugen. Beim Test müssen deshalb zeitbehaftete Signalausgaben verglichen und bewertet werden.

Dieses Aufgabenfeld beschränkt aber auch den Rahmen der zu betrachtenden Daten. Die folgenden Eigenschaften können für den Vergleich von Signalpaaren bestehend aus einem Referenzsignal $o(t)$ und einem Outputsignal $o'(t)$ vorausgesetzt werden:

- $o(t)$ und $o'(t)$ sind zeitdiskret.
- $o(t)$ und $o'(t)$ haben die gleiche Länge.
- Die Datentypen (Wertebereiche) von $o(t)$ und $o'(t)$ sind gleich. Sowohl quasikontinuierliche als auch n-äre oder binäre Signale können auftreten.
- I.d.R. handelt es sich um nichtperiodische Verläufe.
- Die Reihenfolge bestimmter Merkmale (Anstiege, Nulldurchgänge, ...) in den Signalverläufen ist signifikant.
- Sowohl zeitliche als auch Amplitudenabweichungen sind von Interesse

Gängige Signalvergleichsverfahren berücksichtigen diese besonderen Bedingungen i.a. nur unzureichend und führen damit nicht zu zufriedenstellenden Auswertungsergebnissen. So sind statistische Verfahren zwar adäquat zum Herausfiltern stochastischer Störeinflüsse, versagen jedoch bei der Bewertung der Gleichheit zweier Signale, da z.B. peakförmige Abweichungen durch Mittelungen unentdeckt bleiben können.

Aber auch die verschiedenen Testumgebungen und -situationen sind u.U. zu berücksichtigen. So lassen sich i.A. Regressions- und Back-To-Back Tests im Falle von Model-in-the-Loop (MiL) und Software-in-the-Loop (SiL) mit reproduzierbaren Eingabesignalen treiben, im Falle von Hardware-in-the-Loop (HiL) ist dies jedoch nicht mehr immer möglich. Auch sind die Ausgaben bei HiL häufig konkrete Messdaten, die für Back-To-Back Tests noch geeignet interpoliert werden müssen. Dementsprechend werden sich Abweichungen in den jeweiligen Ausgaben systematisch unterscheiden. Kann man im Falle von MiL oder SiL Tests noch annähernd gleiche Resultate erwarten, wird man im HiL Fall nur eine ähnliche ‚Gestalt‘ oder gleiche Merkmale der Signale vermuten.

Beispiel: Messdaten als Ausgabedaten im HiL

Um die Signaldaten mit den Ausgangsdaten der Modelltests vergleichen zu können, sind diese Daten zunächst auf die selben Zeitschritte zu extrapolieren. Eine Möglichkeit dazu ist die lineare Interpolation der Daten. Die systematische Ungenauigkeit wird dann von der jeweiligen Steigung abhängen. Zudem kann eine Messungenauigkeit vorliegen, die als Rauschen eingeht. Deshalb filtert man eventuell die hohen Frequenzen heraus, glättet also die Signale. Erst dann kann man die gewonnenen Daten mit den Ausgangsdaten vom MiL vergleichen. Eine automatisierte Auswertung sähe dann wie folgt aus. Nach geeigneter Interpolation, deren Auswahl direkt vom Signal- und Messkontext abhängt, wird das Signal geeignet geglättet und anschließend über eine steigungsabhängige Differenzenbildung mit der MiL-Ausgabe verglichen. Die letzten Schritte können dabei automatisiert vom Tool *MEval* durchgeführt werden, siehe Abschnitt *MEval* weiter unten.

Beispiel: Nicht vollständig reproduzierbare Eingaben im HiL

Die Eingabedaten zum Treiben der Software können nicht vollständig reproduzierbar eingegeben werden. Eine typische Situation ist beispielsweise ein Fahrversuch. In diesem Fall lassen sich auch durch Interpolation und Glättungen der Signaldaten die Ausgaben nicht direkt mit den MiL oder vorherigen Resultaten z.B. vergleichen. Hier ist eine neue Technologie anzuwenden. Die Idee ist dabei folgende: Bestimmte Merkmale und Eigenschaften, Charakteristika der Signaldaten sind zunächst zu spezifizieren sowie deren zeitliche Abhängigkeiten. So definierte Forderungen können dann operationalisiert und automatisiert überprüft werden. IT Power Consultants entwickelt dazu gerade neue Konzepte, die in ein neu zu entwickelndes Werkzeug einfließen werden.

Aufgaben eines automatisierten Vergleichs (SiL, MiL)

Für Tests im SiL oder MiL können wir reproduzierbar die Eingabesignale setzen und die Ausgaben unverzerrt erhalten. Allerdings spielen hier andere systematische Fehler eine Rolle. Diskretisierungs- und Rundungsfehler machen sich in geringen zeitlichen Abweichungen oder auch in den Signalwerten bemerkbar.

Wie im Abschnitt *Besonderheiten bei Steuergerätesoftware* erläutert, erfordern die fundamentalen Unterschiede in der Zielsetzung eines Signalvergleiches zu bekannten Problemen der Nachrichtenübermittlung oder dem allgemeinen Kurvenvergleich andere Verfahren zur Automatisierung der Testauswertung. Als Beispiel sei hier das bekannte Toleranzschlauchverfahren genannt, welches eine einfache Fenstertechnik implementiert.

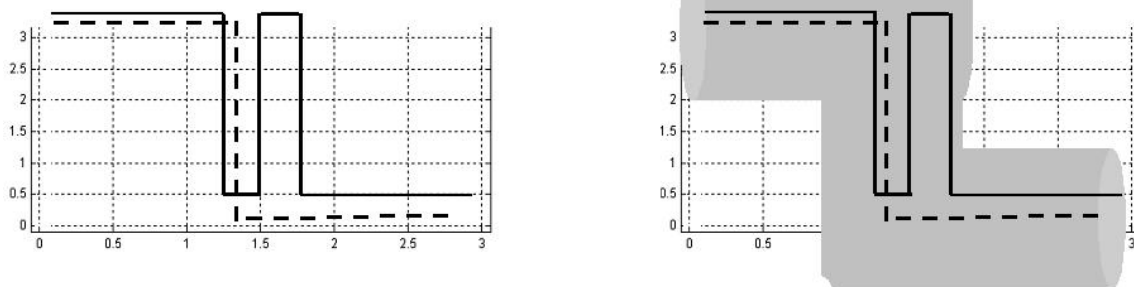


Abbildung 2 Das Toleranzschlauchverfahren

Um das Referenzsignal wird ein geeigneter Schlauch mit vorgegebenem Durchmesser gelegt.

Falls das Testsignal innerhalb dieses Schlauches liegt, sollen die beiden Signale als gleich angesehen werden.

Dieses Verfahren ist für einen Kurvenvergleich sehr verbreitet, nicht jedoch geeignet für eine vertrauenswürdige Testbewertung. In Abbildung 2 ist ein Beispiel stilisiert, welches in einem konkreten Projekt einer Motorsteuerungssoftware auftrat. Irrtümlich sprang das neue Testsignal wieder kurzzeitig auf das Anschaltflag. Da die gesamte Testsequenz eine lange Dauer hatte, wurde dieser Fehler beim „scharfen Hingucken“ übersehen. Aber auch mit dem Toleranzschlauchverfahren konnte dieser Fehler nicht entdeckt werden.

Die Aufgabe, die eine dringend benötigte automatische Testauswertung also zu erfüllen hat, umfasst:

- Diskretisierungs- und Rundungsfehler müssen erkannt und kontrolliert toleriert werden.
- Mögliche vorhandene Fehler müssen entdeckt werden.
- Systematische Fehler sollten erkannt werden.

Die Ergebnisse sind dann dem Tester verdichtet zu präsentieren, um ihn in seiner Arbeit und Entscheidung zu unterstützen. Dies bedeutet insbesondere:

- Lokalisierung der Fehlerstellen
- Kompakte visuelle Darstellungen
- Aussagekräftige Kenngrößen.

IT Power Consultants (www.itpower.de) hat in Zusammenarbeit mit der DaimlerChrysler AG neue Algorithmen zur Lösung dieser Aufgaben entwickelt und im Tool *MEval* implementiert. Mit diesem Analysewerkzeug können Signale unter gesonderter Berücksichtigung zeitlicher Verzögerungen und Amplitudenabweichungen miteinander verglichen werden.

Algorithmen und Implementierung

Der Signalvergleich muss dem Anwendungsbereich spezifisch anpassbar sein, er muss zeitliche Verzögerungen kontrolliert zulassen können, wie auch geringe Amplitudenabweichungen erlauben, hingegen vertauschte Signalfolgen und zu große Abweichungen als Fehler markieren.

Innovative mehrstufige Verfahren

Um die vielfältigen Anforderungen an einen Vergleichsalgorithmus zu erfüllen, wurde ein innovatives mehrstufiges Verfahren vorgeschlagen [4,5]. Die Signale werden zuerst in geeigneter Weise vorverarbeitet und anschließend nach elementaren Kriterien auf ihre Ähnlichkeit verglichen. Zeitliche Verschiebungen und Amplitudenabweichungen werden auf diese Weise getrennt voneinander identifiziert und kritisch beurteilt.

Differenzmatrix-Verfahren

Für die Vorverarbeitung wurde speziell das neue Differenzmatrix-Verfahren entwickelt (Abb. 3). Bei diesem Verfahren wird das Testsignal an das Referenzsignal angepasst, indem es zeitlich gestreckt oder gestaucht wird. Die Stärke der Streckung oder Stauchung gibt einen Hinweis auf die zeitliche Verzögerung.

Die Grundidee ist folgende: Angenommen wir hätten zwei zeitlich zueinander verschobene Signale auf ihre Gleichheit hin zu untersuchen. Wir verschieben dazu das eine Signal auf das andere und würden die verbleibende Abweichung begutachten. Und genau diese Schritte werden in dem neuen Verfahren abgebildet. Ein Verschieben des einen Signals auf das andere entspricht einem zeitlichen Shift, mathematisch formuliert, einer Reparametrisierung des einen Signals, $t \neq t + \Delta$. Allgemeiner suchen wir nun nach einer Reparametrisierung, so dass das eine Signal möglichst gut auf dem anderen Signal liegt.

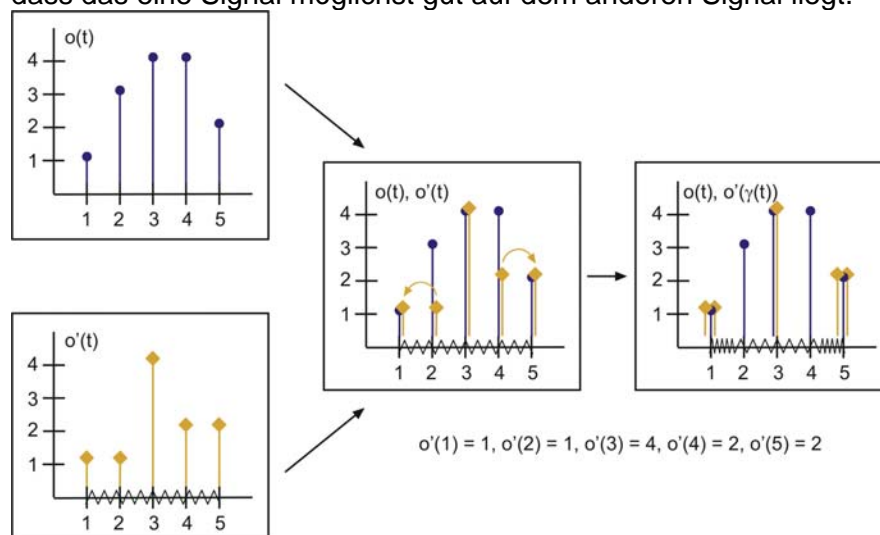


Abbildung 3 Reparametrisierung des Testsignales

Angenommen die Signalwerte säßen auf einer Federkette. Dann verschieben wir die Testsignalwerte so, dass sie möglichst gut zu den Referenzwerten passen. Die Stauchungen und Streckungen der Federn geben dann ein Maß für die zeitlichen Abweichungen.

Die symmetrische Variante, bei der beide Signale jeweils an das andere angepasst wird, erlaubt dann eine genaue Untersuchung der zeitlichen Abweichungen, also, ob das eine Signal in gewissen Zeitbereichen nur zeitlich etwas verschoben zu dem anderen liegt oder verschieden ist.

Das Standard-Verfahren

Die Performanz des (symmetrischen) Differenzmatrix-Verfahrens ist linear zur Anzahl der Signalpunkte. Um diese noch zu steigern, werden im Standard-Verfahren zunächst die Intervalle mit signifikanter Amplitudenabweichung identifiziert und die weiteren Vergleiche

über das (symmetrische) Differenzmatrix-Verfahren auf ihnen beschränkt. Nach den zeitlichen Anpassungen werden die verbliebenen Amplitudenabweichungen über eine geeignet zu wählende Differenzbildung beurteilt. Überschreitet ihre absolute Differenz beispielsweise nicht einen vorgegebenen Schwellwert $\varepsilon > 0$, so werden die Signale als gleich angesehen.

Nun wird man je nach Anwendungsbereich und Signal verschiedene Schwellwerte zu setzen haben, eine Automatisierung wird dadurch erschwert. Lässt sich noch für einen Anwendungsbereich oder Signaltypen angeben, dass nur geringe, mittlere oder auch noch großzügigere Abweichungen erlaubt sind, so bleibt doch noch die Abhängigkeit vom konkreten Signal. Wenn die Signalwerte durchschnittlich sehr hoch liegen und stark variieren, wird man vermutlich einen größeren Schwellwert akzeptieren, da Rechenungenauigkeiten und Rundungsfehler sich hier deutlicher bemerkbar machen können. Um das vorgeschlagene mehrstufige Vergleichsverfahren zu automatisieren und Benutzerinteraktionen zu minimieren, werden deshalb Toleranzschwellwerte und weitere benötigte Verfahrensparameter aus charakteristischen Größen des Referenzsignals abgeleitet, unterstützt durch eine statistische Analyse. Die verbleibenden Parameter können dann anwendungsspezifisch gesetzt werden.

Standardschwellwerte

Basierend auf der Standardabweichung und dem Mittelwert des Referenzsignals werden Default-Toleranzschwellen bestimmt. Der jeweilige Proportionalitätsfaktor legt die Empfindlichkeit des Signalvergleichs fest. Je kleiner er gewählt wird, umso geringer die noch zugelassene Abweichung:

$$tol_{absDiff} \sim \max(|o(t) - E(o(t))|) + \sigma(o(t))$$

$$tol_{relDiff} \sim tol_{absDiff} / \max(E(|o(t)|), 1)$$

$$tol_{relDiff} \sim tol_{absDiff} / \max(E(|o(t)'|), 1), \quad o(t_i)' = \frac{o(t_{i-1}) - o(t_{i+1})}{2 \cdot |t_{i+1} - t_{i-1}|}$$

wobei $E()$ den Erwartungswert, $\sigma()$ die Standardabweichung bezeichnet.

Das vorgeschlagene mehrstufige Vorgehen ermöglicht nicht nur die Erkennung zeitlich begrenzter Abweichungen, sondern auch einen adäquaten Vergleich zeitlich zueinander verschobener Signale. Hierdurch zeichnet es sich bereits gegenüber den im Abschnitt 2 diskutierten herkömmlichen Differenzbildungsverfahren und statistischen Vergleichsverfahren aus. Im Gegensatz zum Toleranzschlauchverfahren werden vertauschte Merkmalsreihenfolgen (wie im Beispiel Abb. 2) als signifikante Unterschiede identifiziert. Darüber hinaus eröffnet das Differenzmatrix-Verfahren weit über herkömmliche Vergleichsmethoden hinausgehende Analysemöglichkeiten: Systematische zeitliche Streckungen, Stauchungen und Verschiebungen, auch wenn sie nur in begrenzten Zeitbereichen auftreten, können identifiziert werden. Die Performanz des Differenzmatrix-Verfahrens ist linear in der Signallänge und hängt von der gewünschten Suchtiefe zur optimalen zeitlichen Anpassung der Signale ab. Sie ist mit der Performanz des Toleranzschlauchverfahrens vergleichbar.

MEval

Diese Algorithmen und Methoden und weitere wurden in MATLAB implementiert und exploriert.

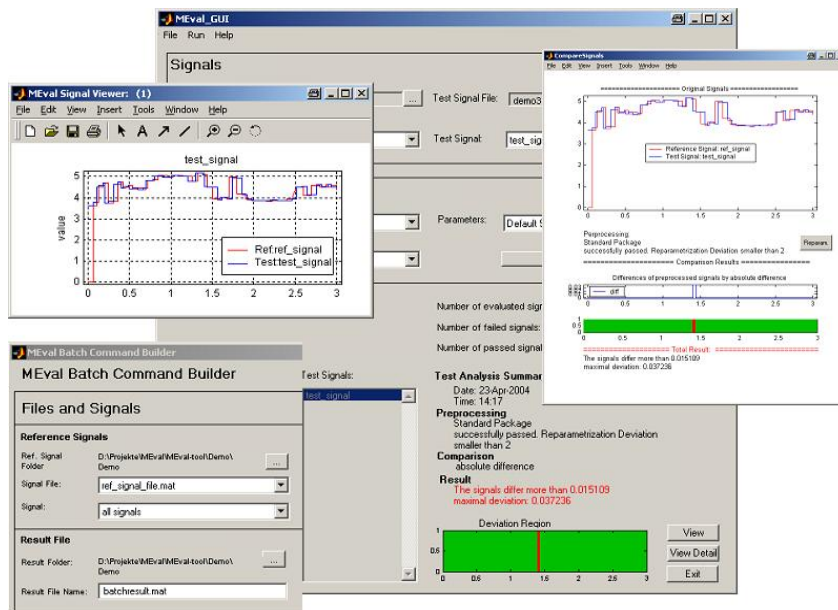


Abbildung 4 Implementierung der Algorithmen in MEval

Der Ansatz zur Ableitung von Default-Toleranzschwellwerten aus einer statistischen Analyse der Referenzsignale wurde anhand zahlreicher Signalpaare in Pilotstudien bestätigt. Auch die Vertrauenswürdigkeit der Algorithmen zur automatisierten Testauswertung konnte eindrucksvoll nachgewiesen werden. Dazu wurden Modelltestdaten aus verschiedenen Projekten (Motorsteuerung, Fahrdynamik etc.) eines großen namhaften Automobilkonzerns u.a. verglichen mit den Testausgaben des generierten Codes. Die automatische Bewertung wurde anschließend den manuellen Beurteilungen gegenüber gestellt. Das Verfahren wurde bei üblichen Signallängen von ~ 2.000 Abtastpunkten online durchgeführt, bei weit umfangreicheren Sequenzen auch im Batch-Modus. Durch die vorangestellte Partitionierung in Intervalle signifikanter Abweichungen konnten vielfach auch Signale mit mehr als 20.000 Abtastpunkten und wenigen zeitbeschränkten Abweichungen in kurzer Zeit analysiert werden.

Einsatz und Möglichkeiten, Grenzen und bisherige Erfahrungen

Die in diesem Beitrag vorgestellten Vergleichsverfahren sind prinzipiell auf beliebige Signalpaare – unabhängig von Ihrer Herkunft – anwendbar. Im Rahmen eines effizienten modellbasierten Entwicklungsprozesses sollten sie jedoch gezielt zum Einsatz kommen. Hauptanwendungsgebiet sind hier Regressionstests. Aufeinanderfolgende Modellversionen müssen auf ihre Konsistenz und korrekte Weiterentwicklung überprüft werden, Fehlerbehebungen und Modelloptimierungen auf ihre funktionale Äquivalenz. Werkzeuggestützte Testdatenverwaltung und automatische Testdurchführung mit anschließender Testauswertung sichern die Entwicklung hier effektiv ab [4].

Aufgrund der komplexeren Vergleichsmöglichkeiten zeitabhängiger Testdaten sollten allerdings die Verfahren und Parameter einer automatisierten Testauswertung projektweit und anwendungsspezifisch von einem Experten definiert werden. Anwendungsexperten, zusammen mit Fachleuten der Qualitätssicherung sollten dann spezifisch für den Anwendungsbereich und die Entwicklungsphase die jeweiligen Kriterien der Tests festlegen. Diese Festlegung entspricht dem Arbitr, Schiedsrichter, im Testing Profile der UML. Der Tester würde die abschließende Begutachtung durchführen, validation action, und könnte

sich ansonsten auf seine eigentliche Kompetenz konzentrieren, der Definition der wichtigen und nötigen Testfälle.

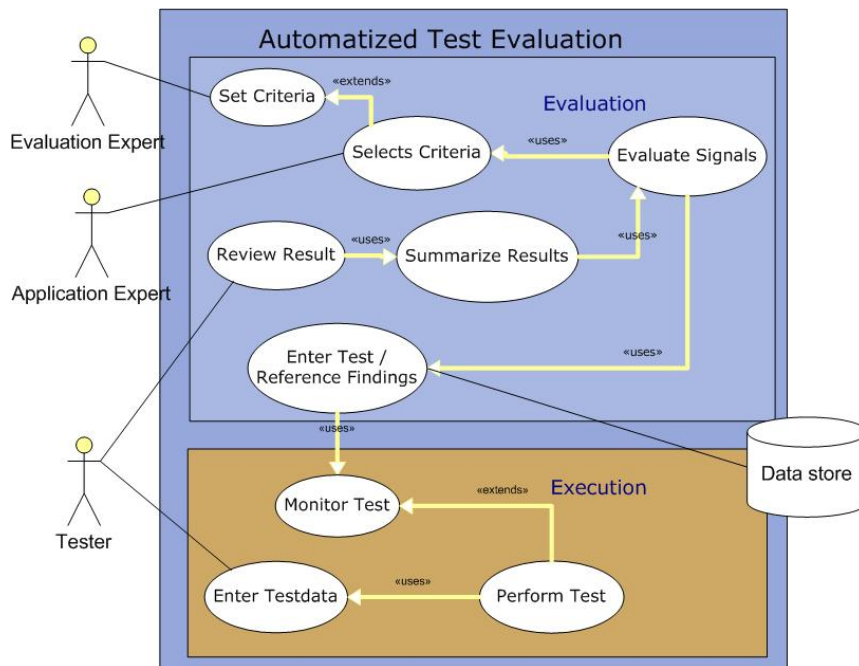


Abbildung 5 Einsatz einer automatischen Testauswertung

Die Pilotstudien haben gezeigt, dass eine Testauswertung ein Vielfaches an Zeit erspart. So konnten Hunderte von Testdaten in einem Batch über Nacht ausgewertet und anderntags begutachtet wurden, eine Aufgabe, die ansonsten bei sorgfältiger Ausführung Tage benötigt hätte. Auch die Qualität der Auswertung war bemerkenswert. So wurde jeder Fehler, den eine manuelle Durchsicht entdeckt hatte, gefunden und zusätzlich einige weitere, die in der Vielzahl übersehen worden waren.

Diese Ergebnisse bestätigten den konservativen Ansatz der Algorithmen und des Tools *MEval*, eher mehr Fehler anzumahnen, als voreilig abweichende Signale als gleich zu markieren, die bei einer manuellen Überprüfung abgelehnt worden wären.

Automatische Testauswertung ist allerdings auch kein Allheilmittel. Fehler sollten und werden als Fehler erkannt, doch wird jeder automatische Vergleich immer wieder auch Signale als verschieden klassifizieren, die ein erfahrener Tester berechtigt als identisch einstufte. Wollte man diese falsche, systemseitige Einstufung beheben, so wären die gesetzten Schwellwerte und Parameter freier zu setzen. Dies birgt jedoch wiederum die Gefahr, dann fehlerhafte Signalpaare nicht als solche zu erkennen. In der Werkzeugumsetzung wurde deshalb ein konservativer Weg beschritten, eher zu viele Signalpaare als verschieden zu klassifizieren, denn die Gleichheitskriterien all zu sehr verwässern.

Referenzen

- [1] Elektronik Automotive, 3/2003, Elektrik-/Elektronik-Strategie wird neu ausgerichtet
- [2] Model-Based Tools Update. The Hansen Report on Automotive Electronic, Vol. 14, No. 5, www.hansenreport.com, 2001.

- [3] "Tools survey 2001/2" in Professional Tester Volume 3 No. 3 2002
- [4] Mirko Conrad, Ines Fey, Hartmut Pohlheim: Automatisierung der Testauswertung für Steuergerätesoftware, Baden-Baden Konferenz
- [5] Hans-Werner Wiesbrock, Mirko Conrad, Ines Fey, Hartmut Pohlheim: Ein neues automatisiertes Auswerteverfahren für Regressions- und Back-to-Back-Tests eingebetteter Regelsysteme, Softwaretechnik-Trends, Band 22, Heft 3, (ISSN 0720-8928), September 2002