

ZAMOMO

Verzahnung von modellbasierter Softwareentwicklung und modellbasiertem Reglerentwurf

Stefan Kowalewski, Jacob Palczynski,
Andreas Polzer
Lehrstuhl für Informatik 11
RWTH Aachen
Ahronstraße 55
52074 Aachen

Thomas Rose, Dominik Schmitz
Fraunhofer Institut für Angewandte
Informationstechnik
Schloss Birlinghoven
53754 Sankt Augustin

Harald Nonn
AVL Deutschland GmbH
Peter-Sander-Str. 32
55252 Mainz-Kastel

Dirk Abel, Ralf Beck, Kai
Hoffmann, Frank Heßeler
Institut für Regelungstechnik
RWTH Aachen
Steinbachstr. 54
52074 Aachen

Martin Düsterhöft,
Michael Reke
VEMAC GmbH & Co. KG
Kranzstr. 7
52070 Aachen

Kurzfassung

Bei der Entwicklung softwareintensiver eingebetteter Systeme in regelungstechnischen Anwendungen wie z.B. Motorsteuergeräten treffen zwei Disziplinen aufeinander, die sich in der Vergangenheit weitgehend getrennt von einander entwickelt haben: die Softwaretechnik und die Regelungstechnik. Das Zusammenwirken der beiden Disziplinen läuft in der Praxis nicht reibungslos: Unterschiedliche Begriffswelten und Perspektiven auf den Entwurfsgegenstand verursachen Missverständnisse. Die fehlende Abstimmung im Entwurfsvorgehen verhindert die durchaus mögliche gegenseitige methodische Ergänzung. Im Ergebnis wird erhebliches Potenzial zur effizienteren Erzielung von Qualität, insbesondere Sicherheits- und Zuverlässigkeitseigenschaften von Regelungssoftware verschenkt.

Das Ziel des Projekts ist, diese Situation durch eine geeignete Integration von softwaretechnischem und regelungstechnischem Entwurf zu verbessern. Insbesondere sollen die in beiden Disziplinen entstandenen modellbasierten Ansätze miteinander verzahnt und ergänzt werden, so dass eine durchgängige Entwurfsmethodik von der Anforderungsanalyse für ganze Produktlinien von softwarebasierten Regelsystemen bis hin zum Feintuning einzelner Regelfunktionen unterstützt wird. Als exemplarische Anwendungsdomäne werden Motorsteuerungen untersucht. Unmittelbar verwertbare Ergebnisse des Projekts werden Methoden- und Prozessdefinitionen mit dazugehörigen Softwarewerkzeugen und einer Anwendungsleitlinie sein.

1. Einleitung und Vorstellung des Themenkomplexes

In der Diskussion über Entwurfsmethodiken für softwareintensive eingebettete Systeme hat sich in Forschung und Praxis die Ansicht durchgesetzt, dass ein „modellbasierter“ Entwurf der geeignete Ansatz ist, um den aktuellen Herausforderungen zu begegnen. Fragt man bei Vertretern der unterschiedlichen am Entwurf beteiligten Disziplinen nach, so muss man feststellen, dass der Begriff „modellbasiert“ mit sehr unterschiedlichen Bedeutungen gebraucht wird. So verstehen die Softwaretechniker unter einer modellbasierten Entwicklung – grob gesagt – den Einsatz von rechen-technisch weiterverarbeitbaren (z.B. ausführbaren, transformierbaren oder verfeinerbaren) Beschreibungen von Anforderungen oder Entwurfsergebnissen. Charakteristika eines modellbasierten Softwareentwurfs sind der frühzeitige Einsatz solcher Modelle im Entwicklungsprozess zur besseren Analyse der Anforderungen und die Verwendung automatischer Modelltransformationen, um von groben Modellen zu feineren und schließlich zum ausführbaren Code zu kommen. Ein tragfähiger modellbasierter Softwareentwurf ist unter anderem dadurch gekennzeichnet, dass die Modelle neben den funktionalen Anforderungen auch Qualitätseigenschaften des zu entwickelnden Systems berücksichtigen und ihre Erfüllung überprüfbar machen.

Sprechen dagegen Regelungstechniker von modellbasiertem Reglerentwurf, so ist die eigentliche Bedeutung die, dass ein Regler auf der Basis eines mathematischen Modells der Regelstrecke und Anforderungen an den geschlossenen Regelkreis entworfen wird, sei es mit Hilfe von Simulationen oder direkt durch automatische Syntheseverfahren. Ein wesentlicher Unterschied im Vergleich zum modellbasierten Softwareentwurf ist also, dass der Modellierungsgegenstand des Ausgangsmodells (Strecke und Anforderungen an den geschlossenen Kreis) und des in diesem Schritt entstehenden Modells (Regler) verschieden sind. Ein weiterer Unterschied besteht darin, dass die zu erzielenden Anforderungen (dynamisches Verhalten des geschlossenen Kreises) aus softwaretechnischer Sicht zu den funktionalen Anforderungen gehören und nicht-funktionale Anforderungen beim Reglerentwurf normalerweise nicht berücksichtigt werden.

Da es heute Stand der Technik ist, Modelle von Reglern (im weitesten Sinne) mit Werkzeugen wie Matlab/Simulink und Targetlink automatisch in ausführbaren Code zu transformieren – also ein Konzept des modellbasierten Softwareentwurfs bei der Entwicklung von Regelungssystemen einzusetzen, geraten die Begriffe häufig durcheinander. Das damit verbundene mangelnde gegenseitige Verständnis der beiden Disziplinen Softwaretechnik und Regelungstechnik ist ein Grund dafür, dass die Zusammenarbeit beim Entwurf softwareintensiver eingebetteter Systeme von Reibungsverlusten geprägt ist. Die in der Regelungstechnik verwendeten Modellformen passen nicht zu den Ansätzen aus dem modellbasierten Software-Entwurf. Diese wiederum eignen sich nicht zum regelungstechnischen Entwurfsvorgehen, da das Paradigma plattformunabhängig / plattformabhängig die unterschiedlichen Modellierungsgegenstände Strecke, Regler und Regelkreis nicht unmittelbar abbilden kann.

Auf der anderen Seite besteht aber durchaus Potenzial für eine gegenseitige Befruchtung der Entwurfsmethodiken durch Konzepte der jeweils anderen Seite. So könnte der regelungstechnische Entwurfsprozess dadurch gewinnen, dass Streckenmodelle in verschiedenen Verfeinerungsstufen verwendet werden und die jeweiligen Verfeinerungsschritte durch Transformationsverfahren (zumindest teilweise) automatisiert werden.

Auch die unterschiedlichen Stufen der Reglermodelle könnten als Verfeinerung aufgefasst werden. Ein Beispiel ist der Schritt der Kalibrierung von Motorsteuerungen, der bisher manuell abläuft, idealerweise aber eine (auf der Basis eines detaillierten Motor-, also Streckenmodells) automatisch ablaufende Verfeinerung sein sollte. Schließlich ist die Berücksichtigung

von nicht-funktionalen Anforderungen zu nennen, die in den regelungstechnischen Entwurf integriert werden muss, um den aktuellen Anforderungen an Variantenvielfalt, Wartbarkeit usw. Rechnung zu tragen.

1.1 Abgrenzung

Das BMBF-Projekt ZAMOMO beschäftigt sich in der Hauptsache, mit den unterschiedlichen Entwicklungsprozessen in der Regelungs- und Softwaretechnik. Dabei soll die Zusammenarbeit dieser beiden Disziplinen verbessert werden. Hauptaugenmerk liegt dabei auf dem gemeinsamen Entwicklungsprozess und der Berücksichtigung von Anforderungen zu einem frühen Zeitpunkt.

Als verwandte Förderprojekte sind vor allem IMMOS und AutoMODE zu nennen, die beide im IT2006-Rahmen gefördert werden. IMMOS hat das Ziel einer Integration der verschiedenen Schritte des modellbasierten Software- und Systementwurfs mit besonderem Schwerpunkt auf dem Test. Eine explizite Berücksichtigung von regelungstechnischen Anforderungen wird von Projekt und Konsortium nicht abgedeckt. ZAMOMO ergänzt daher den IMMOS-Ansatz, da einer vertikalen Integration entlang des V-Modells in IMMOS hier eine horizontale Integration über die Modellierungsgegenstände aus unterschiedlichen Disziplinen zugefügt wird.

Auch AutoMODE hat eine durchgängig modellbasierte Entwicklung eingebetteter Systeme im Automobil zum Ziel. Die Schwerpunkte liegen auf Architekturbeschreibungen und der formalen Definition von Entwicklungsschritten. Regelungstechnische Anforderungen oder die Integration regelungstechnischer Entwurfsmethodiken werden nicht betrachtet. ZAMOMO ergänzt also auch hier diesen Aspekt.

1.2 Ziele

Das Ziel dieses Projektes ist die Auslotung und Ausschöpfung des Potenzials, das sich durch eine Verzahnung des modellbasierten Softwareentwurfs mit dem modellbasierten Reglerentwurf ergibt. Dies beinhaltet neben der oben beschriebenen Übertragung von Konzepten auch eine Synchronisierung und Integration beider Pfade, um die Modelle konsistent zu halten. Gelingt dies, so kann der Entwurf softwareintensiver eingebetteter Systeme mit starker Regelungsfunktionalität deutlich effizienter gestaltet werden, wodurch aufgrund der erhöhten Durchgängigkeit bis in den Test ein Gewinn an Sicherheit und Zuverlässigkeit durch hohen Reifegrad erzielt wird.

Die konsistente Modellbildung sowohl auf der Seite der Regelungstechnik als auch im Bereich des Softwareentwurfs ermöglicht zum Beispiel eine genauere Vorhersagbarkeit der benötigten Hardwareperformance und liefert somit eine deutlich höhere Sicherheit zu Beginn einer Produktplanung. Ebenso liefert die durchgängig verzahnte Modellbildung eine sichere Beurteilung von vorhandenen oder zu schaffende Ressourcen bei nötigen Produktänderungen bzw. Produkthanpassungen.

Die Vorteile des verzahnten, modellbasierten Ansatzes in der Softwareentwicklung ermöglichen in jeder Phase der Produktentwicklung den vollständigen Überblick des noch zu erbringenden Entwicklungsaufwands. Nach der erfolgreichen Produktentwicklung ermöglicht die verzahnte Softwareentwicklung eine deutlich bessere Wartbarkeit und Anpassbarkeit an neue Produkthanforderungen. Aufgrund der vollständigen Transparenz kann die Wirkung einer Änderung in der Software direkt erfasst und getestet werden. So kann ein deutlicher Zeitgewinn in der Weiterentwicklung bzw. Adaption sowie eine Steigerung der Entwicklungssicherheit durch die weitere Nutzung erprobter Module erzielt werden. Als weiteren nicht zu unterschät-

zenden Vorteil liefert eine vollständige Verzahnung zwischen Softwaretechnik und Regelungstechnik erst die Basis für eine wirkliche Softwarezertifizierung.

Abbildung 1 versucht, den beschriebenen Ansatz mit dem in der Automobilindustrie etablierten V-Modell für die Systementwicklung zu illustrieren. Kerngedanke ist, von Anfang an und durchgehend im Entwicklungsprozess nicht nur modellbasiert vorzugehen, sondern auch die regelungstechnischen und softwaretechnischen Anforderungen gemeinsam zu behandeln. Alle Entwurfs- und Validierungsschritte betrachten beide Aspekte und können sich auf eine gemeinsame, konsistente Modellbasis abstützen. Die Abbildung stellt auch dar, an welchen Stellen des so gestalteten Entwicklungsprozesses die einzelnen Arbeitspakete des Projekts Ihren Schwerpunkt haben.

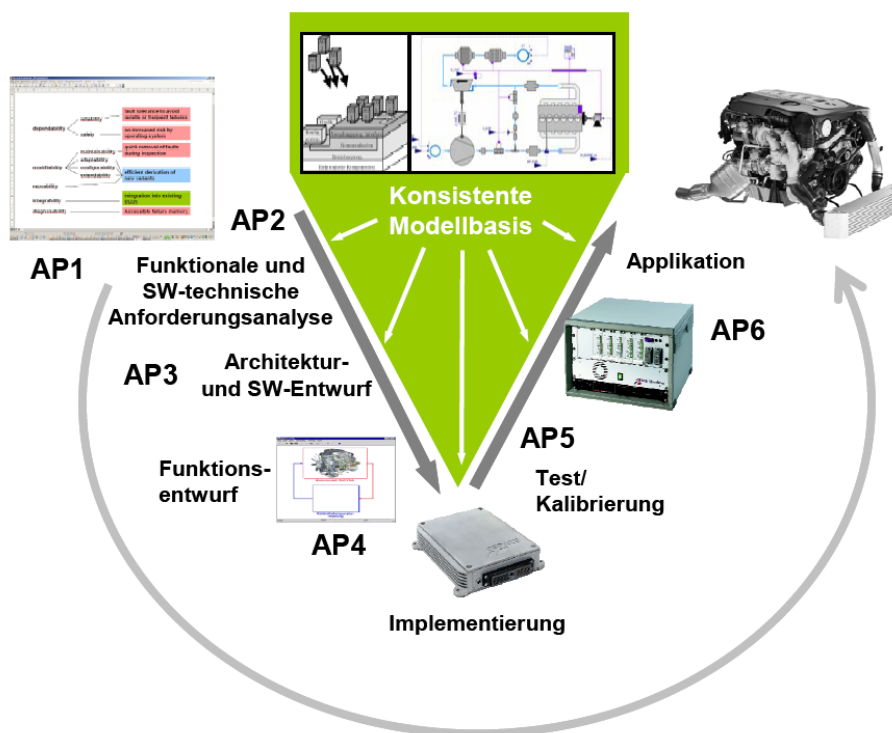


Abbildung 1: Positionierung der Arbeitspakete im V-Modell

Der aktuelle Entwicklungsprozess ist von den funktionellen Anforderungen getrieben. Regler werden modellbasiert entworfen, ohne dass nicht-funktionale Anforderungen berücksichtigt werden. Die Reglermodelle werden simulativ validiert und kalibriert. Die Arbeitspakete decken alle Aspekte und Phasen dieses Prozesses ab. Durch sie wird ein durchgängiger, modellbasierter und werkzeugunterstützter Entwurf von der frühen Phase der Anforderungsanalyse bis hin zur automatischen Kalibrierung realisiert. Dabei werden sowohl software- als auch regelungstechnische Sichtweisen beachtet.

2. Projektstatus

2.1 Vorstellung der Arbeitspakete

Die einzelnen Projektpartner haben sich bezüglich der Aufgabenbereiche und der Möglichkeiten der Zusammenarbeit abgestimmt. Die Ergebnisse dieses Abstimmungsprozesses wurden in Lastenheften festgeschrieben und werden im Folgenden vorgestellt.

Arbeitspaket 1 – Integration von regelungstechnischen Modellen in die Software-Modellierung

Das Arbeitspaket 1 befasst sich mit der Integration von Strecken- und Regelkreismodellen in die Software-Modellierung zur frühzeitigen Berücksichtigung von Anforderungen an die Regelfunktionen bei der Software-Anforderungsanalyse (inkl. Variabilität und Zuverlässigkeit).

Die aus der Regelungstechnik stammenden Entwurfsmethoden sind funktional orientiert. Das heißt, sie unterstützen den Entwickler nur darin, die funktionalen Anforderungen an den Regelkreis (Regelgeschwindigkeit, Überschwingverhalten, Robustheit, Stabilität etc.) zu erfüllen. Die Anforderungen an softwarebasierte Regelungssysteme gehen aber weiter und umfassen auch nicht-funktionale Anforderungen wie Zuverlässigkeit, Sicherheit, Wartbarkeit, Modifizierbarkeit, Variabilität, etc. Häufig sind diese Anforderungen auch schwieriger zu erfüllen und entscheidend für den Geschäftserfolg.

Ziel dieses Arbeitspaketes ist es, die systematische, gleichzeitige Berücksichtigung funktionaler und nicht-funktionaler Anforderungen beim Entwurf eingebetteter Systeme mit Regelfunktionen zu ermöglichen.

Die modellbasierte Softwarebeschreibung ist vom Ansatz her geeignet, diese Ziele zu erreichen, da sie durch Architekturmodelle und Architekturanalysemethoden eine Berücksichtigung von funktionalen und nicht-funktionalen Anforderungen leistet. In diesem Arbeitspaket sollen die nicht-funktionalen Aspekte definiert und an die funktionalen Aspekte der Regelssoftware angekoppelt werden. Dazu müssen Modellierungsmöglichkeiten für das physikalische Streckenverhalten, die Modellbeschreibung des Reglers mit der Softwaremodellierungstechnik so gekoppelt werden, dass eine gemeinsame Betrachtung und Analyse möglich wird.

Die grundsätzliche Idee, den kompletten Entwicklungsprozess der Motorsteuerungssoftware nur noch modellbasiert umzusetzen, soll hier realisiert werden. Zu diesem Zweck wird dem Entwickler ein Modellierungswerkzeug zur Verfügung gestellt. Die Modelle werden aus Komponenten zusammengestellt, die in einer Bibliothek zur Verfügung stehen: Durch Konfiguration, Kombination und Verbindung der Komponenten wird es dann ermöglicht, ein Modell der Motorsteuerung zu entwickeln. Die Auswahl der Komponenten wird durch die Anforderungen an den Einsatzfall und die Ausrüstung des Motors vorgegeben.

Nach der Erstellung des Modells soll dieses mittels weiterer Software, die in die Modellierungsumgebung eingebunden ist, simuliert werden. Für die Simulation wird ein weiteres Modell benötigt, welches den Motor darstellt. Dieses Modell soll wiederum aus Standardkomponenten bestehen und damit leicht zu erstellen sein. Mit dieser Simulation sollen dynamische Aspekte wie Leistung und Verbrauch vorhergesagt werden und die Möglichkeit zur Optimierung der Motorsteuerung gegeben werden.

Abschließend soll automatisch aus dem Softwaremodell die Software für ein Motorsteuerggerät generiert werden. Wichtig ist, dass auch der generierte Quelltext sowohl den funktionalen also auch den nicht-funktionalen Anforderungen genügt.

Arbeitspaket 2 – Parallele Verfeinerung von Software- und Regelstreckenmodellen

In der Praxis sieht der Entwurfsablauf für softwarebasierte Regelungssysteme häufig so aus, dass zunächst grobe Kunden- oder Marketinganforderungen analysiert werden und dann auf der Basis der vorherigen Generation Architekturentscheidungen getroffen werden. In dieser Phase werden zwar schon erste Verhaltensmodelle erstellt, eine Überprüfung durch Ausführung mit einem Streckenmodell findet aber nicht statt.

Dies wird erst beim Entwurf oder Tuning des eigentlichen Regelalgorithmus auf einem wesentlich niedrigeren Abstraktionsniveau durchgeführt. Dies hat den Nachteil, dass die dann gefundenen Fehler nur mit hohem Aufwand wieder behoben werden können.

In Arbeitspaket 2 soll eine Möglichkeit geschaffen werden, Softwaremodelle frühzeitig gegen ein geeignetes Streckenmodell durch Simulation zu überprüfen. Dafür müssen die Streckenmodelle analog zu den Softwaremodellen verfeinert werden können, wobei die Konsistenz über die Verfeinerungsschritte vom groben zum feinen Modell sichergestellt werden muss.

Einer der ersten Schritte ist festzustellen, welche gemeinsamen Abstraktionsebenen bei Software- und Streckenmodell existieren. Zudem ist die Frage zu klären, wodurch sich diese Ebenen voneinander unterscheiden. Zum einen kommt hier die mathematische Komplexität (endliche Automaten, affine Gleichungen, nicht-lineare Differentialgleichungen,...) in Frage. Die Abstraktion kann aber auch in einer anderen Dimension erfolgen. Der Detaillierungsgrad der Streckenbeschreibung selbst kann variieren, je mehr Details der Strecke bekannt sind, desto genauer kann das Reglersoftwaremodell gestaltet werden.

Wenn die Abstraktionsebenen ermittelt wurden, wird ein Konzept entwickelt, mit dem die Strecken konsistent zwischen verschiedenen Ebenen transformiert werden können. Dabei müssen unter anderem die Schnittstellen zum jeweiligen Reglermodell vorhanden sein.

Ein weiterer Aspekt, der durch die Dimension der Details eingeführt wird, ist der mögliche modulare Aufbau der Streckenmodelle. Ist es möglich, die Modelle zu modularisieren, können neue Modelle in späteren Entwicklungsprozessen leichter erstellt werden (siehe Arbeitspaket 5).

Hersteller von Reglerkleinserien sammeln über die Zeit umfangreiches Wissen über Strecken aus verschiedenen Kundenaufträgen und Anforderungen an. Dieses Wissen soll für Arbeitspaket 2 systematisch erfasst und nutzbar gemacht werden.

Arbeitspaket 3 – Anwendung von ziel-orientierter Softwareentwicklung auf Regelungsaufgaben

Das Arbeitspaket 3 befasst sich mit der Anwendung der ziel- und agenten-orientierten Softwareentwicklung TROPOS auf Regelungsaufgaben. Komplexe Regelungsfunktionen sind charakterisiert durch eine Vielzahl interagierender Elemente. TROPOS erlaubt Ziele und wechselseitige Abhängigkeiten derartiger Akteure abstrakt zu spezifizieren und damit auch sehr unterschiedliche Lösungsvarianten im modellbasierten Entwurf auf diese Ziele rückzubeziehen.

Die Herausforderung besteht nun darin, diese für die reine Softwareentwicklung entwickelte Methodologie um die Anwendbarkeit auf den regelungstechnischen Entwurf zu erweitern. Der Schwerpunkt liegt dabei auf der Anforderungserfassung. So müssen etwa die Elemente

der Modellierungssprache auf ihre Eignung hin untersucht werden und ggf. neue Elemente geschaffen werden. Die im Reglerentwurf bisher nicht explizit betrachteten nicht-funktionalen Anforderungen stellen dabei kein zusätzliches Problem dar, sondern werden höchstwahrscheinlich bereits durch das Konzept des „Softgoals“ in TROPOS abgedeckt.

Letztendliches Ziel dieses Arbeitspaketes ist die gemeinsame Behandlung der software-technischen und regelungs-technischen (nicht-funktionalen) Anforderungen zusammen mit einer geeigneten Werkzeugunterstützung und einer zugehörigen Evaluation. Wesentlich ist der Erhalt der Nachvollziehbarkeit: Der evolutionäre Architektur- und Reglerentwurf muss immer auf die Anforderungen rückbezogen werden können, um bei Designentscheidungen die richtige Auswahl treffen zu können.

Als nächstes steht die Analyse der notwendigen methodischen Anpassungen des TROPOS-Rahmenwerks auf regelungstechnische Aufgabenstellungen an. Dazu wird gemeinsam mit den Projektpartnern der bisherige Reglerentwicklungsprozess analysiert, um geeignete Anknüpfungspunkte für eine konzeptuelle Modellierung der (regelungstechnischen) Anforderungen zu finden.

Arbeitspaket 4 – Entwurf von Regelungsfunktionen unter Berücksichtigung nichtfunktionaler Anforderungen

Fortschritte in der Theorie der Regelungstechnik und die stetig steigende Rechenleistung wirtschaftlich einsetzbarer Prozessoren führen zu einem Wandel sowohl im Entwicklungs- als auch im Auslegungsprozess regelungstechnischer Systeme. In klassischen Ansätzen wird das Prozessverhalten lediglich (vorrangig experimentell) untersucht, um daraus nach häufig heuristischen Regeln die Parameter für einfache Regelalgorithmen abzuleiten. Moderne Verfahren zeichnen sich dagegen z.B. dadurch aus, dass das Wissen über die Prozesseigenschaften in unterschiedlichen Formen unmittelbar im Regler genutzt wird.

Ein am Institut für Regelungstechnik vielfach eingesetztes Verfahren, speziell im Bereich der Motorregelung, ist das der modellbasierten prädiktiven Regelung. Hierbei wird ein explizites mathematisches Modell der zu regelnden Strecke verwendet, um das zukünftige Prozessverhalten bei Vorgabe aktueller Stellgrößen abzuschätzen und somit durch z.B. Minimierung der zukünftigen Regelabweichung optimale Stellgrößen zu bestimmen. Hierdurch kann das Verfahren u.a. auf nichtlineare, gekoppelte Mehrgrößenprozesse eingesetzt werden und stellt somit ein skalierbares Verfahren dar.

Der Entwurf der Algorithmen erfolgt – ausgehend von einfachen Modellen für Strukturüberlegungen – meist anhand eines komplexen Simulationsmodells der Regelstrecke in einer „Model-in-the-loop“-Simulation (MIL-Simulation) auf einem PC in einer integrierten Entwicklungsumgebung wie Matlab/Simulink. Hierbei wird für eine erste Validierung in der Entwurfsumgebung idealerweise das gleiche Streckenmodell verwendet, welches abschließend zur Verifikation der im Steuergerät implementierten Funktion in der HIL-Simulation (Hardware-in-the-loop) Anwendung findet.

Im Rahmen des Arbeitspakets werden eine skalierbare Reglerfunktion und eine modellbasierte OBD-Funktion (On-Board Diagnose) für eine Motorsteuerung entwickelt. Hierbei wird zunächst herausgearbeitet, welche Spezifikationen für einen groben Entwurf benötigt werden, welche Skalierungsstufen (z.B. in Abhängigkeit möglicher Varianten des Motors) existieren und welche weiteren nicht-funktionalen Anforderungen wie Wartbarkeit, Parametrierbarkeit, Integrierbarkeit mit weiteren Funktionen etc. im Entwurf zu berücksichtigen sind, um die Wiederverwendbarkeit der Reglerfunktionen zu erhöhen und somit über einen höheren Reifegrad letztendlich einen Gewinn an Sicherheit und Zuverlässigkeit zu erzielen.

Arbeitspaket 5 – Verbesserung von modularen, echtzeitfähigen HIL-Modellen

Für die aktuellen Aufgaben eines HIL-Prüfstandes im Motorenumfeld bieten die heute eingesetzten Plattformen gute Möglichkeiten, ein echtzeitfähiges Modell, das in einer gängigen Modellierungssprache formuliert ist, auszuführen. Allerdings bieten die heutigen HIL-Prüfstände nur unzureichende Möglichkeiten, Modelle zu modularisieren.

Ein Modellmodul besteht immer aus einem echtzeitfähigen und einem nicht-echtzeitfähigen Teil. Zum nicht-echtzeitfähigen Anteil zählen Funktionen wie Parameter-Preprocessing, Parameterauswahl und Bedienung/Visualisierung. Damit die Elektronik von offline- bis Echtzeitsimulation am HIL-Prüfstand durchgängig entwickelt werden kann, müssen geeignete Modellmodule zur Verfügung stehen.

Für die Erstellung von echtzeitfähigen Modellen ist sehr spezifisches Wissen erforderlich. Modellanbieter sind in der Regel nicht in der Lage, für alle Komponenten eines Motors Module in den erforderlichen Detaillierungsstufen zu erstellen. Zukünftige HIL-Systeme müssen daher eine Plattform bereitstellen, die es unterschiedlichen Modellanbietern erlaubt, ihre Module zu integrieren. Idealerweise sollten die Komponentenhersteller auch geeignete echtzeitfähige Modellmodule der Komponenten liefern. Die HIL-Plattform muss dabei gewährleisten, dass entsprechende Verarbeitungsfunktionen für Parameter eingebracht werden können, und die Echtzeiteile müssen so ausgeführt werden, dass sie sich in ihrem Laufzeitverhalten nicht beeinflussen.

Anzustreben ist in diesem Zusammenhang zudem, dass der Prozessschritt „Parametrierung“ sich nicht nur auf Messdaten des Motorprüfstandes abstützt, sondern alternativ auch auf Daten einer deutlich früher verfügbaren Motorprozesssimulation (CFD – Computational Fluid Dynamics).

Diesen neuen Anforderungen an den modellbasierten Test wird im Arbeitspaket 5 durch die Entwicklung einer echtzeitfähigen Modellplattform erfüllt, auf der die Teilmodelle weitgehend durch eine Konfigurationsbeschreibung automatisch integriert werden. Dabei wird neben den Echtzeitanteilen auch die Parametrierung der Modelle berücksichtigt.

Arbeitspaket 6 – Automatische Kalibrierung

Ziel dieses Arbeitspakets ist es zu überprüfen, inwieweit Modellparameter und Regler durch automatische Kalibrierung optimiert werden können. Dieses wird am Beispiel der von AVL vertriebenen Optimierungssoftware CAMEO untersucht.

Die Applikation von Steuergeräten soll zukünftig durch eine durchgängige modellbasierte Entwicklung weiter automatisiert werden. Der Zeit- und Kostenaufwand kann gesenkt werden, indem eine Applikation bzw. Vorapplikation an einem HIL-Prüfstand mit einem sehr genauen Motormodell durchgeführt wird. Die Kostenersparnis wird durch eine weitgehende Automatisierung erreicht und zudem dadurch begünstigt, dass eine HIL/MIL-Prüfstandsstunde erheblich preiswerter ist, als eine Belegungsstunde an einem Motorenprüfstand oder in einem Fahrzeug-Prototyp.

Das heutige Vorgehen sieht vor, dass ein Applikateur einen Motorprüfstand und ein Kalibriersystem bedient. Er kann so für jeden Motorbetriebspunkt die optimalen Parameter für das Steuergerät ermitteln. Dieses Vorgehen ist in Abbildung 2 im linken Teil zu erkennen. Der geplante HIL/MIL-Ansatz ist im rechten Teil der Abbildung dargestellt. In diesem Szenario wird der Motorprüfstand durch einen HIL/MIL-Prüfstand mit einem echtzeitfähigen Motormodell ersetzt. Die Steuergeräteparameter werden durch die Optimierungssoftware kalibriert.

Bei der Applikation am Motorprüfstand wird der Applikateur durch eine leistungsfähige Optimierungssoftware unterstützt. Für die Applikation am HIL/MIL-Prüfstand ist jedoch das

sehr genaue Motormodell die Schlüsselkomponente. Um die Anzahl der Werkzeuge zu begrenzen, wird untersucht, inwieweit die bereits vorhandene Optimierungssoftware dazu genutzt werden kann, die Parameter eines genauen Motormodells auf Basis von Meß- oder Berechnungsdaten zu „optimieren“ und in weiterer Folge die optimalen Parameter von Motorreglern zu finden.

Arbeitspaket 7 – Integration

Die einzelnen Arbeitspakete können vom zeitlichen Ablauf her zunächst weitgehend parallel bearbeitet werden, da sie jeweils unterschiedliche Aspekte oder Phasen des in diesem Bereich meistens zugrunde gelegten V-Modells betreffen. In der letzten Projektphase werden die bis dahin entstandenen methodischen Ansätze, Prozesse und Werkzeuge zu einer zusammenhängenden Entwurfsmethodik integriert. Im Ergebnis liegt dann eine Methodenbeschreibung mit dazugehörigen Werkzeugen vor, die eine software- und regelungstechnisch verzahnte Entwicklung über die ganze Spannbreite des V-Modells anbietet.

An der abschließenden Integration der Ergebnisse in eine werkzeuggestützte Methodik mit Anwendungsleitfaden sind alle Partner beteiligt. Bestandteil des Leitfadens sind insbesondere auch die erarbeiteten konzeptuellen Modelle, welche der präzisen Dokumentation der Zusammenhänge und Abläufe dienen.

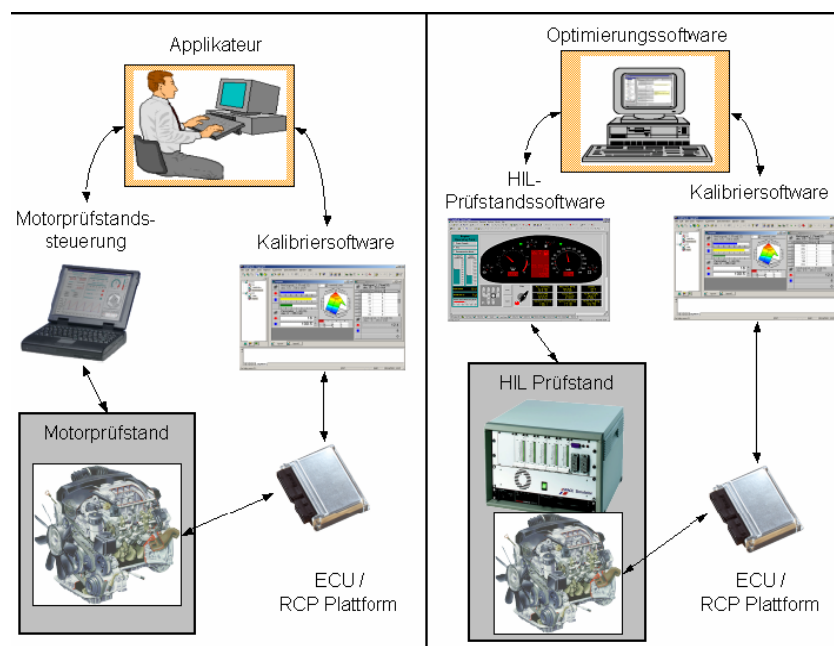


Abbildung 2: Vergleich zwischen dem herkömmlichen Vorgehen bei der Kalibrierung und dem Vorgehen HIL/MIL-Prüfstand

3. Erfahrungen, Bewertungen

Die Erfahrungen im ZAMOMO Projekt sind geprägt von unterschiedlichen Vorstellungen und Grundlagen der beteiligten Disziplinen. Dementsprechend sind die ersten Projektmonate seit Anfang des Jahres davon geprägt, eine gemeinsame Basis zu schaffen. Dazu gehört ein

gemeinsames Verständnis wichtiger Begriffe, im Rahmen von ZAMOMO betreffen diese vor allem funktionale und nicht-funktionale Anforderungen. Um das bisher unterschiedliche Verständnis dieser Begriffe projektweit zu vereinheitlichen, wurde ein gemeinsames Glossar angelegt, das kontinuierlich erweitert werden soll.

Die Anforderungsanalyse-Phase der TROPOS-Methodologie wurde erstmals auf Anwendungsfälle im Umfeld der Reglerentwicklung angewendet. Eine grundsätzliche Eignung konnte festgestellt werden. Parallel erfolgt die Evaluation verschiedener TROPOS-Tools (OME, OpenOME, TAOM4E) im Hinblick auf ihre Eignung als Basis für das zu entwickelnde Softwarewerkzeug. Diese Untersuchung ist noch nicht abgeschlossen. Parallel zur Anwendung des TROPOS Rahmenwerks wurde im selben Kontext eine szenarienbasierte Architekturanalyse durchgeführt. Die Requirement-Erfassung und -Analyse im regelungstechnischen Bereich zeigt, dass beide Verfahren unterschiedliche Schwächen und Stärken haben. Eine weitere Evaluation der Verfahren wird im Bereich des Arbeitspakets 1 weiter fortgeführt.

Bei der Analyse funktionaler und nichtfunktionaler Anforderungen an den Entwicklungsprozess eines Reglers und an den Regler hat sich gezeigt, dass nichtfunktionale Anforderungen bisher nicht explizit betrachtet wurden. Einige aus der Softwareentwicklung bekannten nichtfunktionale Anforderungen haben keine Bedeutung für die im Projekt betrachteten Regler. Aber in Fällen, wie zum Beispiel bei Skalierbarkeit, Wartbarkeit und Performanz besteht Interesse von Seiten der Regelungstechnik zu untersuchen, inwieweit sich solche Eigenschaften in Regler und Reglerentwicklung integrieren lassen.

Mehr Zeit als geplant musste bisher für die noch anstehende Akquise eines Testmotors, der zur Validierung der in Arbeitspaket 4 zu erstellenden Regler und zur Modellbildung benötigt wird, aufgewendet werden. Hindernisse bestehen hinsichtlich der Zurückhaltung der Motorenhersteller, Teile der Motorsteuerung über einen Freischnitt zugänglich zu machen. Ein Freischnitt ermöglicht es, auf bestimmte Daten innerhalb des Motorsteuergeräts zuzugreifen und zu manipulieren. Die Projektpartner stehen deshalb im Moment in Kontakt mit entsprechenden Firmen, um einen Motor zu erhalten, der die im Projekt benötigten Eigenschaften besitzt.

4. Ausblick

Nachdem die Koordinierungsphase zwischen den Projektpartnern abgeschlossen ist, hat die inhaltliche Arbeit begonnen. In allen Arbeitspaketen werden die bestehenden Situationen analysiert und die Anforderungen bezüglich des Projektziels formuliert.

4.1 Nächste Schritte

In Arbeitspaket 1 wird die Architektur der Modellierungs- und Simulationssoftware entworfen und diese danach implementiert. Nachdem in Arbeitspaket 2 die möglichen gemeinsamen Abstraktionsebenen von Strecke und Software ermittelt worden sind, wird ein Konzept für die Streckenmodellhierarchie erarbeitet. Ziele und mögliche Agenten in der Reglersoftwareentwicklung werden in Arbeitspaket 3 identifiziert, mit diesem Wissen kann TROPOS an den Reglerentwurfprozess angepasst werden. In Arbeitspaket 4 werden nicht-funktionale Anforderungen an den Entwurfsprozess und an die Reglersoftware ermittelt und formuliert. Aufgrund der noch laufenden Akquise eines Motors wird in Arbeitspaket 5 aktuell eine Bibliothek von Motormodellmodulen erstellt. In Arbeitspaket 6 wird CAMEO so angepasst, dass es verwendet werden kann, um MIL-/HIL-Reglermodelle automatisch zu kalibrieren.

4.2 Zusammenfassung

Die ZAMOMO-Partner stammen aus unterschiedlichen Institutionen und Disziplinen. Die Zusammenarbeit von Informatikern und Regelungstechnikern, von universitären Forschern und Mitarbeitern von Wirtschaftsunternehmen werden neue Ansätze im Entwurf und der Realisierung von Regelungssoftware ergeben. Durch diese Ansätze wird Software vom Reglerentwurf bis zur Applikation auf ein Steuergerät durchgängig entwickelt werden. Die Werkzeuge und Methoden schaffen eine gemeinsame Arbeitsbasis und verbessern die Zusammenarbeit zwischen den Disziplinen.