



## 2. ASIM Workshop Wismar - Modellierung, Regelung und Diagnose von Verbrennungsmotorprozessen

September 16-17, 2004  
Wismar

Zusammenfassung der Vorträge

ARGESIM Report no. 27  
ASIM-Mitteilungen AMB 92



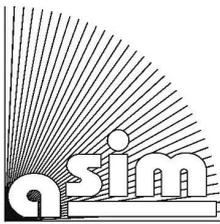
ARGESIM Report



ARGESIM Report

ARGESIM Report

ISBN 3-901608-27





*ARGESIM Report no. 27*

**C. Deatcu (Hrsg.)  
P. Dünow (Hrsg.)  
S. Pawletta (Hrsg.)  
T. Pawletta (Hrsg.)**

**ASIM – Mitteilungen AMB 92**

**2. ASIM Workshop Wismar –  
Modellierung, Regelung und Diagnose  
von Verbrennungsmotorprozessen**

**September 16-17, 2004  
Wismar**

organisiert durch:



**Hochschule Wismar  
Universität Rostock  
IAV GmbH Gifhorn  
ASIM FG Simulation technischer Systeme**

Zusammenfassung der Vorträge

**ARGESIM - Verlag, Vienna, 2004  
ISBN 3-901608-27**

## **ARGESIM Reports**

Published by **ARGESIM** and **ASIM**, Arbeitsgemeinschaft Simulation,  
Fachausschuss 4.5 der GI

### **Series Editor:**

Felix Breitenecker (ARGESIM / ASIM)  
Div. Simulation, Vienna University of Technology  
Wiedner Hauptstrasse 8 - 10, A - 1040 Vienna  
Tel: +43-1-58801-11452, Fax: +43-1-58801-11499  
Email: Felix.Breitenecker@tuwien.ac.at

### **ARGESIM Report no. 27** **ASIM – Mitteilungen AMB 92**

**Titel:** 2. ASIM Workshop Wismar –  
Modellierung, Regelung und Diagnose von Verbrennungsmotorprozessen

**Herausgeber:** C. Deatcu  
P. Dünow  
S. Pawletta  
T. Pawletta

Email: c.deatcu@mb.hs-wismar.de

### **ISBN 3-901608-27**

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Funksendung, der Wiedergabe auf photomechanischem oder ähnlichem Weg und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten.

© by ARGESIM / ASIM, Wien, 2003 – Hochschule Wismar

ARGE Simulation News (ARGESIM)  
c/o F. Breitenecker, Div. Simulation, Vienna Univ. of Technology  
Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria  
Tel.: +43-1-58801-11452, Fax: +43-1-58801-42098  
Email: info@argesim.org; WWW: <http://www.argesim.org>

**Druck:**  
Hochschule Wismar

## Vorwort

Mit der Serieneinführung neuartiger Konzepte wie der Direkteinspritzung, der variablen Ventilsteuerung, verschiedenen Aufladetechniken oder speziellen Abgasnachbehandlungsmethoden wurden in den letzten Jahren große Fortschritte hinsichtlich der abgegebenen Leistung, der Minimierung des Verbrauchs und der Verringerung der Abgasemission von Verbrennungskraftmaschinen erreicht.

Um das Potenzial dieser Konzepte maximal ausnutzen zu können, ist allerdings ein entsprechend leistungsfähiges elektronisches Motormanagement erforderlich. Wegen der gewachsenen Anzahl an Stellmöglichkeiten und zusätzlich verfügbarer Sensorsignale ist die Komplexität der Prozesse im steuerungstechnischen Sinne enorm angestiegen. Damit ergeben sich Herausforderungen für die Entwickler von Motorsteuerungsfunktionen, die nur durch die Nutzung moderner modell- und simulationsbasierter Ansätze und neuer regelungstheoretischer Erkenntnisse zu bewältigen sind. Die Anwendung von Ergebnissen der Theorieentwicklung zur Lösung einer gegebenen praktischen Aufgabenstellung gestaltet sich allerdings häufig schwierig, da neben der fachlichen Durchdringung der Ansätze die Beantwortung verschiedener Fragen nach der Auswahl des richtigen Verfahrens, der zu verwendenden Werkzeuge oder der Einsatzgrenzen einer Methode erforderlich ist.

Das Hauptanliegen dieses Workshops besteht deshalb darin, einen Meinungsaustausch zwischen in der Forschung tätigen Ingenieuren und Praxisvertretern zur Modellierung, Regelung und Diagnose von Verbrennungsmotorprozessen anzuregen. Es ist deshalb sehr erfreulich, dass Beiträge von Vertretern aus beiden "Lagern" in einem ausgewogenen Verhältnis präsentiert werden.

Die Herausgeber bedanken sich bei Herrn Prof. Breiteneker und Herrn Dr.-Ing. Wohnhaas für die Unterstützung des Workshops von Seiten der ASIM sowie bei den Mitorganisatoren B. Lampe und W. Drewelow von der Universität Rostock, M. Schultalbers und T. Jeinsch von der IAV-GmbH und den Mitarbeitern der Forschungsgruppe Computational Engineering und Automation der Hochschule Wismar, C. Kadow und R. Fink.

Wismar, den 15.09.2004,

P. Dünow, C. Deatcu, S. Pawletta, T. Pawletta



## Inhalt

<b>T. Jeinsch, M. Köller, W. Drewelow, M. Schultalbers, P. Dünow</b> Modellbasierte Detektion von Verbrennungsaussetzern.....	1
<b>S. X. Ding, P. Zhang</b> Entwurf eines fehlertoleranten Überwachungssystems und dessen Anwendung in Kraftfahrzeugen.....	5
<b>N. Weinhold, S. X. Ding, D. Heyden, M. Schultalbers, T. Jeinsch</b> Integration von Fehlerdiagnose in bestehende Regelkreisstrukturen: Ein Ansatz am Beispiel der Füllungserfassung eines BDE-Ottomotors.....	13
<b>M. S. Sangha, J. B. Gomm, D. L. Yu, G. F. Page</b> An Investigation of Neural Networks in Fault Diagnosis of an Automotive Engine Air Path.....	23
<b>H.-G. Schulz, S. Schneider, S. X. Ding, E. L. Ding</b> Beobachterentwurf für Systeme mit veränderlichen Parametern.....	33
<b>T. Schmidt, H. Fehrenbach</b> Verbrennungsmerkmale aus der Motordrehzahl.....	43
<b>S. Lochmann, G. Bramann</b> <i>Fibre-optical Monitoring of Exhaust Emissions from Road Vehicles.....</i>	57
<b>C. Wilhelm, M. Ayeb, Th. Winsel, H. J. Theuerkauf</b> HiL-gestützte Kalibrierung mit kombinierten physikalisch/neuronalen Prozessmodellen.....	63
<b>H.-M. Schulz</b> Verteilte Simulationen für Automotive Anwendungen.....	77
<b>J.-P. Blath</b> <i>Modellbasierte Auslegung von Fahrfunktionen.....</i>	89
<b>U. Lezius, W. Drewelow, M. Schultalbers</b> Potenzial zur Verbesserung der Klopfregelung in Ottomotoren.....	101
<b>C. Janssen, E. Hassel, J. Nocke, A. Schulze, N. Chindaprasert, M. Schultalbers, W. Gottschalk</b> <i>Modellierung und Simulation des ottomotorischen Verbrennungsprozesses mittels Ein- und Zweizonenmodell.....</i>	111
<b>N. Chindaprasert, E. Hassel, J. Nocke, C. Janssen, H. Rabba, M. Köller</b> <i>Two-Zone Combustion Model in S.I. Engine Using Polytopic Index.....</i>	119
<b>A. Schulze, E. Hassel, J. Nocke, C. Janssen, O. Magnor, G. Hübner</b> <i>Rechnergestützte Katalysatorauslegung.....</i>	133
<b>R. Dittmar</b> Model Predictive Control - Erfahrungen der Anwendung in der Prozessindustrie.....	141
<b>S.W. Wang, D. L. Yu, J. B. Gomm, G. F. Page, S. S. Douglas</b> Neural Network Modelling and Predictive Control of An IC Engine.....	151
<b>O. Simanski, R. Kähler</b> Anwendung von adaptiven, prädiktiven Reglern in der Anästhesie.....	161
<b>K. Lekhadia, H.-P. Dünow, T. Jeinsch</b> Active Set Method Approach for Real Time Model Predictive Engine Control.....	173

(kursiv gedruckte Vorträge sind ausschließlich mit Präsentationsfolien im Tagungsband enthalten)



# Modellbasierte Detektion von Verbrennungsaussetzern

T. Jeinsch, M. Köller, W. Drewelow, M. Schultalbers, P. Dünow

## Einleitung und Problemstellung

Nachdem die Verfügbarkeit, Sicherheit und der Komfort lange Zeit bestimmende Kräfte bei der Entwicklung von Verbrennungsmotoren waren, rückt nun verstärkt durch den Gesetzgeber die Umweltverträglichkeit in den Blickpunkt der Automobilindustrie. Dabei kommt der Überwachung abgasrelevanter Teilsysteme des Motors im Betrieb eine wachsende Bedeutung zu. Richtungsweisend dafür sind die von der kalifornischen Umweltbehörde, California Air Resource Board (CARB), herausgegebenen Vorschriften zur On-Board Diagnose II (OBD II)[2]. Ein wesentlicher Punkt der On-Board Diagnose dient der Sicherstellung, dass die Katalysatorwirkung während der Lebensdauer des Motors sichergestellt wird. In diesem Zusammenhang spielt die Erkennung von Verbrennungsaussetzern eine wesentliche Rolle. Als Verbrennungsaussetzer bezeichnet man Ereignisse, bei denen eine Verbrennung des Kraftstoffs ausgeblieben ist. Die Ursachen für Verbrennungsaussetzer sind vielfältig und reichen von fehlerhaften Zündanlagen bis zur ungünstigen Zusammensetzung des Luft-Kraftstoffgemischs [4]. Das Auftreten von Mehrfach- oder Daueraussetzern kann den Katalysator durch die Nachverbrennung von unverbranntem Gemisch überhitzen und damit dauerhaft schädigen. Doch bereits Einzelaussetzer führen zu einer Erhöhung des Schadstoffausstoßes und können somit zur Überschreitung der vorgeschriebenen Grenzwerte für die Emission von Kohlenwasserstoffen und Kohlenmonoxid führen.

Neben den gesetzlichen Regelungen zur Entwicklung und Bereitstellung von Überwachungs- und Diagnoseverfahren ist aus dem Blickwinkel der Automobilhersteller für die serientaugliche Umsetzung eine weitgehend kostenneutrale Realisierung wünschenswert. Dies ist ein wichtiges Argument bei der Auswahl möglicher Methoden und Verfahren. In diesem Zusammenhang wird das Ziel verfolgt, die steigenden Kosten durch Mehrfachnutzen dieser Verfahren beispielsweise zur Verbesserung des Komforts und der Leistungsentfaltung anzuwenden.

In der Praxis erfolgt die Detektion von Verbrennungsaussetzern häufig über die Erkennung von Drehzahlschwankungen und über die Ionisationsstrommessung. Dem erstgenannten Ansatz liegt die Idee zugrunde, dass ein Verbrennungsaussetzer zu geringen aber messbaren Drehzahlschwankungen führt. Dieses Verfahren wird jedoch durch verschiedene Einflussfaktoren, wie z.B. der Zylinderzahl sowie bestimmten Last- und Drehzahlbetriebspunkten, in seiner Anwendung eingeschränkt. Das zweite Verfahren basiert auf der unterschiedlichen Leitfähigkeit des brennenden und unverbrannten Gemischs. Es ist jedoch durch eine zusätzliche hardwaretechnische Modifikation zur Strommessung mit Zündkerzenelektroden geprägt. Ein weiterer Ansatz zur Detektion von Verbrennungsaussetzern ist die Auswertung des Körperschallsignals der bereits installierten Klopfensoren [4]. Trotz guter bis sehr guter Ergebnisse hat sich dieses Verfahren in der Serie bisher nicht durchgesetzt.

In der Zukunft müssen fehlende Verbrennungen über den gesamten Drehzahl- und Lastbereich erkannt werden, so dass signalgestützte Verfahren auf Basis von Drehzahlauswertungen an ihre Grenzen stoßen. Das Motorenangebot umfasst neben 3- und 4- auch 5-, 6-, 8-, 10- und 12- Zylinder-Motoren. Damit wird die Anwendung der Ionisationsstrommessung durch spezielle Hardwarelösungen sehr kostenintensiv. Diese steigenden Anforderungen machen es notwendig zukünftig neue Ideen und Ansätze zu verfolgen.

Im folgenden wird ein modellbasierter Ansatz zur Detektion von Verbrennungsaussetzern vorgestellt. Zur Detektion von Verbrennungsaussetzern nutzt man die Erkenntnis aus, dass eine nicht erfolgte Verbrennung zu signifikanten Änderungen des Drehmomentenverlaufs des Motors gegenüber dem Normalbetrieb führt. Durch den Vergleich der Drehmomentenverläufe kann zwischen dem Nominalbetrieb und einem Betrieb mit Verbrennungsaussetzer unterschieden werden. Motoren in Serienfahr-

zeugen besitzen keine Sensoren zur Erfassung des erzeugten Gasdrehmoments. Der modellbasierte Ansatz nutzt deshalb den zeitlichen Verlauf der Kurbel- und Nockenwellendrehung, um den Drehmomentenverlauf des Motors zu berechnen. Diese Überlegung ist auf Untersuchungen von Fehrenbach et al. zur Bestimmung der Wechselanteile des Gasdrehmoments [1] zurückzuführen.

## Modellgestützte Bestimmung der Wechselanteile des Gasdrehmoments

Die dem vorliegenden Verfahren zugrundeliegende Modellvorstellung geht von der vereinfachten Annahme aus, dass die Kurbelwelle hinreichend steif und entkoppelt vom Antriebsstrang vorliegt [1]. Damit ergibt sich die Leistungsbilanz an der Kurbelwelle mit dem Massenträgheitsmoment der rotierenden Massen  $\Theta$ , der Winkelgeschwindigkeit  $\dot{\varphi}$ , dem Gasdrehmoment  $M_g$ , dem Reibdrehmoment  $M_r$  und dem Last- oder Nutzdrehmoment  $M_l$  zu

$$\frac{\partial}{\partial t} \left( \frac{1}{2} \Theta(\varphi) \dot{\varphi}^2 \right) = (M_g - M_r - M_l) \dot{\varphi}. \quad (1)$$

Das veränderliche Massenträgheitsmoment der rotierenden Massen  $\Theta(\varphi)$  wird durch die oszillierende Masse des Pleuels und des Kolbens geprägt und ergibt sich nach der Differentiation zu:

$$\frac{\partial}{\partial t} \left( \frac{1}{2} \Theta(\varphi) \dot{\varphi}^2 \right) = \Theta(\varphi) \dot{\varphi} \ddot{\varphi} + \frac{1}{2} \frac{\partial \Theta(\varphi)}{\partial t} \dot{\varphi}^2 = \Theta(\varphi) \dot{\varphi} \ddot{\varphi} + \frac{1}{2} \frac{\partial \Theta(\varphi)}{\partial \varphi} \dot{\varphi}^3. \quad (2)$$

Damit folgt aus der Leistungsbilanz (1) die Momentenbilanz an der Kurbelwelle

$$\Theta(\varphi) \ddot{\varphi} + \frac{1}{2} \Theta'(\varphi) \dot{\varphi}^2 = M_g - M_r - M_l. \quad (3)$$

Die Grundidee des Modellansatzes beruht auf der Zerlegung des freien Drehmomentes an der Kurbelwelle  $\Theta(\varphi) \ddot{\varphi}$  in einen Gleich- und Wechselanteil. Das Gasdrehmoment  $M_g$  besteht aus dem eigentlichen Antriebsmoment in Form eines positiven Gleichanteils  $\bar{M}_g$  und einem Wechselanteil  $\tilde{M}_g$ . Das Reibmoment  $M_r$  und Last- oder Nutzmoment  $M_l$  kann nach Fehrenbach et al. in erster Näherung als in einem Arbeitspunkt (*Drehzahl/Last*) konstante Anteile  $\bar{M}_r$  und  $\bar{M}_l$  angenommen werden. Das Masseudrehmoment der oszillierenden Bauteile  $\frac{1}{2} \Theta'(\varphi) \dot{\varphi}^2$  bildet einen reinen Wechselanteil. Betrachtet man die Gleichung (3) für einen stationären Fall, in dem sich der positive Gleichanteil des Gasdrehmoments  $\bar{M}_g$  und die Summe von Reibmoment  $\bar{M}_r$  und Nutz- oder Lastmoment  $\bar{M}_l$  im Gleichgewicht befinden

$$\bar{M}_g - \bar{M}_r - \bar{M}_l = 0, \quad (4)$$

so ergibt sich durch folgende Vereinfachung

$$\Theta(\varphi) \ddot{\varphi} + \frac{1}{2} \Theta'(\varphi) \dot{\varphi}^2 = \tilde{M}_g. \quad (5)$$

Für den instationären Fall gilt:

$$\bar{M}_g - \bar{M}_r - \bar{M}_l \neq 0. \quad (6)$$

Fehrenbach et al. haben mit Hilfe von Instationärversuchen gezeigt, dass  $\bar{M}_g - \bar{M}_r - \bar{M}_l \ll \tilde{M}_g$  ist. Damit ist der Verlauf der Drehmomente innerhalb eines Arbeitsspiels von  $\varphi = 720^\circ$  als hinreichend stationär zu betrachten, so dass die Gleichung (5) auch für den instationären Fall angewendet werden kann.

Um den Wechselanteil des Gasdrehmoments als Summe des freien Drehmoments  $\Theta(\varphi) \ddot{\varphi}$  und Masseudrehmoments  $\frac{1}{2} \Theta'(\varphi) \dot{\varphi}^2$  aus Gleichung (5) zu berechnen, müssen  $\Theta(\varphi)$  und  $\Theta'(\varphi)$  sowie  $\dot{\varphi}$  und  $\ddot{\varphi}$  bestimmt werden. Das Massenträgheitsmoment der rotierenden Massen  $\Theta(\varphi)$  und dessen Ableitung nach dem Kurbelwinkel  $\Theta'(\varphi)$  sind ausschließlich von konstruktiven Motordaten abhängig. Die Ermittlung der Winkelgeschwindigkeit  $\dot{\varphi}$  und der Winkelbeschleunigung  $\ddot{\varphi}$  der Kurbelwelle basiert auf einer hochaufgelösten Winkelgeschwindigkeitsmessung mit anschließender Differenziation.

## Realisierung

Die Anwendung des vorgestellten Verfahrens zur modellgestützten Erkennung von Verbrennungsaussetzern basiert wesentlich auf der hochaufgelösten Messung der Kurbelwellendrehzahl. Im Gegensatz zur Methode der Ionisationsstrommessung ist keine zusätzliche Sensorik notwendig. Es ist jedoch erforderlich, das Signal des Sensors zur Messung der Kurbelwellendrehzahl gegenüber der konservativen Auswertung im Motorsteuergerät zu verbessern. Neben der Kurbelwellendrehzahl wird für Synchronisationszwecke das Nockenwellensignal sowie weitere Informationen über den Verbrennungsprozess, wie z.B. Liefergrad<sup>1</sup>, Zündwinkel, usw. genutzt.

Die Struktur der modellgestützten Erkennung von Verbrennungsaussetzern ist in der Abbildung 1 dargestellt. Der modellgestützte Residuengenerator beinhaltet die Berechnung des nominalen und aktuellen Gaswechsellmoments. Das erzielte Residuensignal wird durch die Residuenauswertungsfunktion bewertet und im Block Entscheidung mit dem berechneten Schwellwert  $J_{Sch}$  verglichen. Die Entscheidungsfunktion liefert schließlich die gewünschten Merkmale der Verbrennungsaussetzer, Anzahl und Zylinder.

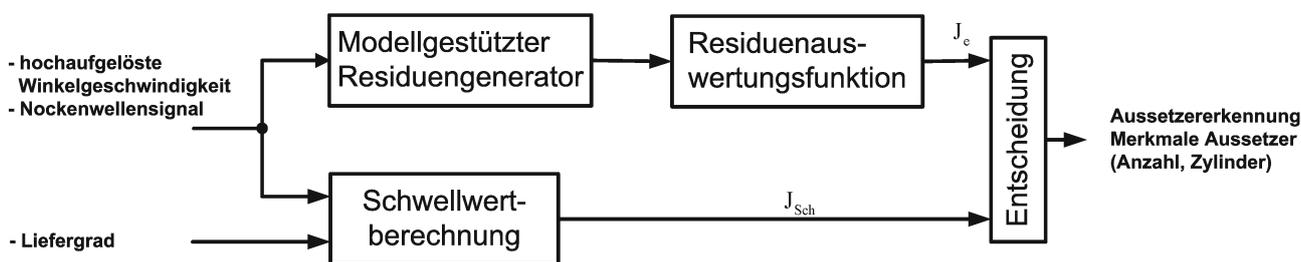


Bild 1: Struktur Modellbasierte Erkennung von Verbrennungsaussetzern

## Anwendung und Ergebnisse

Der vorgestellte modellbasierte Ansatz zur Erkennung von Verbrennungsaussetzern wurde unter anderem an einem 12 Zylinder-Motor untersucht. In der Abbildung 2 werden die Messgrößen aus dem Steuergerät für die Drehzahl  $n_{mot}$ , Auflösung  $120^\circ KW$  und das hochaufgelöste Drehzahlssignal  $n_{highres}$ , Auflösung  $6^\circ KW$  über dem Kurbelwinkel dargestellt. Eine reine signalbasierte Aussetzererkennung ist in diesem Fall schwierig, da die erwarteten Motordrehzahlschwankungen aufgrund der großen Zylinderzahl bei gleichzeitig niedriger Last und hoher Drehzahl nicht signifikant sind. Die Anwendung des modellbasierten Verfahrens zeigt, dass das ausgewertete Residuensignal  $J_e$  aus dem Vergleich der berechneten Gaswechsellmomente zum Zeitpunkt des Verbrennungsaussetzers die berechnete Fehlerschwelle  $J_{Sch}$  signifikant überschreitet.

## Zusammenfassung und Ausblick

Die Anwendung des vorgestellten modellgestützten Verfahrens zur Erkennung von Verbrennungsaussetzern hat gezeigt, dass es möglich ist, sehr gute Ergebnisse auch bei schwierigen Betriebsbedingungen zu erzielen. Zusätzlich bietet sich bei diesem Verfahren die Möglichkeit, ausgehend vom berechneten Gaswechsellmoment, auf weitere wichtige Größen des Verbrennungsprozesses zu schließen.

<sup>1</sup>Der Liefergrad  $\lambda_l$  ist mit der im Zylinder verbleibenden Frischladung (Luft oder Gemisch) je Arbeitsspiel  $m_{Fr}$  und der theoretischen Ladung je Arbeitsspiel bei Füllung des Hubvolumens mit Luft bzw. Gemisch bei Außenzustand  $m_{th}$  definiert als  $\lambda_l = \frac{m_{Fr}}{m_{th}}$ .

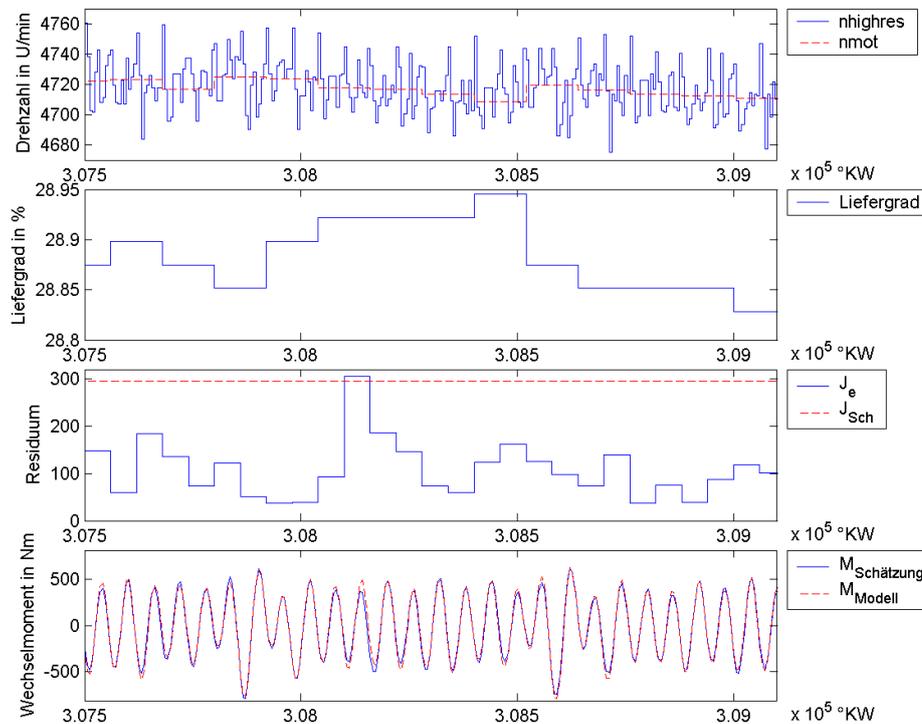


Bild 2: Realisierung: Detektion Einzelverbrennungsaussetzer Zylinder 12

## Literatur

- [1] Fehrenbach, Hermann; *Berechnung des Brennraumdruckverlaufes aus der Kurbelwellen-Winkelgeschwindigkeit von Verbrennungsmotoren*, VDI-Verlag GmbH, Düsseldorf, 1991
- [2] California Air Resources Board; *Technical Status Update and Proposed Revisions to Malfunctions and Diagnostic Systems Requirements Applicable to 1994 and Subsequent California Passenger Cars, Light-Duty Trucks, and Medium-Duty Vehicles - OBD II*, CARB Staff Report, 1991
- [3] Henn, Michael.; *On-Board Diagnose der Verbrennung von Ottomotoren*, Dissertation, Universität Fridericiana Karlsruhe, 1995
- [4] Lindemann, Michael; Filbert, Dieter; *Methoden der Verbrennungsaussetzer-Erkennung mit Klopfensoren*, MTZ (63), 3/2002, Seiten 194-200
- [5] Kiencke, Uwe; *Engine misfire detection*, Control engineering practice (7), 2/1999, Seiten 203-208

## Verfasser

Dr.-Ing. Torsten Jeinsch  
 Dipl.-Ing. Malte Köller  
 Dr.-Ing. Wolfgang Drewelow  
 Dipl.-Ing. Matthias Schultalbers  
 IAV GmbH  
 Büro Wismar  
 Philipp-Müller-Straße 12  
 D-23966 Wismar  
 Tel.-Nr.: +49-(0)3841-36001-21  
 Fax-Nr.: +49-(0)3841-36001-24  
 Email: torsten.jeinsch@iav.de

Prof.Dr.-Ing. Hans-Peter Dünow  
 Hochschule Wismar  
 Fachhochschule für Technik  
 Wirtschaft und Gestaltung  
 Fachbereich Elektrotechnik und Informatik  
 Philipp-Müller-Straße  
 Postfach 1210  
 D-23952 Wismar  
 Tel.-Nr.: +49-(0)3841-753 511  
 Fax-Nr.: +49-(0)3841-753 130  
 Email: p.duenow@et.hs-wismar.de

# Entwurf eines fehlertoleranten Überwachungssystems Und dessen Anwendung in Kraftfahrzeugen

S. X. Ding, P. Zhang

Institut für Automatisierungstechnik und komplexe Systeme  
Universität Duisburg-Essen, Bismarckstr. 81 BB, 47057 Duisburg

**Kurzfassung:** Fehlerdiagnose von dynamischen Prozessen gewinnt in den letzten Jahren zunehmend an Bedeutung. Modellgestützte Prozessüberwachung markiert den Stand der Technik. Im vorliegenden Beitrag wird ein Entwurfsverfahren für fehlertolerante Überwachungssysteme vorgestellt. Um möglichen Sensorfehler zu tolerieren, wird eine Reihe von Beobachtern/Estimator entworfen. Das vorgeschlagene Verfahren wird anhand von einem Beispiel des Fahrquerdynamikregelsystems veranschaulicht.

## Design of Fault Tolerant Monitoring Systems and Its Application to Vehicle Dynamic Control Systems

**Abstract:** Fault detection of dynamic systems have attracted increasingly attention in the last years. Model-based process monitoring is the state of art. In this paper, a design approach of fault tolerant monitoring systems is introduced. To be tolerant to possible sensor faults, different observers/estimator are then proposed. The proposed approach is illustrated by an example of vehicle lateral dynamic system.

### 1. INTRODUCTION

In recent years, the application of advanced model-based estimation technologies to increasing the reliability and safety of vehicle control systems like ESP, ACC has received considerable attention (van Zanten et al., 1998; Isermann, 2001). One of the current research activities in this area is to improve the performance of the estimation systems, to ensure an early detection of faults in the sensors integrated in the control systems and to make the estimation systems as tolerant as possible to the sensor faults (Ding et al., 2000; Isermann, 2001).

In this contribution, we focus our attention on the development of a fault tolerant estimation and monitoring system for the vehicle lateral dynamics. Considering that „car driving” is a process that is strongly influenced by many unknown factors, the core of our estimation and monitoring system is an adaptive observer that delivers a satisfactory estimation of the vehicle lateral dynamics for a wide range of driving situations. This adaptive observer is constructed on the basis of system knowledge presented in the form of a mathematical model and on the processing of on-line information like (known) input signals and system measurements delivered by the sensors integrated in the vehicle. To be tolerant to possible sensor faults, different observers/estimator are integrated in the estimation system, which are designed by taking into account different fault situations such that the degradation in the estimation

performance is limited to an acceptable level when one of the sensors fails.

Due to the limited paper length, we shall concentrate our attention in this contribution on the development of the adaptive observer scheme and on the description of the basic idea of the fault tolerant estimation and monitoring as well as the evaluation of the system performance.

### 2. PROBLEM FORMULATION

#### 2.1 Lateral dynamics and sensor models

For the description of vehicle lateral dynamics, the well-known bicycle model given below is considered:

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \delta_{St}^* \quad (1)$$

$$a_{11} = -\frac{c'_{\alpha f} \cos \delta_{St}^* + c_{\alpha r}}{m v_{ref} \cos \beta}$$

$$a_{12} = \frac{l_r c_{\alpha r} - l_f c'_{\alpha f} \cos \delta_{St}^*}{m v_{ref}^2 \cos \beta} - 1$$

$$a_{21} = \frac{1}{I_z} (l_r c_{\alpha r} - l_f \cos \delta_{St}^* c'_{\alpha f})$$

$$a_{22} = \frac{-1}{I_z v_{ref}} (l_f^2 c'_{\alpha f} \cos \delta_{St}^* + l_r^2 c_{\alpha r})$$

$$b_1 = \frac{c'_{\alpha_f} \cos \delta_{St}^*}{m v_{ref} \cos \beta}, \quad b_2 = \frac{l_f c'_{\alpha_f} \cos \delta_{St}^*}{I_z}$$

where  $\beta$ ,  $r$  denote the side slip angle and yaw rate,  $c'_{\alpha_f}$ ,  $c_{\alpha_r}$  the cornering stiffness of the front and rear tyres respectively. It is assumed that  $\cos \beta = 1$ ,  $\cos \delta_{St}^* = 1$  and the reference velocity  $v_{ref}$  is known.  $l_f$ ,  $l_r$ ,  $I_z$ ,  $m$  are some constants depending on the vehicle type under consideration (Mitschke, 1990; Kiencke and Nielsen, 1999; Börner et al., 2002). The input signal  $\delta_{St}^*$  denotes the relative steering wheel angle.

Our task of estimating and monitoring lateral dynamics consists in a) *estimation of the side slip angle* b) *estimation of the yaw rate and lateral acceleration*, which can be used for generating residual signals.

To this end, two sensor signals are available:

$$a_y = v_{ref} (\dot{\beta} + r) := y_1, \quad y_2 = r \quad (2)$$

with  $a_y$  denoting the lateral acceleration.

It is well-known that model (1) holds only under certain conditions (Mitschke, 1990; Kiencke and Nielsen, 1999). For our purpose, it is assumed in the following study that these conditions hold.

## 2.2 A brief description of the fault tolerant scheme

Fig.1 sketches the structure of the fault tolerant estimation and monitoring system under consideration.

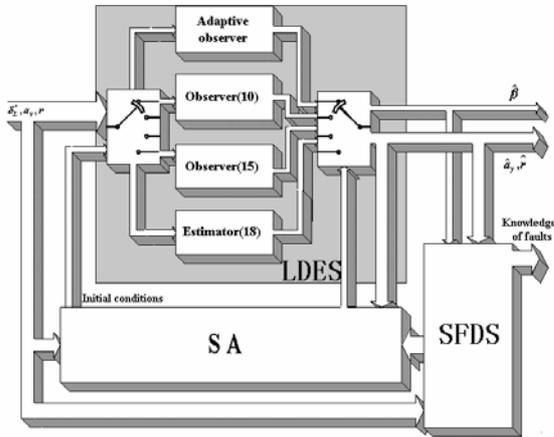


Fig.1: structure of the fault tolerant estimation and monitoring system

This system consists of three blocks: a) the lateral dynamics estimation system (LDES) b) the sensor fault detection system (SFDS) and c) the switching algorithm (SA). LDES delivers an estimation of side slip angle as well as estimations for yaw rate and lateral acceleration for the purpose of residual generation (Isermann 2001). LDES consists of four observers/estimators used in four different situations: S1) fault-free S2) yaw rate sensor is defect S3) lateral acceleration sensor fails S4) both of the sensors are defect. The switching between the observers (which can also be understood as an observer reconfiguration) will be controlled by SA. The task of SFDS consists in the detection of sensor faults based

on an evaluation of the residual signals, including  $r_1 = a_y - \hat{a}_y$ ,  $r_2 = r - \hat{r}$  with  $\hat{a}_y, \hat{r}$  denoting the estimation of lateral acceleration and yaw rate delivered by LDES. The unit SA has two functions: a) to decide which sensor signals are available and correspondingly which observer will be used for the estimation purpose. This decision is made based on the fault knowledge delivered by SFDS. b) to calculate the needed switching procedure.

## 2.3 Task formulation

In this contribution our attention is focused on the design of the observers/estimator integrated in LDES and on the analysis of the estimation performance of each observer/estimator.

Remember that one of the objectives of our study is to improve the observer performance in normal (fault-free) situations. In the vehicle bicycle model of the lateral dynamics, model uncertainty is mainly caused by the change of tyre cornering stiffness (Börner et al., 2002). In an early study, it was recognised that the estimation performance of an observer strongly depends on the vehicle velocity and driving manoeuvres. Thus, *our first task consists in the development of an adaptive observer on the basis of the vehicle bicycle model (1) and both sensor models given in (2).*

*Our further task is the observer reconfiguration such that the lateral dynamics can be estimated even if one or both of the sensors fail and the SFDS can thus detect other fault.*

*To ensure that the performance degradation is limited to an acceptable level, a performance analysis is of practical interest. This is the last part of our study presented in this contribution.*

## 3. ADAPTIVE OBSERVER SCHEME

During the last two decades, adaptive observer technology has been well established. Given system model (1) and sensor model (2), standard methods, for instance as described in (Narendra and Annaswamy, 1989), can be used for the design of an adaptive observer. On the other side, the strictly limited capacity available for the on-line calculation and for the microprocessor memory requires that the demand for the on-line calculation should be kept as small as possible. This motivates us to develop an adaptive observer which fulfils the requirements both on the performance and on the on-line calculation.

### 3.1 Model used for the adaptive observer design

It follows from model (1) that changes of the tyre side stiffness lead to changes in all parts of the system matrix and of the input vector. For such kind of model uncertainty, observer performance will be strongly affected. On the other hand, remember that the task of an observer is, different from the control objective, to process on-line sensor signals in order to re-construct the system state variables. To this end, we introduce below a model which is a combination of dynamics

model (1) and sensor model (2):

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \end{bmatrix} \delta_{St}^* + \begin{bmatrix} 1 \\ v_{ref} \\ 0 \end{bmatrix} a_y \quad (3)$$

It is evident that the main difference between models (1) and (3) consists in that in model (3) the model uncertainty is well structured and restricted.

The adaptive observer presented in this section is based on the assumption that

$$c_{ar} = kc'_{af}$$

This assumption is made on the stiffness of the steering mechanism and holds in different driving situations (Börner et al., 2002). We assume that  $k$  is a known constant whose value can be identified depending on the vehicle types under consideration. As a result, in system model (3), only  $c'_{af}$  is uncertain

and the model can be re-written into:

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ k_{14}c'_{af} & k_{15}c'_{af} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{v_{ref}} \\ k_{16}c'_{af} & 0 \end{bmatrix} \begin{bmatrix} \delta_{St}^* \\ a_y \end{bmatrix}$$

$$k_{14} = \frac{1}{I_z}(l_r k - l_f), \quad k_{15} = \frac{-1}{I_z v_{ref}}(l_f^2 + l_r^2 k), \quad k_{16} = \frac{l_f}{I_z} \quad (4)$$

### 3.2 Design of the adaptive observer

For our purpose, the following state transformation is first introduced

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} k_{14}c'_{af} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} \quad (5)$$

As a result, we have

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & -k_{14}c'_{af} \\ 1 & k_{15}c'_{af} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 & \frac{k_{14}c'_{af}}{v_{ref}} \\ k_{16}c'_{af} & 0 \end{bmatrix} \begin{bmatrix} \delta_{St}^* \\ a_y \end{bmatrix}$$

$$+ \begin{bmatrix} k_{14} \\ 0 \end{bmatrix} \beta \dot{c}'_{af} \quad (6)$$

$$a_y = v_{ref}(\dot{\beta} + r)$$

$$= -\frac{(k+1)}{m}c'_{af}\beta + \frac{(l_r k - l_f)}{m v_{ref}}c'_{af}r + \frac{1}{m}c'_{af}\delta_{St}^*$$

$$= k_{11}z_1 + k_{12}c'_{af}r + k_{13}c'_{af}\delta_{St}^*$$

$$k_{11} = \frac{-1-k}{m k_{14}}, \quad k_{12} = \frac{(l_r k - l_f)}{m v_{ref}}, \quad k_{13} = \frac{1}{m}$$

Based on system model (6), an adaptive observer is now constructed as follows:

$$\begin{bmatrix} \dot{\hat{z}}_1 \\ \dot{\hat{z}}_2 \end{bmatrix} = \begin{bmatrix} 0 & -l_{12} \\ 1 & -l_{22} \end{bmatrix} \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \end{bmatrix} + \begin{bmatrix} \frac{k_{14}}{v_{ref}}y_1 - k_{14}y_2 \\ k_{16}\delta_{St}^* + k_{15}y_2 \end{bmatrix} \hat{c}'_{af}$$

$$+ \begin{bmatrix} l_{11} \\ l_{21} \end{bmatrix} \Delta y_1 + \begin{bmatrix} l_{12} \\ l_{22} \end{bmatrix} y_2$$

$$\Delta y_1 = y_1 - k_{11}\hat{z}_1 - (k_{12}y_2 + k_{13}\delta_{St}^*)\hat{c}'_{af}$$

$$\dot{\hat{c}}'_{af} = \frac{1}{c_{11}} \begin{bmatrix} k_{14} \\ v_{ref} \end{bmatrix} y_1 - k_{14}y_2 - l_{11}(k_{12}y_2 + k_{13}\delta_{St}^*) \Delta y_1$$

$$+ \begin{bmatrix} k_{16}\delta_{St}^* + k_{15}y_2 - l_{21}(k_{12}y_2 + k_{13}\delta_{St}^*) \end{bmatrix} \Delta y_2$$

$$\hat{\beta} = \frac{\hat{z}_1}{k_{14}\hat{c}'_{af}}, \quad \Delta y_2 = y_2 - \hat{z}_2$$

whose estimation error dynamics is governed by

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} -l_{11}k_{11} & -l_{12} \\ 1 - l_{21}k_{11} & -l_{22} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} - \begin{bmatrix} k_{14} \\ 0 \end{bmatrix} \dot{c}'_{af} \beta$$

$$+ \begin{bmatrix} \frac{k_{14}}{v_{ref}}y_1 - k_{14}y_2 - l_{11}(k_{12}y_2 + k_{13}\delta_{St}^*) \\ k_{16}\delta_{St}^* + k_{15}y_2 - l_{21}(k_{12}y_2 + k_{13}\delta_{St}^*) \end{bmatrix} \Delta c'_{af}$$

$$\Delta \dot{c}'_{af} = \dot{c}'_{af} - \hat{c}'_{af}$$

$$= \dot{c}'_{af} - \begin{bmatrix} \frac{k_{14}}{v_{ref}}y_1 - k_{14}y_2 - l_{11}(k_{12}y_2 + k_{13}\delta_{St}^*) \\ -[k_{16}\delta_{St}^* + k_{15}y_2 - l_{21}(k_{12}y_2 + k_{13}\delta_{St}^*)] \\ -\frac{1}{k_{11}} \begin{bmatrix} k_{14} \\ v_{ref} \end{bmatrix} y_1 - k_{14}y_2 - l_{11}(k_{12}y_2 + k_{13}\delta_{St}^*) \end{bmatrix} \\ (k_{12}y_2 + k_{13}\delta_{St}^*) \Delta c'_{af} \end{bmatrix} e_1$$

$$e_1 = z_1 - \hat{z}_1, \quad e_2 = z_2 - \hat{z}_2, \quad \Delta c'_{af} = c'_{af} - \hat{c}'_{af}$$

By choosing observer gain matrix

$$L = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix}$$

in such a way that the eigenvalues of matrix

$$A_L = \begin{bmatrix} -l_{11}k_{11} & -l_{12} \\ 1 - l_{21}k_{11} & -l_{22} \end{bmatrix}$$

lie in the desired region of the left complex plane and

$$\begin{bmatrix} \frac{k_{14}}{v_{ref}}y_1 - k_{14}y_2 - l_{11}(k_{12}y_2 + k_{13}\delta_{St}^*) \\ k_{16}\delta_{St}^* + k_{15}y_2 - l_{21}(k_{12}y_2 + k_{13}\delta_{St}^*) \end{bmatrix} (k_{12}y_2 + k_{13}\delta_{St}^*) \leq 0$$

we have an asymptotically stable system. In order to prove it, consider the following Lyapunov function

$$V(e, \Delta c'_{af}) = e^T e + (\Delta c'_{af})^2, \quad e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

It turns out for  $\dot{c}'_{af} = 0$

$$\begin{aligned}
\dot{V}(e, \Delta c'_{af}) &= e^T (A_L^T + A_L) e \\
&+ 2e^T \begin{bmatrix} \frac{k_{14}}{v_{ref}} y_1 - k_{14} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \\ k_{16} \delta_{St}^* + k_{15} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \end{bmatrix} \Delta c'_{af} \\
&- 2e^T \begin{bmatrix} \frac{k_{14}}{v_{ref}} y_1 - k_{14} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \\ k_{16} \delta_{St}^* + k_{15} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \end{bmatrix} \Delta c'_{af} \\
&- \frac{2}{k_{11}} \left( \frac{k_{14}}{v_{ref}} y_1 - k_{14} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \right) \\
&(k_{12} y_2 + k_{13} \delta_{St}^*) (\Delta c'_{af})^2 \\
&= e^T (A_L^T + A_L) e \\
&- \frac{2}{k_{11}} \left( \frac{k_{14}}{v_{ref}} y_1 - k_{14} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \right) \\
&(k_{12} y_2 + k_{13} \delta_{St}^*) (\Delta c'_{af})^2
\end{aligned}$$

Note that  $k_{11} < 0$ . Thus, if the eigenvalues of matrix  $A_L$  lie in the left half complex plane and

$$\left[ \frac{k_{14}}{v_{ref}} y_1 - k_{14} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \right] (k_{12} y_2 + k_{13} \delta_{St}^*) \leq 0$$

holds, we have

$$\begin{aligned}
A_L^T + A_L &< 0, \\
-\frac{2}{k_{11}} \left( \frac{k_{14}}{v_{ref}} y_1 - k_{14} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \right) &(8) \\
(k_{12} y_2 + k_{13} \delta_{St}^*) (\Delta c'_{af})^2 &< 0 \Rightarrow \\
\dot{V}(e, \Delta c'_{af}) &< 0
\end{aligned}$$

This proves that the adaptive observer (7) is an asymptotically stable system.

For the re-construction of measurement variables, the following equations are used:

$$\hat{a}_y = c_{11} \hat{z}_1 + (k_{12} y_2 + k_{13} \delta_{St}^*) \hat{c}'_{af}, \quad \hat{r} = \hat{z}_2$$

and the residual signal

$$r_1 = a_y - \hat{a}_y, \quad r_2 = r - \hat{r}$$

is used for the aim of sensor fault detection.

The above-developed adaptive observer has the following features: a) Both sensor signals are used for the estimation purpose, although the sensor model used for the lateral acceleration estimation includes the tyre cornering stiffness to be identified; b) An auxiliary filter is not needed. These two features mark the main differences between the known adaptive observer schemes and the one developed in this contribution, and ensure that the requirements on the performance and on-line calculation are satisfied.

#### 4. FAULT TOLERANT ESTIMATION AND MONITORING

The adaptive observer presented above is based on the

processing of the signals delivered by the lateral acceleration and yaw rate sensors. It is well-known that a failure of one of the sensors will lead to a strong performance degradation. In this section, different estimation schemes will be presented, which deliver a stable estimation in case that one of the sensors fails.

##### 4.1 Estimation of lateral dynamics when the lateral acceleration sensor fails

In the presence of a fault in the lateral acceleration sensor, model (3) is no more valid and only the measurement from the yaw rate sensor is available. Hence, for our purpose of estimating lateral dynamics, model (1) is used for the observer construction. Suppose that

$$c'_{af} = c'_{af,0} + \Delta c'_{af}, \quad c_{ar} = c_{ar,0} + \Delta c_{ar}$$

with  $c'_{af,0}$ ,  $c_{ar,0}$ ,  $\Delta c'_{af}$ ,  $\Delta c_{ar}$  denoting nominal values and the possible changes of the cornering stiffness respectively, then model (1) can be re-written into

$$\begin{aligned}
\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} &= A_{1,0} \begin{bmatrix} \beta \\ r \end{bmatrix} + B_{1,0} \delta_{St}^* + \begin{bmatrix} d_1 (\Delta c'_{af}, \Delta c_{ar}) \\ d_2 (\Delta c'_{af}, \Delta c_{ar}) \end{bmatrix} \\
A_{1,0} &= \begin{bmatrix} \frac{c'_{af,0} + c_{ar,0}}{mv_{ref}} & \frac{l_f c'_{af,0} - l_r c_{ar,0}}{mv_{ref}^2} - 1 \\ \frac{l_r c_{ar,0} - l_f c'_{af,0}}{I_z} & \frac{l_r^2 c_{ar,0} + l_f^2 c'_{af,0}}{I_z v_{ref}} \end{bmatrix} \\
B_{1,0} &= \begin{bmatrix} \frac{c'_{af,0}}{mv_{ref}} \\ \frac{l_f c'_{af,0}}{I_z} \end{bmatrix} \\
d_1 &= -\frac{\Delta c'_{af} + \Delta c_{ar}}{mv_{ref}} \beta + \frac{l_f \Delta c'_{af} - l_r \Delta c_{ar}}{mv_{ref}^2} r + \frac{\Delta c'_{af} \delta_{St}^*}{mv_{ref}} \\
d_2 &= \frac{l_r \Delta c_{ar} - l_f \Delta c'_{af}}{I_z} \beta - \frac{l_r^2 \Delta c_{ar} + l_f^2 \Delta c'_{af}}{I_z v_{ref}} r \\
&+ \frac{l_f \Delta c'_{af} \delta_{St}^*}{I_z}
\end{aligned} \tag{9}$$

where  $d_1, d_2$  denote unknown inputs that represent the influence of changes in cornering stiffness. Based on model (9), an observer is constructed as follows:

$$\begin{bmatrix} \dot{\hat{\beta}} \\ \dot{\hat{r}} \end{bmatrix} = A_{1,0} \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + B_{1,0} \delta_{St}^* + L(y_2 - \hat{r}) \tag{10}$$

whose dynamics is governed by

$$\begin{aligned}
\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} &= (A_{1,0} - LC_1) \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + Ed \\
\begin{bmatrix} e_1(s) \\ e_2(s) \end{bmatrix} &= (sI - A_{1,0} + LC_1)^{-1} Ed(s)
\end{aligned} \tag{11}$$

$$C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

In order to minimise the estimation error, an  $H_\infty$  observer is designed under the performance index

$$\begin{aligned} \left\| \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \right\|_2 &\leq \gamma_1 \|d\|_2, \gamma_1 \rightarrow \min \\ \|(sI - A_{1,0} + LC_1)^{-1}E\|_\infty &\leq \gamma_1 \end{aligned} \quad (12)$$

with a desired assignment of the poles of the estimation error dynamics. To solve this optimisation problem, MATLAB LMI toolbox is used.

It follows from (2) that the following relationship can be used for the lateral acceleration estimation

$$\hat{a}_y = v_{ref} (\dot{\hat{\beta}} + \hat{r}) \quad (13)$$

and the residual signal

$$r_2 = r - \hat{r}$$

can still be used to detect the fault in the yaw rate sensor.

#### 4.2 Estimation of lateral dynamics when the yaw rate sensor fails

For our purpose of estimating the lateral dynamics using the available measurement, model (3) and sensor model given by

$$\begin{aligned} y_1 = a_y = v_{ref} (\dot{\beta} + r) &= C_2 \begin{bmatrix} \beta \\ r \end{bmatrix} + D_2 \delta_{St}^* \\ C_2 &= \begin{bmatrix} -\frac{c'_{af} + c_{ar}}{m} & \frac{l_r c_{ar} - l_f c'_{af}}{mv_{ref}} \end{bmatrix}, \quad D_2 = \frac{c'_{af}}{m} \end{aligned} \quad (14)$$

are used. The observer is constructed as follows:

$$\begin{aligned} \begin{bmatrix} \dot{\hat{\beta}} \\ \dot{\hat{r}} \end{bmatrix} &= A_{2,0} \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + B_{2,0} \delta_{St}^* + \begin{bmatrix} 1 \\ v_{ref} \\ 0 \end{bmatrix} a_y + L(a_y - \hat{a}_y) \\ \hat{a}_y &= C_{2,0} \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + D_{2,0} \delta_{St}^*, \quad B_{2,0} = \begin{bmatrix} 0 \\ l_f c'_{af,0} \\ I_z \end{bmatrix}, \quad D_{2,0} = \frac{c'_{af,0}}{m} \\ C_{2,0} &= \begin{bmatrix} \frac{c'_{af,0} + c_{ar,0}}{m} & \frac{l_r c_{ar,0} - l_f c'_{af,0}}{mv_{ref}} \end{bmatrix}, \quad L = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \\ A_{2,0} &= \begin{bmatrix} 0 & -1 \\ \frac{l_r c_{ar,0} - l_f c'_{af,0}}{I_z} & -\frac{l_f^2 c'_{af,0} + l_r^2 c_{ar,0}}{I_z v_{ref}} \end{bmatrix} \end{aligned} \quad (15)$$

To determine the design parameter  $L$ , we consider the estimation error dynamics which can be written as

$$\begin{aligned} \dot{e} &= (A_{2,0} - LC_{2,0})e + (E - LF)\bar{d} \\ e &= \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \beta - \hat{\beta} \\ r - \hat{r} \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \bar{d} &= \begin{bmatrix} \frac{l_r \Delta c_{ar} - l_f \Delta c'_{af}}{I_z} \beta - \frac{l_r^2 \Delta c_{ar} + l_f^2 \Delta c'_{af}}{I_z v_{ref}} r + \frac{l_f \Delta c'_{af} \delta_{St}^*}{I_z} \\ \frac{\Delta c_{ar} + \Delta c'_{af}}{m} \beta + \frac{l_r \Delta c_{ar} - l_f \Delta c'_{af}}{mv_{ref}} r + \frac{\Delta c'_{af} \delta_{St}^*}{m} \end{bmatrix} \end{aligned} \quad (16)$$

For the observer design, the following optimisation problem is solved for  $L$

$$\begin{aligned} \|e\|_2 \leq \gamma_2 \|\bar{d}\|_2, \gamma_2 \rightarrow \min \\ \|(sI - A_{2,0} + LC_{2,0})^{-1}(E - LF)\|_\infty \leq \gamma_2 \end{aligned} \quad (17)$$

Again, for the design purpose, MATLAB LMI

toolbox is used.

The residual signal

$$r_2 = a_y - \hat{a}_y = a_y - v_{ref} (\dot{\hat{\beta}} + \hat{r})$$

can still be used to detect the fault in the lateral acceleration sensor.

#### 4.3 Estimation of lateral dynamics when both sensors fail

It is evident that if no sensor signals are available we are only able to use model (1) for our estimation purpose, i.e.

$$\begin{aligned} \begin{bmatrix} \dot{\hat{\beta}} \\ \dot{\hat{r}} \end{bmatrix} &= A_{1,0} \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + B_{1,0} \delta_{St}^* \\ \hat{a}_y &= v_{ref} (\dot{\hat{\beta}} + \hat{r}) \end{aligned} \quad (18)$$

## 5. PERFORMANCE EVALUATION

In the last two sections, different estimation schemes have been presented aiming at a fault tolerant estimation of vehicle lateral dynamics and improving the estimation performance under the fault-free system operation condition. In practice, it is of significant interest to specify the estimation performance under normal operation condition and, in case that a sensor fails, the performance degradation, in order to ensure the monitoring performance.

#### 5.1 Performance evaluation of the adaptive observer

The adaptive observer is a time-varying system which can be written as follows:

$$\begin{aligned} \begin{bmatrix} \dot{\hat{z}}_1 \\ \dot{\hat{z}}_2 \\ \dot{\hat{c}}'_{af} \end{bmatrix} &= \begin{bmatrix} -l_{11}k_{11} & -l_{12} & \theta_1(t) \\ 1 - l_{21}k_{11} & -l_{22} & \theta_2(t) \\ \theta_1(t) & \theta_2(t) & \theta_3(t) \end{bmatrix} \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \hat{c}'_{af} \end{bmatrix} \\ &+ \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{21} \\ \theta_1(t) & \theta_2(t) \\ k_{11} & \theta_2(t) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ \theta_1(t) &= \frac{k_{14}}{v_{ref}} y_1 - k_{14} y_2 - l_{11} (k_{12} y_2 + k_{13} \delta_{St}^*) \\ \theta_2(t) &= k_{16} \delta_{St}^* + k_{15} y_2 - l_{21} (k_{12} y_2 + k_{13} \delta_{St}^*) \\ \theta_3(t) &= \frac{\theta_1(t)}{k_{11}} (k_{12} y_2 + k_{13} \delta_{St}^*) \end{aligned} \quad (19)$$

The required on-line calculation is therefore a time-varying system of the third order.

To evaluate the estimation performance, we summarise the estimation error dynamics into:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \Delta \dot{c}'_{af} \end{bmatrix} = \begin{bmatrix} -l_{11}k_{11} & -l_{12} & \theta_1(t) \\ 1 - l_{21}k_{11} & -l_{22} & \theta_2(t) \\ -\theta_1(t) & -\theta_2(t) & -\theta_3(t) \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \Delta c'_{af} \end{bmatrix} + \begin{bmatrix} -k_{14}\beta \\ 0 \\ 1 \end{bmatrix} \dot{c}'_{af}$$

(20)

Thus, it follows from the definition of the introduced Lyapunov function and (9) that for  $\dot{c}'_{\alpha f} = 0$

$$\begin{aligned} V(e, \Delta c'_{\alpha f}) &= e^T e + (\Delta c'_{\alpha f})^2 = e_\alpha^T e_\alpha > 0 \\ \dot{V}(e, \Delta c'_{\alpha f}) &\leq e_\alpha^T Q e_\alpha < 0 \\ e_\alpha^T &= [e_1 \quad e_2 \quad \Delta c'_{\alpha f}] \end{aligned}$$

where  $Q$  denotes a positive definite matrix, this, as well-known, means that there exist constants  $K_1, K_2 > 0$  such that for  $|\dot{c}'_{\alpha f}| \leq M$

$$\|e_\alpha\| < K_1 \|e_\alpha(0)\| + K_2$$

Note that

$$\lim_{t \rightarrow \infty} \|e_\alpha\| \leq K_3 M, \quad K_3 > 0 \quad \text{some constant}$$

since system (20) is an asymptotically stable system, the estimation error will become very small if the tyre cornering stiffness changes slowly. Moreover, it follows from (20) that the influence of the cornering stiffness change on the estimation system can be modelled as an unknown input.

### 5.2 Performance evaluation of observer schemes (10) and (15)

Both observers (10) and (15) are linear time invariant system of the second order. Thus, in comparison with adaptive observer (19), the demand for the on-line calculation is lower.

Since the estimation error dynamics of observers (10) and (15), given by (11) and (16) respectively, can be described by a linear time invariant system with a unknown input vector, there exist constants  $K_4, K_5 > 0$  such that for observer (10)

$$\lim_{t \rightarrow \infty} \|e\| \leq K_4 (d_1^2 + d_2^2)^{1/2} \quad (21)$$

and for observer (15)

$$\lim_{t \rightarrow \infty} \|e\| < K_5 (\bar{d}_1^2 + \bar{d}_2^2)^{1/2} \quad (22)$$

Recall that

$$\begin{aligned} d_1 &= -\frac{\Delta c'_{\alpha f} + \Delta c_{ar}}{mv_{ref}} \beta + \frac{l_f \Delta c'_{\alpha f} - l_r \Delta c_{ar}}{mv_{ref}^2} r + \frac{\Delta c'_{\alpha f} \delta_{St}^*}{mv_{ref}} \\ d_2 &= \frac{l_r \Delta c_{ar} - l_f \Delta c'_{\alpha f}}{I_z} \beta - \frac{l_r^2 \Delta c_{ar} + l_f^2 \Delta c'_{\alpha f}}{I_z v_{ref}} r \\ &\quad + \frac{l_f \Delta c'_{\alpha f} \delta_{St}^*}{I_z} \\ \bar{d}_1 &= \frac{l_r \Delta c_{ar} - l_f \Delta c'_{\alpha f}}{I_z} \beta - \frac{l_r^2 \Delta c_{ar} + l_f^2 \Delta c'_{\alpha f}}{I_z v_{ref}} r + \frac{l_f \Delta c'_{\alpha f} \delta_{St}^*}{I_z} \\ \bar{d}_2 &= \frac{\Delta c_{ar} + \Delta c'_{\alpha f}}{m} \beta + \frac{l_r \Delta c_{ar} - l_f \Delta c'_{\alpha f}}{mv_{ref}} r + \frac{\Delta c'_{\alpha f} \delta_{St}^*}{m} \end{aligned}$$

it becomes evident that the size of the estimation error, for both observers (10) and (15), not only depends on the change in the cornering stiffness but

also on the lateral dynamics. It results in a conservative estimation of the bound of the estimation error in order to cover a large range of driving situations.

It is worth pointing out that although the observer design is optimal in sense of (12), (17) respectively, the improvement of the system robustness against the unknown inputs is limited. The reason is evident, because for the given system structure with only one measurement and two unknown inputs the design freedom is strongly limited.

### 5.3 Performance evaluation of estimator (18)

From the viewpoint of the demand for the on-line calculation and the estimation error dynamics,

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = A_{1,0} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

estimator (18) is similar to observer (10). However, it is open-loop structured and there exists no possibility to improve the system robustness against the changes caused by the cornering stiffness. As a result, the performance of estimator (18) is the worst one among the observer schemes presented in this contribution.

## 6. CONCLUDING REMARKS

In this paper, a fault tolerant system for the estimation and monitoring of vehicle lateral dynamics has been presented. The core of this system is an adaptive observer that delivers a satisfactory estimation when both of the sensor signals are available. Further observers/estimator are integrated in the fault tolerant estimation and monitoring system such that an observer reconfiguration can be realised when a sensor fails.

All observers/estimator have been simulated for different and well-specified driving manoeuvres. To this end, software Carsim has been used, which provides users with a powerful tool that well simulates real driving situations.

It is well-known that uncertainties of the bicycle model depend strongly on driving manoeuvres and road conditions. An analytical computation of model uncertainties generally lead to very conservative estimates. Experimental analysis including simulation and driving tests is the way followed in the industry.

It should be remarked that the fault tolerant scheme presented in this contribution is only a part of an overall fault tolerant monitoring system for vehicle control systems, in which different sub-systems for vehicle dynamics estimation and detection of sensor faults are integrated. Also, both for the purposes of vehicle dynamics estimation and sensor fault detection, the multi-model scheme (Ding et al., 2000) is adopted. The valid estimation and fault detection are achieved by an evaluated fusion of estimations and residual signals delivered by different sub-estimation systems based on different models. The lateral dynamics estimation and the residual signals delivered by the observers proposed in this

contribution build only a part of these fusion data.

*Acknowledgement:* This work was in part supported by the European Commission under grant IST-2001-32122, IFATIS.

#### REFERENCES

- Börner, M, L. Andréani, P. Albertos and R. Isermann (2002) Detection of lateral vehicle driving conditions based on the characteristic velocity, *Proc. IFAC World Congress 2002*.
- Ding, E. L, H Fennel and S. X. Ding (2000), Model based diagnosis of sensor faults for ESP-systems, *Proc. of IFAC Symp. Safeprocess 2000*.
- Isermann, R. (2001) Diagnosis methods for electronic controlled vehicles, *Vehicle system dynamics*, Vol.36, p.77-117.
- Kiencke, U. and L. Nielsen, (1999), *Automotive Control Systems*. Springer-Verlag.
- Mitschke, M. (1990), *Dynamik der Kraftfahrzeuge*. Band C. Springer-Verlag.
- Narendra, K and A.M. Annaswamy (1989), *Stable Adaptive Systems*, Prentice-Hall International, Inc.
- van Zanten, A., Erhardt, R., K. Landesfeind und G. Pfaff (1998), VDC systems development and perspectives, *SAE Technical Paper Series 980235*.



# Integration von Fehlerdiagnose in bestehende Regelkreisstrukturen: Ein Ansatz am Beispiel der Füllungserfassung eines BDE-Ottomotors

N. Weinhold <sup>\*</sup>, S.X. Ding <sup>\*</sup>, D. Heyden <sup>\*</sup>, M. Schultalbers <sup>†</sup>, T. Jeinsch <sup>†</sup>

## Zusammenfassung

Mit zunehmender Verschärfung der Abgasemissionen in den 90er Jahren rückte die modellgestützte Regelungstechnik immer stärker in den Fokus der Automobilhersteller. Ziel des Einsatzes von modellgestützter Regelungstechnik ist es, auf der einen Seite vor allem den Anforderungen eines umweltbewussten Betriebs der Fahrzeuge Rechnung zu tragen und die geforderten Abgasnormen zu erfüllen, ohne jedoch auf der anderen Seite die Kosten für die Automobile erheblich in die Höhe zu treiben. Heutzutage ist die Nutzung von modellgestützten Verfahren in modernen Motormanagementsystemen Stand der Technik.

In diesem Beitrag wird die Integration neuer Fehlerdiagnosekonzepte in bestehenden Regelungssysteme am Beispiel der Füllungsregelung als Teil eines modernen Ottomotor-Managementsystems beschrieben.

## 1 Einleitung

Die dramatische Zunahme des Verkehrsaufkommens hat in den letzten Jahrzehnten zu immer strengeren Gesetzgebungen hinsichtlich der Abgasemissionen von Kraftfahrzeugen geführt. Aus heutiger Sicht werden Aspekte wie

- das steigende Umweltbewusstsein in der Öffentlichkeit,
- die zunehmend bedachte Nutzung von Ressourcen im Bereich der fossilen Brennstoffe,
- die viel diskutierte Klimaerwärmung durch die Kohlendioxidemission und
- die damit verbundenen gesetzlichen Anstrengungen zur Reduzierung des Kohlendioxidausstosses

zu einer zukünftig weiteren Verschärfung der Abgasemissionen führen. Die damit verbundenen stetig steigenden Anforderungen bezüglich der Abgasemissionen von modernen Verbrennungsmotoren stellen die Hersteller vor immer größere Herausforderungen.

---

<sup>\*</sup>Fachgebiet Automatisierungstechnik und komplexe Systeme, Fak. 5, Universität Duisburg-Essen, 47048 Duisburg

<sup>†</sup>Fachbereich Antriebselektronik Management-Systeme Ottomotoren, IAV GmbH, 38518 Gifhorn

Der Einsatz von modellgestützten Verfahren ist Stand der Technik und ermöglicht die Einhaltung der gesetzlichen Vorschriften. Jedoch führt die durch die vorgeschriebenen Grenzwerte nötige Genauigkeit bei den Modellen dazu, dass diese bereits sehr komplex sind. In Zukunft ist durch den mit der Genauigkeit stark steigenden Detailierungsgrad mit einer weiter steigenden Komplexität zu rechnen.

Neben dem Einsatz von z.B. Katalysatoren zur Reinigung des Abgases ist insbesondere die richtige, d.h. fehlerfreie Zuführung der Luft- und Kraftstoffmenge für die Abgaszusammensetzung und damit für die Einhaltung der gesetzlichen Bestimmungen entscheidend. Ziel ist es, ein Diagnosesystem in die bereits existierende, für regelungstechnische Zwecke eingesetzte Modellstruktur zu integrieren und somit die bereits in den Modellen vorhandenen Informationen zu nutzen. Für den hier angestrebten integrierten Entwurf des Fehlerdiagnosesystems sprechen insbesondere die Tatsachen, dass

- die Realisierung eines Fehlerdiagnosesystems auf Basis von Hardware-Redundanzen bei weitem zu teuer ist,
- die zu integrierenden Prozess- bzw. Teilprozessmodelle, welche die Abläufe im Motor in ausreichend genauem Maße beschreiben sehr komplex sind und
- die im Steuergerät und damit zur Online-Implementierung zur Verfügung stehenden Recourcen sehr begrenzt sind, wodurch die bei einer klassischen modellgestützten Fehlerdiagnose notwendige Vervielfältigung der Modelle zur Schaffung von Software-Redundanzen nicht möglich ist.

Die Integration eines Diagnosesystems in eine bestehende regelungstechnische Struktur setzt zunächst deren Kenntnis sowie die Kenntnis der Rahmenbedingungen voraus. Des Weiteren ist eine Analyse und die Interpretation dieser Struktur erforderlich.

In diesem Beitrag wird die für die Integration der Fehlerdiagnose in das Luft- und Kraftstoff-Regelungssystem relevante Systemstruktur skizziert. Für dieses, auch als Füllungserfassung bezeichnete, System wird eine Auswahl an Signalen (Residuen) zur Fehlerentdeckung vorgestellt und deren Wahl aus regelungstechnischer Sicht begründet. Die Wirksamkeit der damit verbundenen Fehlerdiagnose wird mit Hilfe von Motorprüfstandsdaten verifiziert und in Ansätzen dokumentiert.

## **2 Stand der Technik und Ausgangspunkt der Betrachtungen**

Dieser Abschnitt befasst sich mit der für die Integration der Diagnose vorhandenen und relevanten Regelungsstruktur für den Teilprozess der Luft- und Kraftstoffbefüllung eines BDE-Ottomotors. Für einen ersten allgemeineren Überblick über die vorherrschenden komplexen Strukturen im Motormanagementbereich sei auf [1] verwiesen. Für eine Eruerung der physikalischen Hintergründe sowie möglicher Regelungsstrategien eignen sich [2–5]. Die für die Füllungsregelung in Steuergeräten genutzte prinzipielle Struktur ist in Ansätzen in [6] beschrieben. Aus heutiger Sicht steht bei den eingesetzten modellbasierten Regelungsstrukturen die Erzielung eines hohen Maßes an stationärer Genauigkeit bei gleichzeitig guter dynamischer Aktualität im Vordergrund.

## 2.1 Relevante Regelkreisstruktur

Der Einsatz der modellgestützten Verfahren in modernen Motormanagementsystemen zeigt sich insbesondere durch die Implementierung eines physikalisch basierten Prozessmodells. Die für die Füllungserfassung relevante Struktur ist in Abbildung 1 dargestellt.

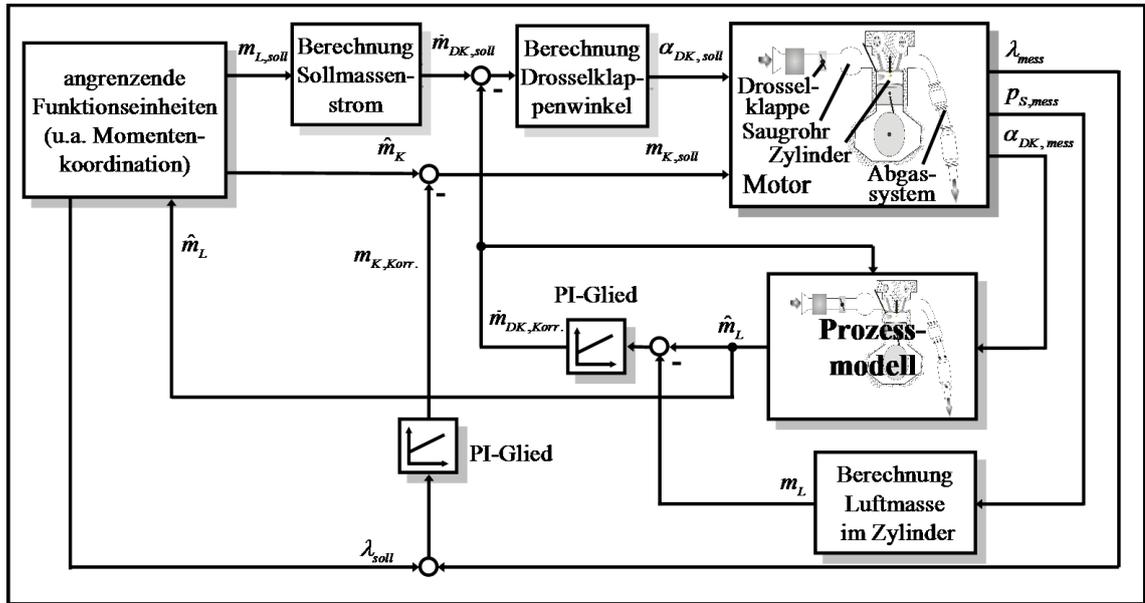


Abbildung 1: Regelungsstruktur der Füllungserfassung

Als Stellgrößen werden der Solldrosselklappenwinkel  $\alpha_{DK,soll}$  und die Kraftstoffsollmenge  $m_{K,soll}$  an die Aktuatoren ausgegeben. Gemessen werden neben der Drosselklappenposition  $\alpha_{DK,mess}$  der Saugrohrdruck  $p_{S,mess}$  und das Luft/ Kraftstoff-Verhältnis  $\lambda_{mess}$ . Sowohl mit Hilfe des Prozessmodells ( $\hat{m}_L$ ) als auch über den Saugrohrdrucksensor ( $m_L$ ) wird eine Aussage über die Luftmasse in den Zylindern generiert. Die Differenz wird über ein PI-Glied zur Korrektur der Sollwertvorgabe auf der einen und zur Korrektur innerhalb des Prozessmodells auf der anderen Seite genutzt ( $\dot{m}_{DK,Korr.}$ ). Mit Hilfe des Vergleichs von  $\lambda_{soll}$  und  $\lambda_{mess}$  wird ebenfalls über ein PI-Glied die Sollkraftstoffmenge entsprechend korrigiert ( $m_{K,Korr.}$ ).

## 2.2 Ausgangspunkt der Betrachtungen

Um die wesentlichen Aspekte in diesem Beitrag hervorzuheben werden einige Annahmen und Voraussetzungen getroffen. Im einzelnen lassen sich diese durch die folgenden Aussagen spezifizieren:

- Betrachtet wird ein Ottomotor, bei dem der Saugrohrdruck sowie die Position der Drosselklappe als Messung zur Verfügung stehen.
- Der Ottomotor läuft im Homogenbetrieb, d.h.  $\lambda_{soll} = 1$ .
- Für die hier betrachtete Situation wird vorausgesetzt, dass die externe Abgasrückführung gesperrt ist.

- Die Darstellung der Fehlerauswirkungen beschränkt sich auf den stationären Betriebszustand.
- Es tritt kein Fehler in der Drosselklappe bzw. deren Sensorik auf, wodurch  $\alpha_{DK,soll} = \alpha_{DK,mess} = \alpha_{DK}$  gesetzt werden kann.

Insgesamt ist anzumerken, dass die im nachfolgenden betrachtete Systembeschreibung lediglich zur qualitativen Beschreibung der Struktur und damit als Interpretationsgrundlage für die Integration einer Diagnose dient. Es wird daher kein Anspruch auf Vollständigkeit des hier vorgestellten Modells erhoben, insbesondere der in der Praxis durch die zu erreichende Genauigkeit vorhandene hohe Detaillierungsgrad wurde bewusst vernachlässigt.

### 3 Integration der Fehlerdiagnose

In diesem Abschnitt wird das zuvor skizzierte Regelungssystem auf Basis der in Abschnitt 2.2 gemachten Anmerkungen aus Sicht der Diagnose betrachtet. Dazu wird zunächst eine erneute, die Situation darstellende und an die Sichtweise der Diagnose angepasste Systembeschreibung vorgenommen. Ferner erfolgt die Identifizierung der für die Fehlerdiagnose relevanten Systemstrukturen und Signale (Residuen).

Die dabei vorgenommene Umstrukturierung bzw. Interpretation des Systems ist in Abbildung 2 veranschaulicht. Die identifizierte Beobachterstruktur rund um das Prozessmodell kann als Basis für die Fehlerdiagnose eingesetzt werden. Dabei ist der in der Rückführung enthaltene Integralanteil entsprechend zu berücksichtigen, worin die hier verwendete Bezeichnung PI-Füllungsbeobachter begründet liegt. Wegen des vorhandenen Integralanteils kann bei der angestrebten stationären Betrachtungsweise der Fehlerauswirkungen nicht, wie normalerweise üblich, die Ausgangsschätzfehlergleichung des Beobachters mit

$$\bar{r}_1(s) = m_L - \hat{m}_L \quad (1)$$

als Residuum genutzt werden, da bei den betrachteten Fehlern stets ein Abgleich zwischen  $m_L$  und  $\hat{m}_L$  stattfindet ( $m_L - \hat{m}_L = 0$ ). Deutet man das PI-Glied jedoch als einen Nachfilter des Residuengenerators, so lässt sich das Residuum in der Form

$$r_1(s) = \frac{k_{p1}s + k_{I1}}{s} (m_L - \hat{m}_L) \quad (2)$$

bilden. Damit wird die Fehlerdetektierbarkeit im stationären Zustand gewährleistet (vgl. Abschnitt 4). Bei der Struktur des  $\lambda$ -Regelkreises handelt es sich um eine einfache PI-Regelungsstruktur. Für die Definition eines weiteren Residuums ( $r_2$ ) auf Basis dieser Struktur muss zunächst vorausgesetzt werden, dass die Schätzung der Kraftstoffmenge  $\hat{m}_K$  ausreichend genau ist, d.h. keine größeren stationären Abweichungen für  $\lambda$  ( $\lambda \approx \lambda_{soll}$ ) im fehlerfreien Fall entstehen. Diese Bedingung ist jedoch näherungsweise durch die vorhandene Modellgenauigkeit erfüllt, sodass der Ausgang der  $\lambda$ -Regelung ebenfalls durch

$$r_2(s) = \frac{k_{p2}s + k_{I2}}{s} (\lambda - \lambda_{soll}) \quad (3)$$

als ein Residuum betrachtet werden kann.

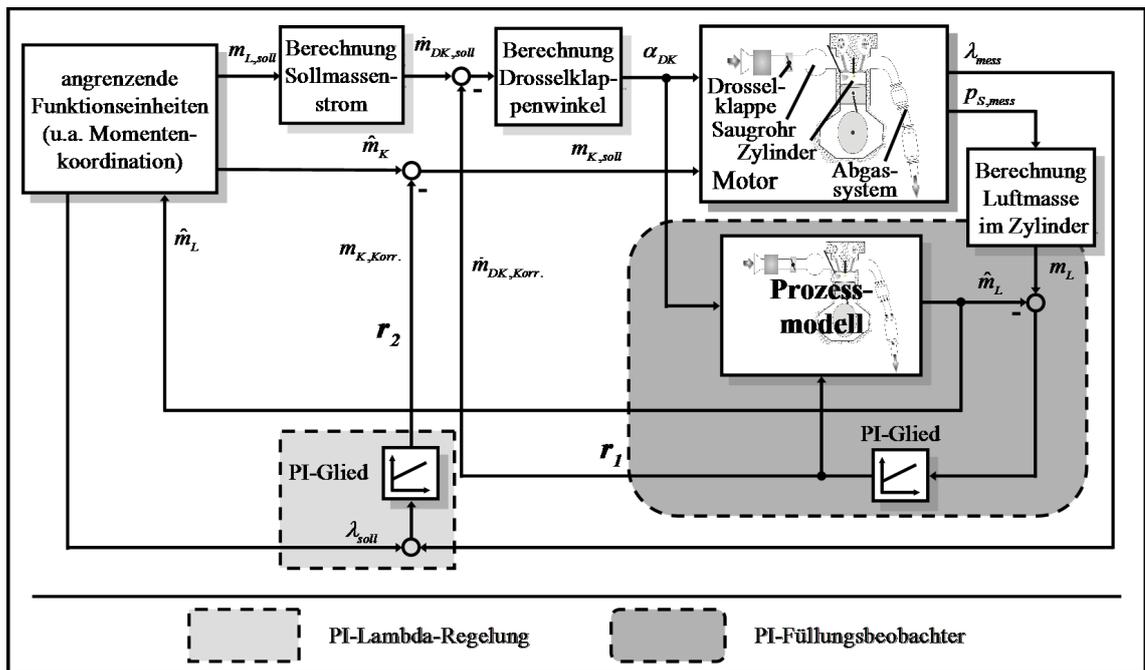


Abbildung 2: Integration der Fehlerdiagnose

**Anmerkung:** Für die hier vorgenommene Betrachtung des Systems im Frequenzbereich wird vorausgesetzt, dass das in seiner Natur nichtlineare System um jeden Arbeitspunkt linearisierbar ist.

Auf Grundlage der Systemstruktur können darüber hinaus die Einflüsse möglicher Fehler mit Hilfe von physikalischen Fehlermodellen beschrieben werden und für eine Fehlerentdeckung und -isolierung herangezogen werden. Die Überprüfung der Eignung der gewählten Residuen sowie eine Verifizierung der Fehlermodelle wird im nachfolgenden Abschnitt mit Hilfe von Motorprüfstandsdaten dargestellt.

## 4 Verifikation mit realen Daten

Zur Verifikation der theoretisch ermittelten und physikalisch basierten Fehlermodelle werden im Folgenden nun die Auswirkungen der mit hoher Priorität zu entdeckenden Fehler

- Saugrohrdrucksensorfehler (*DSSF*) durch einen Offset von  $\Delta p_S \approx 50 \text{ mbar}$ ,
- Leck im Saugrohr (*LS*) mit einem Durchmesser von ca.  $D_{Leck} \approx 3 \text{ mm}$ ,
- Verschmutzung im Bereich der Drosselklappe (*DKV*) und
- $\lambda$ -Sondenfehler (*LSF*) durch einen Offset von  $\Delta \lambda \approx 0.1$

auf die Residuen untersucht. Ziel ist es, die generierten Fehlermodelle anhand von realen Daten auf ihre Genauigkeit und Leistungsfähigkeit hin zu überprüfen. Dabei wird vorausgesetzt, dass nicht mehrere Fehler gleichzeitig auftreten. Für die Verifizierung werden die sich am Prüfstand einstellenden Werte in den stationären Arbeitspunkten mit den durch

die Fehlermodelle berechneten Verläufen für die unterschiedlichen Fehlersituationen verglichen. Zur Untersuchung der Einflüsse wurde ein Raster von jeweils 18 Arbeitspunkten im unteren und mittleren Lastbereich gewählt.

Bei der Betrachtung der Fehlerauswirkungen ist eine Abweichung der Form

$$\lim_{t \rightarrow \infty} r_1 \Big|_{OF} := \Delta M_F \approx -2,5 \frac{kg}{h} \quad (4)$$

im Residuum  $r_1$  zu berücksichtigen (vgl. Abbildungen). Aufgrund von Modellungenauigkeiten im Füllungsbeobachter kommt es darüber hinaus zu einer kontinuierlichen Abweichung am  $\lambda$ -Reglerausgang und damit in  $r_2$ . Diese Abweichungen werden mit

$$\lim_{t \rightarrow \infty} r_2 \Big|_{OF} := \Delta M_\lambda \approx 0,01 \quad (5)$$

angenommen (vgl. Abbildungen). Die insgesamt auf Basis dieser zusätzlichen Anpassungen ermittelten Ergebnisse sind in den Abbildungen 3 bis 6 dargestellt.

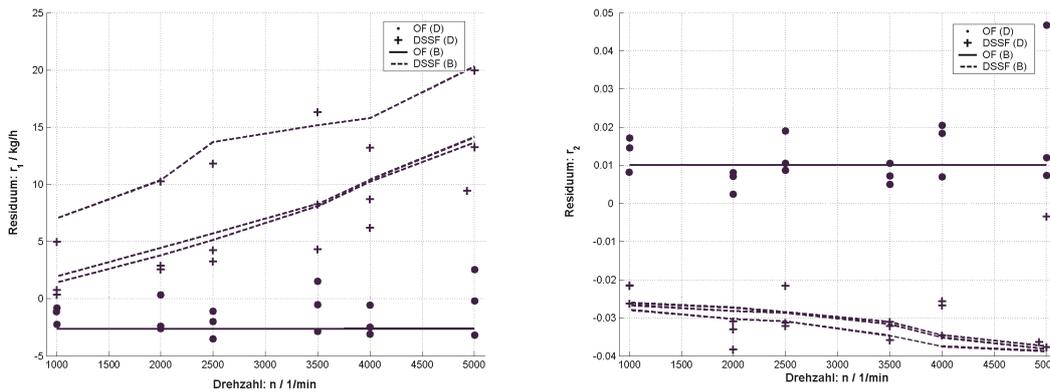


Abbildung 3: Residuen bei Auftritt eines Saugrohrdrucksensorfehlers

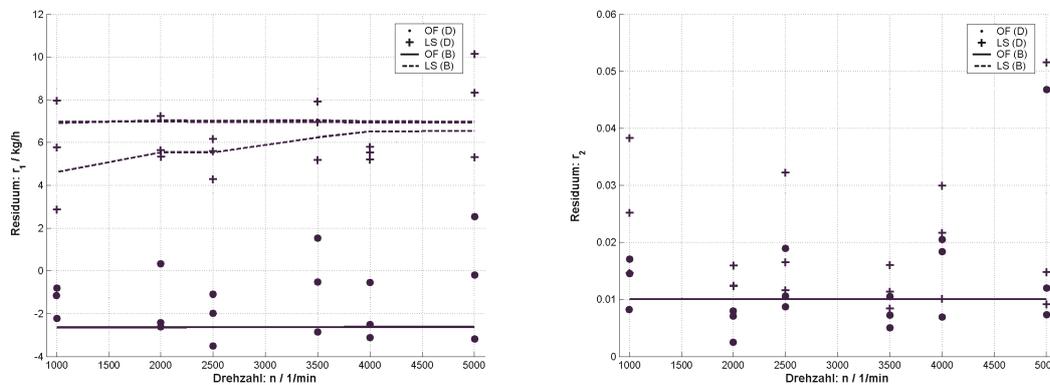


Abbildung 4: Residuen bei Auftritt eines Lecks im Saugrohr

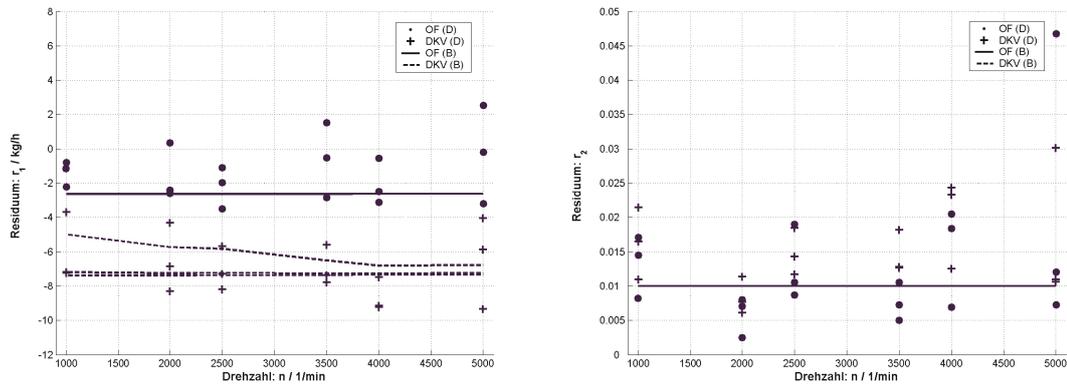


Abbildung 5: Residuen bei Auftritt einer Verschmutzung im Bereich der Drosselklappe

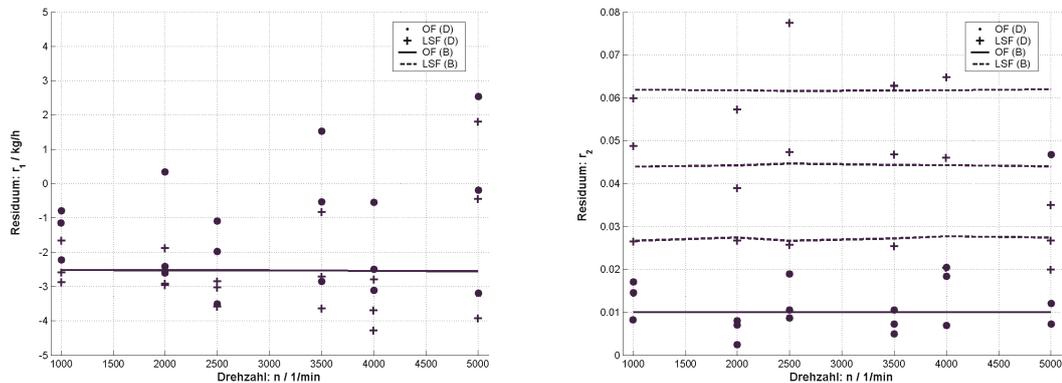


Abbildung 6: Residuen bei Auftritt eines  $\lambda$ -Sondenfehlers

Zur Unterscheidung kennzeichnen

- die Punkte die Prüfstandsdaten (D) des fehlerfreien Falls in den Arbeitspunkten,
- die Kreuze die Prüfstandsdaten (D) des jeweiligen Fehlerfalls in den Arbeitspunkten,
- die durchgezogenen Linien die berechneten Verläufe (B) im fehlerfreien Fall und
- die gestrichelten Linien die berechneten Verläufe (B) im jeweiligen Fehlerfall.

Die in den Abbildungen 3 bis 6 dargestellten guten Übereinstimmungen zwischen den theoretisch und aus Prüfstandsdaten ermittelten Fehlerauswirkungen bestätigen die gute Eignung der gewählten Residuen und die Genauigkeit der generierten Fehlermodelle. Dabei ist festzustellen, dass die unterschiedlichen Fehler jeweils zu deutlichen Ausschlägen in mindestens einem Residuum führen, wodurch zunächst eine Detektierbarkeit sichergestellt ist. Ferner erlauben die durch die Nichtlinearitäten entstehenden unterschiedlichen Auswirkungen der verschiedenen Fehler im weiteren Verlauf unter den hier betrachteten Gesichtspunkten eine eindeutige Fehlerisolierung.

## 5 Zusammenfassung und Ausblick

In dem hier vorgestellten Beitrag wurde die Integration von Fehlerdiagnose in eine bereits bestehende modellgestützte Regelkreisstruktur diskutiert. Ausgehend von der Problematik, dass aufgrund der hohen Komplexität der Modelle und der nur begrenzt zur Verfügung stehenden Systemleistung die Verwendung von analytischen modellgestützten Diagnoseverfahren praktisch ausgeschlossen ist, wurde ein alternativer Ansatz gewählt, der auf die bereits im System generierten Informationen zurückgreift. Die im Abschnitt 4 vorgestellten Ergebnisse zeigen gute Ansätze zur Fehlerdetektion für den Fall, dass sich der Motor in einem stationären Arbeitspunkt befindet. Ferner sind die durch die Fehlermodelle physikalisch nachgebildeten nichtlinearen Auswirkungen der Fehler auf die betrachteten Residuen bei einer späteren Residuenevaluierung durchzuführenden Fehlerunterscheidung, d.h. Fehlerisolierung, von großer Bedeutung. Die bei den Untersuchungen aufgedeckten Modellungenauigkeiten können im Rahmen der Residuenevaluierung durch entsprechende robuste Verfahren berücksichtigt bzw. kompensiert werden.

Insgesamt konnte mit dem hier vorgestellten Entwurf durch eine ganzheitliche Betrachtung des Systems eine in ihrer Form effiziente, ressourcenextensive und fehlerisolierende Diagnose in Ansätzen entwickelt werden.

## Nomenklatur

### verwendete Abkürzungen

$B$	Berechnung	$LSF$	$\lambda$ -Sondenfehler
$D$	Prüfstandsdaten	$MW$	Messwert
$DK(V)$	Drosselklappe(nverschmutzung)	$SW$	Sollwert
$DSSF$	Saugrohrdrucksensorfehler	$S$	Schätzung
$LS$	Leck im Saugrohr		

### Symbolverzeichnis

$\alpha_{DK}$	DK-Winkel (allg.)	$\Delta M_\lambda$	Auswirkungen der Modellungenauigkeiten im Füllungsbeobachter auf die $\lambda$ -Regelung
$\alpha_{DK,mess}$	DK-Winkel (MW)		
$\alpha_{DK,soll}$	DK-Winkel (SW)		
$\lambda$	Luft/Kraftstoff-Verhältnis	$\dot{m}_{DK,soll}$	Luftmassenstrom über DK (SW)
$\lambda_{mess}$	Luft/Kraftstoff-Verh. (MW)		
$\lambda_{soll}$	Luft/Kraftstoff-Verh. (SW)	$\dot{m}_{DK,Korr.}$	Luftmassenstromkorrektur durch Füllungsregler
$\Delta\lambda$	Luft/Kraftstoff-Verh. (MW-Offset)	$m_{L,soll}$	Luftmasse im Zylinder (SW)
$D_{Leck}$	Durchmesser des Lecks	$m_L$	Luftmasse im Zylinder (aus $p_{S,mess}$ )
$K_{In}$	Integralkonstanten PI-Glieder	$\hat{m}_L$	Luftmasse im Zylinder (S)
$K_{pn}$	Verstärkungsfaktoren PI-Glieder	$m_{K,soll}$	einzuspritzende Kraftstoffmasse (SW)
$\Delta M_F$	Füllungskorrektur im fehlerfreien Fall		

$m_{K,Korr.}$	Kraftstoffmengenkorrektur durch $\lambda$ -Regler	$p_{S,mess}$	Saugrohrdruck (MW)
$\hat{m}_K$	einzuspritzende Kraftstoffmasse (S)	$\Delta p_S$	Saugrohrdruck (MW-Offset)
		$r_i$	i-tes Residuensignal

## Literatur

- [1] Robert Bosch GmbH (Hrsg.). *Ottomotor-Management: Systeme und Komponenten*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 2. überarbeitete Auflage, 2003.
- [2] E. Hendricks and S.C. Sorenson. Mean value modelling of spark ignition engines. *SAE Technical Paper No. 900616*, 1990.
- [3] P. Andersson, L. Eriksson, and L. Nielsen. Modeling and architecture examples of model based engine control. *Linköping University, Sweden*, 1999.
- [4] E. Hendricks and J. B. Luther. Model and observer based control of internal combustion engines. *2nd International Workshop on modelling, emissions and control of automotive engines (MECA02), University of Salerno, Italy*, 2002.
- [5] R. W. Weeks and J. J. Moskwa. Transient air flow rate estimation in a natural gas engine using a nonlinear observer. *SAE Technical Paper No. 940759*, 1994.
- [6] J. Gerhardt, H. Hönninger, and H. Bischof. A new approach to functional and software structure for engine management systems - bosch me7. *Society of Automotive Engineers*, 1998.



# **An Investigation of Neural Networks in Fault Diagnosis of an Automotive Engine Air Path**

M. S. Sangha, J. B. Gomm, D. L. Yu, G. F. Page

*Control System Research Group, School of Engineering,  
Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK  
Email: j.b.gomm@livjm.ac.uk*

**Abstract:** The main objective of this research is to provide an efficient and reliable means for the diagnosis of faults in systems modelled by nonlinear differential equations, with a specific reference to the air path of an automobile engine. This involves the application of algorithms to data obtained from engine systems in order to detect component faults and sensor degradation. For this purpose a generic SI mean value engine model (MVEM) is used with some improvements and modifications. The faults considered in this study are air leakage in the intake manifold, exhaust gas recycle (EGR) valve stuck in different positions, intake-manifold pressure and temperature sensor faults. Neural networks are trained statically and tested for the same throttle angle data with zero percent misclassification achieved. A further neural network is trained and tested dynamically with uniformly distributed throttle angle inputs to investigate the performance on transient fault data over a wider range of throttle angles.

## **1 Introduction**

Electronic engine control was introduced in the 1970s, mainly as a result of two government requirements. The first was legislation to reduce and regulate automobile exhaust emissions and the second was the desire to improve the national average fuel economy. Evidently, the same factors are the main reasons for more recent legislation for on-board engine diagnostic capability. The Environmental Protection Agency (EPA), as well as the California Air Resource Board (CARB) mandated “On Board Diagnosis II” (OBD-II) for all light duty vehicles sold in American fleet starting from 1996 (*Tan and Saif, 2000*). All diesel cars sold in the EU must have an OBD system from year 2003 (*Nyberg, 2004*). The same is expected for gasoline cars and heavy trucks, etc. OBD-II requires continuous monitoring and fault detection capability for all vehicle components whose failures can result in emission levels beyond 1.5 times of the Federal Test Procedure (FTP) standards. This means that the OBD-II system should trigger and detect the deterioration of any component that results in a FTP

exhaust emission increase of 0.075 g/mi of HC, 1.7 g/mi of CO, and 0.2 g/mi of NOx. Once a failure has been detected, a malfunction indicator light (MIL) is to be illuminated to inform the driver of a problem, and a diagnostic trouble code (DTC) is to be stored in the on board memory.

The analytical or model based approach, among the few approaches to diagnostics in complex systems, has become a subject of tremendous research in recent years. A variety of diagnosis methods have been proposed under the umbrella of model-based techniques. The feature of all these techniques is that some form of mathematical knowledge of the process of interest along with inputs and outputs are used to generate superfluous information about that process. This redundant information is then used in a diagnostic process to arrive at decisions regarding fault or no-fault conditions. The development of simplified models of automotive engines is appropriate for control/diagnosis system research and design. Commonly, engine models are developed based on the engine's physical characteristics and some steady-state engine data (*Cook and Powell, 1993*). Such models consist of non-linear differential equations, table look-ups, regression-based functions and empirical formulae. It is important to note that the models derived from empirical data or physical phenomena are generally constructed under idealised situations, as such, they may not completely represent the practical dynamics or non-linear behaviour of engines. In this research neural network (NN) fault diagnosis techniques are studied (*Gomm et al., 2000*).

All IC engines contain significant nonlinearities, which dominate their dynamic behaviour. The MVEM is a fairly good approximation of medium speed IC engine dynamics. This model includes the latest results and efficiency enhancement system such as exhaust gas recycle (EGR) unit. A detailed derivation of all the sub-models of the MVEM can be found in *Hendricks, et al. (1996)* and *(1993)*. Only relevant sub-models with minor mathematical details are discussed in this paper because the main focus of this research is on Fault Detection & Isolation (FDI) and not on the MVEM simulation. The simulation is run for no fault and all fault conditions one by one for different throttle angle positions. Data is collected and normalised. Some noise is also introduced in the data to make it more realistic. Multilayer perceptron (MLP) and radial basis function (RBF) neural networks are trained to learn the non-linear behaviour of the MVEM with an appropriate level of error threshold. Test data samples are generated with faults for different throttle angle. The NN is found to correctly classify the faults as desired.

## 2 Mean Value Engine Model (MVEM)

Manifold filling dynamics, crankshaft speed equation and Matlab/Simulink simulation of the adiabatic engine system are described in this section.

### 2.1 Manifold Filling Dynamics

Assuming the intake manifold temperature to be constant and equal to the ambient and EGR temperature, the manifold filling state equation can be given as

$$\dot{P}_i = \frac{RT_i}{V_i} (-\dot{m}_{at} + \dot{m}_{ap} + \dot{m}_{EGR}) \quad (1)$$

This isothermal assumption is a big approximation and may not lead to good results because the EGR temperature can easily approach up to 200 degree centigrade and the quantity of EGR flow can be as high as 25% of the total flow. The manifold filling dynamics in reality is based on an adiabatic operation rather than isothermal. For the best accuracy the above equation is modified as follows:

$$\dot{P}_i = \frac{\kappa R}{V_i} (-\dot{m}_{ap} T_i + \dot{m}_{at} T_a + \dot{m}_{EGR} T_{EGR}) \quad (2)$$

The only problem with adiabatic assumption is that the intake manifold temperature must be known accurately and instantaneously, whereas the traditional temperature transducers have a time constant of three seconds. Using the law of energy conservation, a state equation which describes the time development of the intake manifold temperature can be given as:

$$\dot{T}_i = \frac{RT_i}{p_i V_i} [-\dot{m}_{ap} (\kappa - 1) T_i + \dot{m}_{at} (\kappa T_a - T_i) + \dot{m}_{EGR} (\kappa T_{EGR} - T_i)] \quad (3)$$

## 2.2 Crank Shaft Speed State Equation

Applying the law of conservation of rotational energy, the crank-shaft dynamics of an SI engine MVEM is described by the following equation:

$$\dot{n} = -\frac{1}{I_n} (P_f(p_i, n) + (P_p(p_i, n) + P_b(n))) + \frac{1}{I_n} H_u \eta_i(p_i, n, \lambda) \dot{m}_f(t - \Delta\tau_d) \quad (4)$$

## 2.3 Simulation of Adiabatic System

A Simulink block diagram of the adiabatic system is shown in Fig.1. The model has throttle angle as the only input and manifold temperature, pressure and crankshaft speed as the outputs. The simulation is run for 6 seconds for every fault with solver options set to variable step for ODE45.

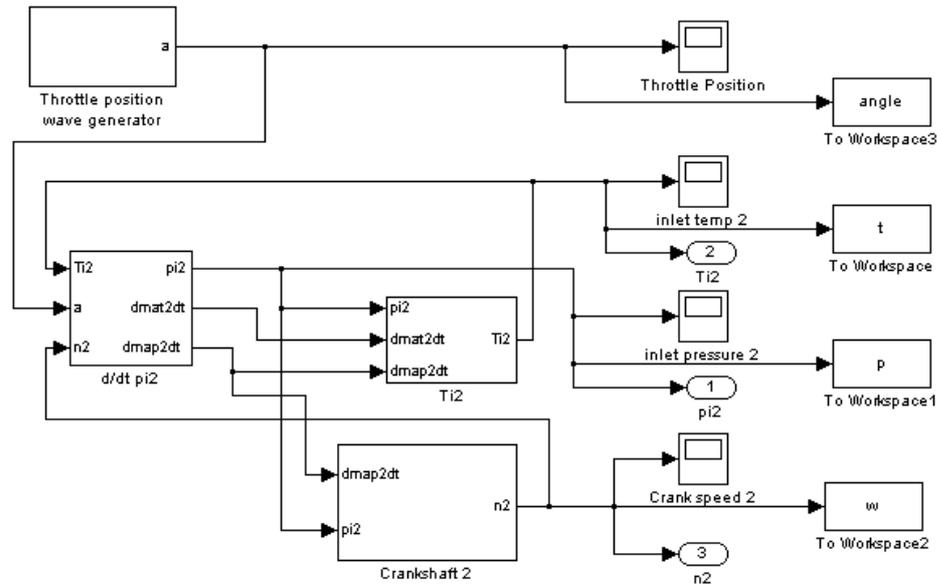


Figure 1: Simulink Model

### 3 Fault Simulations

Four faults are simulated. A leakage in the intake manifold fault and EGR valve stuck in different positions are simulated in the Simulink model of the engine, whereas sensor faults are calculated from normal temperature and pressure values of the intake manifold. The four faults with four different intensities are simulated one at a time in the model. All three outputs, i.e. pressure, temperature and crankshaft speed, are recorded for different faults.

#### 3.1 No Fault

For no fault situation, EGR is assumed 1/6 (16.67%) of the total air mass flow in the intake manifold. It is also assumed that all the sensors are working well and there is no leakage in the intake manifold. To illustrate the responses, the simulation is run for 6 seconds for a throttle angle input of 30 degrees. All three outputs are recorded and reach steady state in six seconds or less. The input and output responses are shown in Fig. 2.

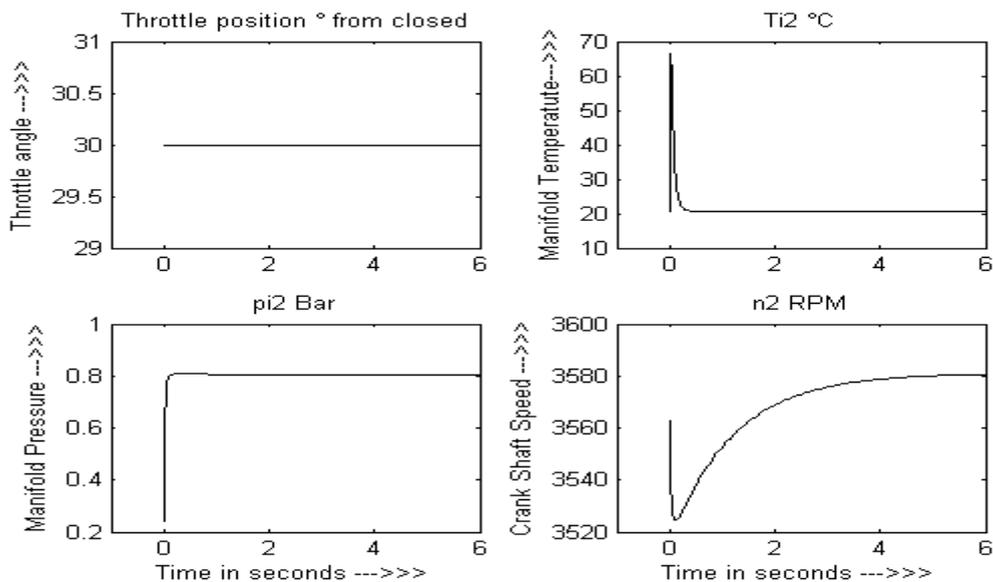


Figure 2: No Fault Response Curves

#### 3.2 Air Leakage Fault

The simulation is modified to introduce a constant  $k$  which increases the outflow of the intake manifold, treated as an air leakage. The air leakage is decided as a percentage (5%, 10%, 15% and 20%) of the total air intake in the intake manifold. The fault is implemented using a multiplying factor  $k = 0.05 - 0.2$ , which decides the intensity of the air leakage fault.

### 3.3 EGR Valve Faults

The normal value of EGR is kept as 1/6 of the total air mass flow, i.e. 16.67%. In an actual car running on road, the EGR can be as high as 20% of the total air mass flow in the intake manifold. Thus, a very realistic value of EGR feedback is chosen for the experiments. The value of  $d_{megrdt}$  for different fault intensities is regulated as 0%, 25%, 50%, 75% and 100% of the total EGR air mass flow. Where 0% corresponds to the EGR valve stuck in the closed position and 100% corresponds to full EGR air mass flow, i.e. no fault condition.

### 3.4 Temperature/Pressure Sensor Faults

Temperature and pressure sensor faults are considered in four different intensities: sensors over-reading and under-reading by 10% or 20% of the normal value. Fault data for the sensors is generated using multiplying factors (MF) of 1.2, 1.1, 0.9 and 0.8 for over-reading 20%, 10% and under-reading 10%, 20% respectively.

Fault data for 20°, 30° and 40° throttle angles is generated for all the 17 conditions including no fault condition. The nominal fault data for 30° throttle angle is given in table 1.

**Table 1: Nominal fault data for 30° throttle angle**

S. N.	Fault Name	Pressure Bar	Temperature Kelvin	Speed kRPM	Comments
1	No Fault	0.8038	293.79	3.5810	
2	Leakage 5%	0.7913	299.61	3.4947	
3	Leakage 10%	0.7781	305.45	3.4040	
4	Leakage 15%	0.7642	311.29	3.3082	
5	Leakage 20%	0.7497	317.16	3.2065	
6	EGR stuck 25% closed	0.7855	293.60	3.6299	
7	EGR stuck 50% closed	0.7671	293.40	3.6742	
8	EGR stuck 75% closed	0.7486	293.20	3.7144	
9	EGR stuck 100% closed	0.7299	293.00	3.7509	
10	Temp. sensor 20% over	0.8038	352.55	3.5810	MF=1.2
11	Temp. sensor 10% over	0.8038	323.17	3.5810	MF=1.1
12	Temp. sensor 10% under	0.8038	264.41	3.5810	MF=0.9
13	Temp. sensor 20% under	0.8038	235.03	3.5810	MF=0.8
14	Pressure sensor 10% over	0.9646	293.79	3.5810	MF=1.2
15	Pressure sensor 20% over	0.8841	293.79	3.5810	MF=1.1
16	Pressure sensor 20% under	0.7234	293.79	3.5810	MF=0.9
17	Pressure sensor 10% under	0.6430	293.79	3.5810	MF=0.8

### 4 Normalisation of data

Data normalisation is a must before inputting it to a neural network for learning. It is obvious from the data that if data is used without normalisation then the temperature

data will dominate the learning because of high numerical values. Moreover, the graphical data analysis is difficult without normalisation due to huge differences in the numerical data values. One way to normalise data for fault detection and diagnosis is to find deviations from the normal steady state values,  $Deviation = (x - x_{ss}) / x_{ss}$ .

The maximum deviation for temperature after normalisation is -20 to +20 with respect to no fault temperature, i.e. 293.79 Kelvin. In the same way normalised data for pressure and speed is calculated for different throttle angles and a 3D graph is shown in Fig. 3 for fault trend study with respect to changes in throttle angle input.

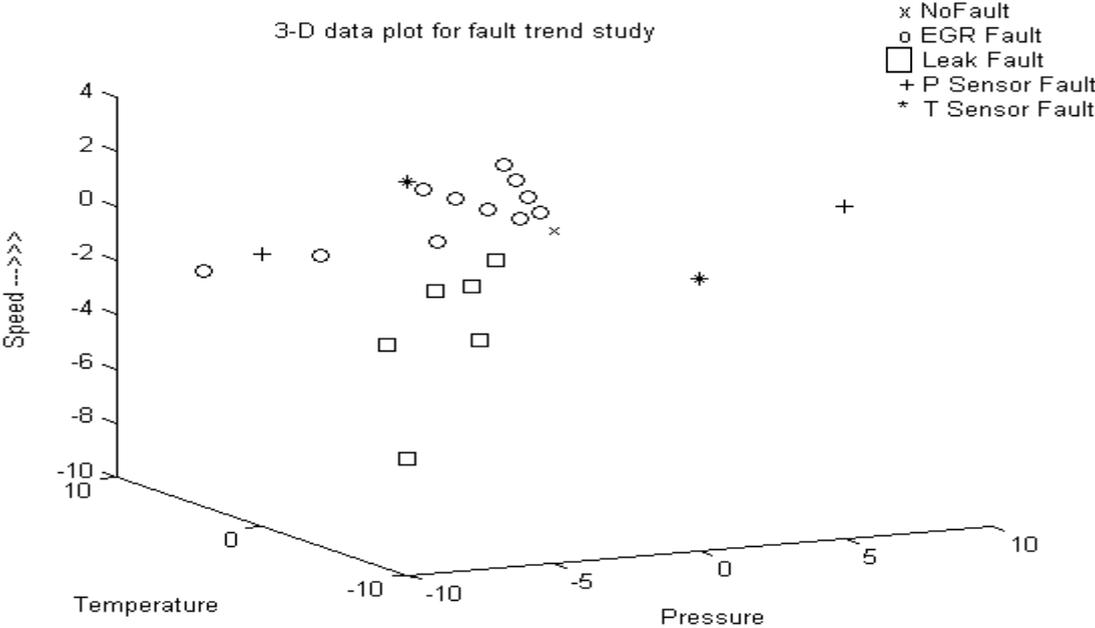


Figure 3: Nominal Fault Data 3D-Graph for different throttle angles

**5 Noise Introduction in Data**

In order to make the data more realistic, a small amount of noise is included in the data. One hundred normally distributed random numbers with an appropriate multiplier are added to each data point. Thus, 1700 noisy data points are generated for the 17 normal and fault data points for each different throttle angle. Fifty percent of the data is used for training and the remaining fifty percent is used for testing of the neural networks.

**6 Training & Testing of Neural Networks**

Both MLP and RBF networks have been trained and tested on different sets of data for the same throttle angle. The MATLAB Neural Network Toolbox is used for training all networks. Subsequently, a RBF network is trained for equally distributed throttle angles (virtually trained for the whole range of throttle angle) and tested for any particular angle for all the 17 faults.

## 6.1 Training & Testing an MLP

In order to train an MLP for fault diagnosis and isolation (FDI), a user-friendly graphical user interface to the MATLAB Neural Network Toolbox, developed by Dr. Barry Gomm, is used. Fifty percent fault data (850 data points) are stored in a matrix (xi) and placed in the Matlab workspace. The fault data 'xi' (850x3) is graphically shown in Fig. 4 and is the network input data.

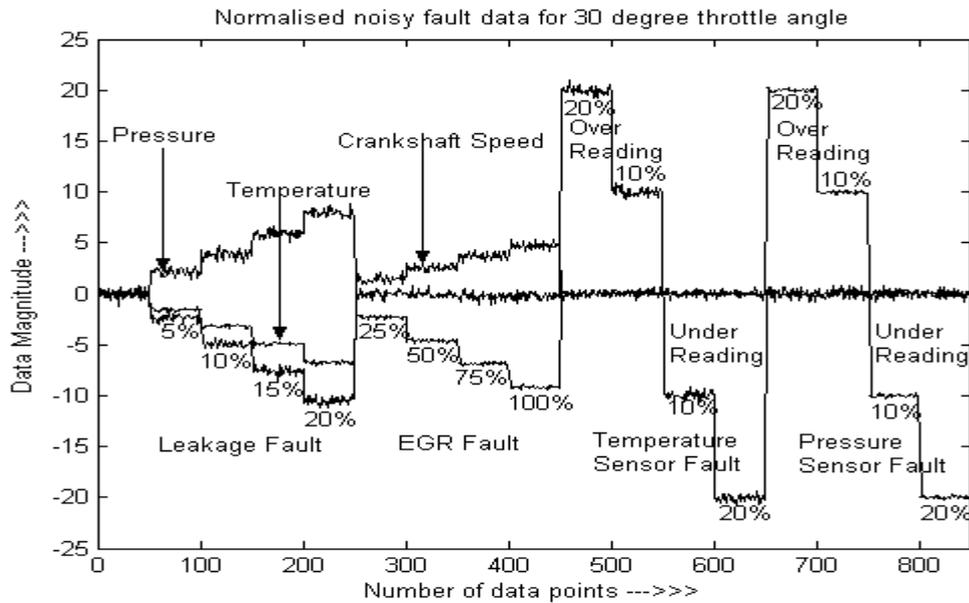


Figure 4: Normalised Noisy Fault Data

The target values are tabulated in Microsoft Excel and imported into the Matlab workspace in a matrix (xo). The target output data to the network, matrix 'xo' (850x17), is defined as follows:

Rows	xo
1~50	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
50~100	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
101~150	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
...	...
801~850	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

The target matrix xo has ones in the first column up to the 50<sup>th</sup> row, the second column has ones from the 51<sup>st</sup> row to the 100<sup>th</sup> row and so on. The last column has ones from the 801<sup>st</sup> row to the 850<sup>th</sup> row. When one column corresponding to one particular error has ones in it for a range, then rest all the columns hold zeros in it for that range as shown above. The MLP has three inputs and seventeen outputs. Tanh function is used for the hidden layer and linear for the output neurons. The optimal number of hidden nodes is experimentally found to be 20. The data scaling is kept in the range of -0.8 to +0.8. Levenberg-Marquardt algorithm is used for training with an error threshold level of 100 initially. The training takes only 32 epochs to reach the required error level. The trained network is tested on another set of data and excellent results are achieved. The

first five columns and some rows of the network output matrix ynn (850x17) on the test data are:

Row	Network outputs				
1	<b>0.77787</b>	0.10578	-0.0083665	-0.026224	0.0075598
51	0.087105	<b>0.97885</b>	-0.026192	0.0034296	0.0007112
101	0.0004388	0.0052001	<b>0.99036</b>	-0.0045401	-0.0059711
151	0.004207	0.0061615	0.0023197	<b>0.997</b>	-0.013213
201	-0.0089038	0.0099608	-0.0034402	0.0063744	<b>0.99195</b>

All the output values above +0.5 are interpreted as ones and below 0.5 as zeros. Not a single misclassification is found in the output matrix ynn. If the training error threshold is made very low, i.e. 0.0001 or even lower, then the result improves by a little but the training time increases significantly. It seems that for this classification problem, a training error threshold of 100 is appropriate. The MLP test result graph for the first two and last two faults is shown in Fig. 5.

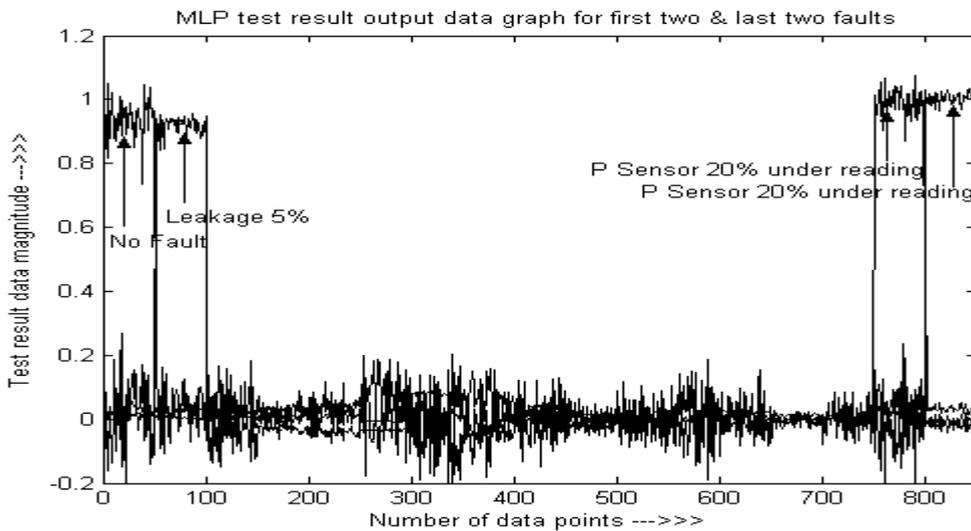


Figure 5: MLP Test Result Graph for first 2 and last 2 faults

## 6.2 Training & Testing an RBF network

The same inputs  $x_i$  and target outputs  $x_o$  are used for training the RBF neural network. A number of trials have been made with different spread constant and training error level and an error level of 1 with a spread constant of 2.4 is experimentally found optimal. With this value of spread constant, the network structure needs to have only 31 neurons to reach the error level.

The trained RBF network is then tested on another set of data for the same throttle angle. The first five columns and some rows of the resultant network output matrix are:

Row	Network outputs				
1	<b>0.93771</b>	-0.003213	0.0041629	0.0038185	-0.0013929
51	0.032872	<b>1.0064</b>	-0.041617	-0.0073498	0.0030937
101	-0.0036037	-0.029252	<b>1.0108</b>	0.020804	0.0037323

151	-0.0010543	-0.00010678	0.017346	<b>0.9108</b>	0.040882
201	-0.0022111	0.0021787	0.00643	0.00090494	<b>1.0102</b>

As before, all values above +0.5 are interpreted as 1 (one) and all values below +0.5 as 0 (zero). +0.5 is used as the output decision boundary. The RBF test results were similar to those shown in Fig.5 but closer to the target outputs of 1 and 0 than the MLP network. Excellent results have been achieved without a single misclassification.

## 7 Dynamic Training & Testing of Neural Network

In section 6 it was observed that if the neural network is trained and tested on the same throttle angle data (different data sets) then no misclassification occurs. In this section an RBF neural network is trained on dynamic fault data and tested on different throttle angle fault data. The RBF is trained for the whole range of throttle angles from 20° to 40° with an error threshold of 100 and spread constant of 0.2. The RBF is trained for ten different throttle angles for all the 17 fault conditions. Each fault is simulated for 6 seconds (20 data points) for each throttle angle. It creates a bulky input training data matrix of 4x3400 points. The throttle angle is also used as an input along with pressure, temperature and speed. Training takes as long as 15 minutes on a P4 computer and forms a network of 191 neurons.

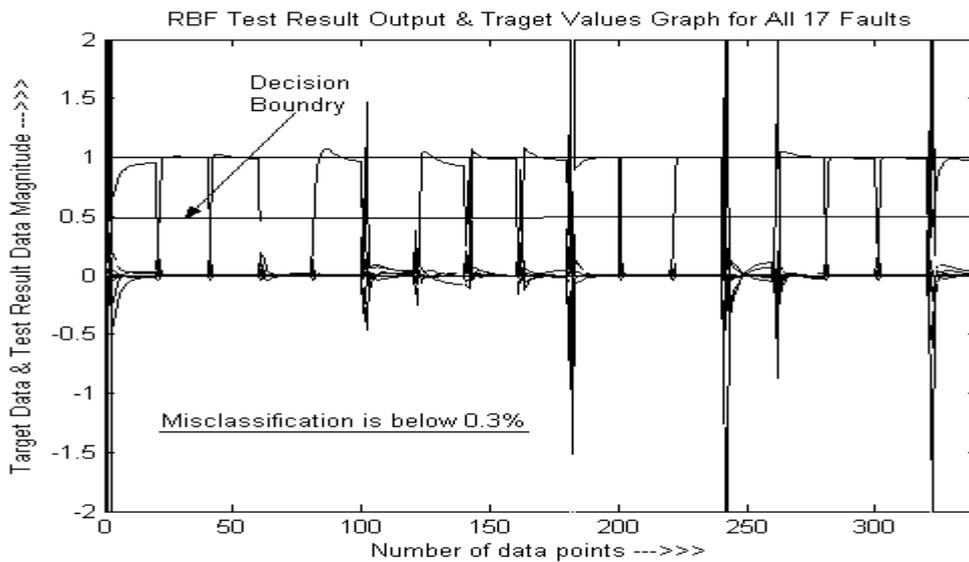


Figure 6: RBF Dynamic Test Results

Misclassification on test data is less than 0.3%, which can be further reduced by reducing the error threshold for the training data. However, a lower error threshold takes a much longer time to train and forms a huge network of neurons.

## 8 Conclusions

Static fault detection and isolation (FDI) for an engine simulation has successfully been demonstrated with zero percent misclassification. It implies that the neural

network learns the non-linearities of the system very well and thus the training is ideal especially in the case of RBF neural network because it results in less misclassification for the same error threshold.

The output response trends of the engine system for different throttle angle inputs are different and therefore, when a neural network was trained on different sets of throttle angle data then leakage and EGR faults were not classified properly. This problem is overcome by training a neural network dynamically with a range of equally distributed throttle angle inputs, i.e. training it virtually for all the range of the throttle angles, and then testing for a random input throttle angle.

There is a lot of scope for further research work in this subject, i.e. study of all the throttle angles with one neural network, dynamic FDI with real data, etc.

## Nomenclature

$t$	time (sec)	$n$	engine speed (rpm/1000)
$\dot{m}_f$	engine port fuel mass flow (kg/sec)	$T_a$	ambient temperature (Kelvin)
$P_i$	absolute manifold pressure (bar)	$T_i$	intake manifold temperature (Kelvin)
$\dot{m}_{at}$	air mass flow past throttle plate(kg/sec)	$T_{EGR}$	EGR temperature (Kelvin)
$\dot{m}_{ap}$	air mass flow into intake port (kg/sec)	$\dot{m}_{EGR}$	EGR mass flow (kg/sec)
$V_i$	manifold + port passage volume ( $m^3$ )	$R$	gas constant (here $287 \times 10^{-5}$ )
$\kappa$	ratio of specific heats = 1.4 for air	$I$	crank shaft load inertia ( $kg\ m^2$ )
$P_f$	friction power (kW)	$P_b$	load power (kW)
$P_p$	pumping power (kW)	$H_u$	fuel lower heating value (kJ/kg)
$\Delta\tau_d$	injection torque delay time (sec)		

## REFERENCES

- Cook J.A., Powell B.K., (1993), *Modelling of an internal combustion engine for control analysis*, IEEE Control Systems, 13(3), pp.62-68.
- Gomm J.B., *et al.* (2000), *Diagnosis of process faults with neural networks and principal component analysis*, Proc ImechE Part E, 214, pp.131-143.
- Hendricks E., *et al.* (2000), *A Generic Mean Value Engine Model for Spark Ignition Engines*. Institute of Automation, DTU.
- Hendricks E., *et al.*, (1996) *Modelling of the Intake Manifold Filling Dynamics*, SAE Technical Paper No. 960037.
- Hendricks E., *et al.*, (1993) *Nonlinear Transient Fuel Film Compensation*, SAE Technical Paper No. 930767.
- Nyberg M. and Stutte T., (2004) *Model based diagnosis of the air path of an automotive diesel engine*, Control Engineering Practice.
- Tan Y. and Saif M., (2000) *Neural-networks-based nonlinear dynamic modelling for automotive engines*, Neurocomputing, 30, pp.129-142.

# Beobachterentwurf für Systeme mit veränderlichen Parametern

H.-G. Schulz <sup>\*</sup>, S. Schneider <sup>\*,†</sup>, S.X. Ding <sup>\*</sup>  
E. L. Ding <sup>†</sup>

28. Juli 2004

## Zusammenfassung

Im vorliegenden Beitrag werden Methoden zum Entwurf von Beobachtern für lineare parametervariante (LPV) Systeme vorgestellt, mit denen der Beobachterentwurf unabhängig vom veränderlichen Parameter ist. Dabei wird vorausgesetzt, dass der veränderliche Parameter messbar ist. Am Beispiel des Einspurmodells, welches die Querdynamik eines Fahrzeuges beschreibt und die Geschwindigkeit (in Längsrichtung) als veränderlichen Parameter beinhaltet, wird die Vorgehensweise verdeutlicht. Der Beobachterentwurf erfolgt im Hinblick auf die Diagnose von Sensorfehlern der Sensorik der Fahrzeugquerdynamik.

## 1 Einleitung

Systeme mit veränderlichen Parametern treten in zahlreichen Anwendungen der Regelungstechnik und der Fehlerdiagnose auf. Die Tatsache, dass veränderliche Parameter vorhanden sind, führt zu einem erhöhten Aufwand sowohl für den Entwurf einer Regelung als auch für den Beobachterentwurf. Ist der veränderliche Parameter messbar, so existieren verschiedene Möglichkeiten den Entwurf zu erleichtern, was insbesondere für eine einfache und schnelle Applikation vorteilhaft ist. In diesem Beitrag werden verschiedene Verfahren für einen vereinfachten Entwurf von Beobachtern für Systeme mit veränderlichen Parametern am Beispiel der Querdynamik eines Fahrzeuges vorgestellt.

## 2 Fahrzeugmodell

Für die Beschreibung der Querdynamik eines Kraftfahrzeuges wird meist das bekannte Einspurmodell [1] eingesetzt, da dieses trotz seiner Einfachheit (Modell 2. Ordnung) für moderate Querbeschleunigungen ( $a_y < 4\text{m/s}^2$ ) ausreichend genau ist.

---

<sup>\*</sup>Fachgebiet Automatisierungstechnik und komplexe Systeme, Fakultät 5, Universität Duisburg-Essen, 47048 Duisburg

<sup>†</sup>Fachbereich Physikalische Technik, Fachhochschule Gelsenkirchen, 45877 Gelsenkirchen

Das um den stationären Einfluss des Wankens erweiterte Fahrzeugmodell (erweitertes Einspurmodell) kann durch das Zustandsraummodell (1), (2) dargestellt werden [2]. Als Messgrößen stehen der Lenkwinkel  $\delta_L$  (Einganggröße), die Querbeschleunigung  $a_y$  und die Gierrate  $r$  (Ausgangsgrößen) sowie die Geschwindigkeit  $v$  zur Verfügung. Die Geschwindigkeit ist also ein (messbarer) Parameter des Systems.

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{Y_\beta K_{\phi R}}{m} \frac{1}{v} & \frac{Y_r K_{\phi R}}{m} \frac{1}{v} - 1 \\ \frac{N_\beta}{I_z} & \frac{N_r}{I_z} \end{bmatrix}}_{=:A} \begin{bmatrix} \beta \\ r \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{Y_\delta K_{\phi R}}{m} \frac{1}{v} \\ \frac{N_\delta}{I_z} \end{bmatrix}}_{=:B} \delta_L^* + \underbrace{\begin{bmatrix} -\frac{g}{v} \\ 0 \end{bmatrix}}_{=:E_1} \sin \alpha_x \quad (1)$$

$$\begin{bmatrix} a_y \\ r \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{Y_\beta}{m} & \frac{Y_r}{m} \\ 0 & 1 \end{bmatrix}}_{=:C} \begin{bmatrix} \beta \\ r \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{Y_\delta}{m} \\ 0 \end{bmatrix}}_{=:D} \delta_L^* \quad (2)$$

mit:

$$\begin{aligned} Y_\beta &= -(c'_{\alpha V} + c_{\alpha H}), & Y_r &= \frac{1}{v} (l_H c_{\alpha H} - l_V c'_{\alpha V}), & Y_\delta &= c'_{\alpha V}, \\ N_\beta &= l_H c_{\alpha H} - l_V c'_{\alpha V}, & N_r &= -\frac{1}{v} (l_V^2 c'_{\alpha V} + l_H^2 c_{\alpha H}), & N_\delta &= l_V c'_{\alpha V}. \end{aligned}$$

In dem in (??), (2) beschriebenen Zustandsraum-Modell tritt die Geschwindigkeit  $v$  als Parameter in den Matrizen  $A$ ,  $B$  und  $C$  auf. Dies führt zu einem erhöhten Aufwand beim Beobachterentwurf, bei dem die Abhängigkeit von veränderlichen Parameter berücksichtigt werden muss.

Aus diesem Grund werden im folgenden Abschnitt Verfahren diskutiert, die einen von dem veränderlichen Parameter unabhängigen Beobachterentwurf für dieses System ermöglichen.

### 3 Behandlung der Parameter-Abhängigkeit

Es gibt verschiedene Möglichkeiten für den Entwurf eines Beobachters für ein System mit veränderlichen Parametern. Einige davon werden im folgenden kurz diskutiert. Es wird vorausgesetzt, dass der veränderliche Parameter ausreichend genau bekannt ist (z.B. durch Messung).

#### Gain Scheduling:

Beim *Gain Scheduling* wird für verschiedene (konstante) Werte des Parameters (Arbeitspunkte) jeweils ein Beobachter entworfen. Die Kunst besteht darin, in geeigneter Weise zwischen den Beobachtern umzuschalten. Die Stabilität des Beobachters mit *Gain Scheduling* ist i.d.R. unkritisch, aber nur schwierig theoretisch zu garantieren. *Gain Scheduling* ist ein geeigneter und in der Praxis vielfach genutzter Ansatz. Der Nachteil liegt im höheren Aufwand für den Entwurf mehrerer Beobachter und den Entwurf der Umschaltung zwischen den Beobachtern sowie im höheren Bedarf an Ressourcen.

#### Aufschaltung einer bekannten Größe:

Hierbei wird das System ebenfalls so modelliert, dass der veränderliche Parameter in einer (oder mehreren) Eingangsgröße(n) auftritt. Diese Eingangsgrößen sind aber bekannt

bzw. messbar und können daher auf den Beobachter aufgeschaltet werden (wie auch bei einer Störgrößenaufschaltung, wenn die Störgröße messbar ist). Die Beobachtermatrix  $L$  wird ohne Berücksichtigung der bekannten Eingangsgröße entworfen und ist daher nicht abhängig von dem veränderlichen Parameter. Dieser Ansatz ermöglicht auf einfache und natürliche Weise den vom Parameter unabhängigen Entwurf des Beobachters. Der Beobachter selbst ist natürlich - wegen der Aufschaltung der messbaren Eingangsgröße - vom Parameter abhängig.

Transformation auf Beobachtungsnormalform:

Mit der Transformation des Modells auf *Beobachtungsnormalform* wird dasselbe Ziel verfolgt wie bereits für die *Aufschaltung einer bekannten Größe* beschrieben. Die bekannte bzw. messbare, veränderliche Größe soll auf den Beobachter aufgeschaltet werden, um einen parameterunabhängigen Entwurf zu ermöglichen. Vorteil dieser Vorgehensweise ist, dass im Gegensatz zu der zuvor erwähnten Methode nur eine Messgröße auf den Beobachter aufgeschaltet werden muss. Diese Lösung bietet die Möglichkeit einer vollständigen Fehlerdiagnose, das heißt sowohl einer Fehlererkennung als auch einer Fehlerisolation.

## 4 Geschwindigkeitsunabhängiger Beobachterentwurf

In diesem Abschnitt werden explizit zwei Verfahren beschrieben, mit denen ein parametervariantes System in ein nicht parametervariantes System transformiert werden kann. Für den eigentlichen Entwurf des Beobachters können dann die bekannten Entwurfsverfahren verwendet werden. In Abschnitt 4.2 wird gezeigt, dass durch eine Transformation auf Beobachtungsnormalform ein Beobachter entworfen werden kann, bei dem nur eine Messgröße aufgeschaltet werden muss, um Unabhängigkeit von der Geschwindigkeit zu erreichen.

### 4.1 Aufschalten einer bekannten Größe

#### 4.1.1 Variante 1

Das System

$$\begin{aligned} \dot{x} &= A(v)x + B(v)u \\ y &= C(v)x + D(v)u \end{aligned} \quad (3)$$

mit den Parameter-abhängigen Matrizen  $A, B$  wird folgendermaßen umgeformt:

$$\begin{aligned} \dot{x} &= \bar{A}x + \bar{B}u + g_1(v, y) \\ y &= \bar{C}x + \bar{D}u + g_2(v, y) \end{aligned}$$

Die Matrizen  $\bar{A}, \bar{B}, \bar{C}, \bar{D}$  sind nun konstante Matrizen, unabhängig vom Parameter  $v$ . Die Matrizen  $\bar{B}, \bar{D}$  müssen nicht notwendigerweise parameterunabhängig sein, dies ergibt sich aber zufällig im unten aufgeführten Beispiel.  $g_i(v, y), i = 1, 2$  sind Funktionen des Parameters Geschwindigkeit und der Ausgangsgröße  $y$ .

Für den Beobachter werden die Gleichungen:

$$\begin{aligned}\dot{\hat{x}} &= \bar{A}\hat{x} + \bar{B}u + g_1(v_S, y_S) + L(y_S - \hat{y}) \\ \hat{y} &= \bar{C}\hat{x} + \bar{D}u + g_2(v_S, y_S)\end{aligned}$$

angesetzt. Dabei ist  $y_S$  die vom Sensor gemessene Ausgangsgröße und  $v_S$  ist der gemessene bzw. geschätzte Parameter Geschwindigkeit. Unter der Annahme, dass die Messung fehlerfrei ist ( $y = y_S$ ) und der Parameter  $v$  exakt bekannt ist (d.h.  $g_i(v_S, y_S) = g_i(v, y)$ ,  $i = 1, 2$ ), ist die Schätzfehler-Differentialgleichung gegeben durch:

$$\begin{aligned}\Delta x &= x - \hat{x} \\ \Delta \dot{x} &= (\bar{A} - L\bar{C}) \Delta x\end{aligned}$$

und das Residuum ist:

$$r = \Delta y = y - \hat{y} = \bar{C}\Delta x$$

In der Praxis wird die Messung jedoch fehlerbehaftet sein mit dem unbekanntem Messfehler  $f$ :

$$y_S = y + f$$

Vereinfachend sei angenommen, dass die Funktionen  $g_i(v, y)$ ,  $i = 1, 2$  linear in  $y$  sind (dies ist im hier betrachteten Beispiel der Fall, siehe Gl. (6), d.h.  $g_i(v_S, y_S) = G_i(v)y$ ,  $i = 1, 2$ ). Im Falle einer fehlerbehafteten Messung ist das Residuum:

$$\begin{aligned}r = y_S - \hat{y} &= \bar{C}\Delta x + (G_2(v) - G_2(v_S))y - G_2(v_S)f + f \\ &\approx \bar{C}\Delta x + (I - G_2(v_S))f\end{aligned}\quad (4)$$

und die Schätzfehler-Differentialgleichung berechnet sich zu:

$$\begin{aligned}\Delta \dot{x} &= \bar{A}\hat{x} + \bar{B}u + G_1(v)y - (\bar{A}\hat{x} + \bar{B}u + G_1(v_S)y_S + L(y_S - \hat{y})) \\ &= (\bar{A} - L\bar{C}) \Delta x - L(I - G_2(v_S))f - G_1(v_S)f + (G_1(v) - G_1(v_S))y \\ &\approx (\bar{A} - LC) \Delta x - (L(I - G_2(v_S)) + G_1(v_S))f\end{aligned}\quad (5)$$

Dabei ist jeweils angenommen, dass der Parameter Geschwindigkeit sehr genau bekannt ist, so dass  $G_i(v_S) = G_i(v)$ ,  $i = 1, 2$  und die entsprechenden Terme in (4) und (5) vernachlässigt werden können. Bei einer fehlerbehafteten Messung konvergiert der Schätzfehler also nicht gegen Null. Das Residuum ist also:

$$r = -\bar{C}(sI - \bar{A} + L\bar{C})^{-1}(L(I - G_2(v_S)) + G_1(v_S))f + (I - G_2(v_S))f$$

Der stationäre Wert des Residuums ist:

$$r = \left( \bar{C}(\bar{A} - L\bar{C})^{-1}(L(I - G_2(v_S)) + G_1(v_S)) + (I - G_2(v_S)) \right) f = R(v_S)f$$

### Anwendung auf das erweiterte Einspurmodell:

Das System Gl. (3) ohne Berücksichtigung der Fahrbahnneigung kann man z.B. wie folgt darstellen:

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ \frac{N_\beta}{I_z} & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{N_\delta}{I_z} \end{bmatrix} \delta_L^* + \begin{bmatrix} \frac{Y_\beta}{mv} K_{\phi_R} & \frac{Y_r}{mv} K_{\phi_R} & \frac{Y_\delta}{mv} K_{\phi_R} \\ 0 & \frac{N_r}{I_z} & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ \delta_L^* \end{bmatrix}$$

Mit (1.4) folgt:

$$a_y = \frac{Y_\beta}{m} \beta + \frac{Y_r}{m} r + \frac{Y_\delta}{m} \delta_L^* \quad \Rightarrow \quad \beta = \frac{m}{Y_\beta} a_y - \frac{Y_r}{Y_\beta} r - \frac{Y_\delta}{Y_\beta} \delta_L^*$$

und daraus:

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ \frac{N_\beta}{I_z} & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{N_\delta}{I_z} \end{bmatrix} \delta_L^* + \begin{bmatrix} \frac{K_{\phi_R}}{v} & 0 \\ 0 & \frac{N_r}{I_z} \end{bmatrix} \begin{bmatrix} a_y \\ r \end{bmatrix}$$

$$g_1(v, y) = \begin{bmatrix} \frac{K_{\phi_R}}{v} & 0 \\ 0 & \frac{N_r}{I_z} \end{bmatrix} \begin{bmatrix} a_y \\ r \end{bmatrix} \quad (6)$$

Die Ausgangsgleichung wird in entsprechender Weise umgeformt wie die Zustandsgleichung:

$$y = \bar{C}x + \bar{D}\delta_L + g_2(v, y) = \begin{bmatrix} \frac{Y_\beta}{m} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} \frac{Y_\delta}{m} \\ 0 \end{bmatrix} \delta_L^* + \begin{bmatrix} 0 & \frac{Y_r}{m} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_y \\ r \end{bmatrix}$$

Zahlreiche andere Darstellungen sind möglich. So kann z.B. die Abweichung um einen Arbeitspunkt aufgeschaltet werden etc..

#### **4.1.2 Variante 2**

In diesem Abschnitt wird der Beobachterentwurf mittels einer geeigneten zusätzlichen Rückführung beschrieben. Das zugrunde liegende Prinzip ist auch hier wieder die Aufschaltung einer bekannten Größe. Die vorgestellte Lösung ist speziell auf das Einspurmodell zugeschnitten.

Der wesentliche Unterschied zum vorangehenden Abschnitt ist, dass hier statt der Ausgangsgleichung (2) die Messgleichung (8) verwendet wird.

Ziel ist auch hier wiederum die Elimination der Geschwindigkeit aus der Zustandsgleichung des Beobachters. Dies geschieht nun durch Addition geeigneter Funktionen der Messgrößen, d.h. durch Rückführung der Messgrößen. Dabei wird diesmal von der Systembeschreibung in Descriptor-Form ausgegangen und die Fahrbahnneigung sei wieder vernachlässigt. Die Zustandsgleichung des Beobachters ist dann:

$$\begin{bmatrix} mv & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\hat{\beta}} \\ \dot{\hat{r}} \end{bmatrix} = \begin{bmatrix} Y_\beta K_{\phi_R} & Y_r K_{\phi_R} - mv \\ \frac{N_\beta}{I_z} & \frac{N_r}{I_z} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + \begin{bmatrix} Y_\delta K_{\phi_R} \\ \frac{N_\delta}{I_z} \end{bmatrix} \delta_L^* + L \begin{bmatrix} a_y - \hat{a}_y \\ r - \hat{r} \end{bmatrix} \quad (7)$$

Die Beobachter-Matrix  $L = L_1 + L_2$  wird zunächst gedanklich in zwei Teile aufgespalten. Mit  $L_1$  wird die Geschwindigkeitsabhängigkeit der Systemmatrix eliminiert. Der Parameter Geschwindigkeit tritt einmal direkt auf und ist auch in den Größen  $Y_r$  und  $N_r$  enthalten. Mit:

$$L_1 = \begin{bmatrix} -mK_{\phi_R} + \frac{K_{\phi_R}}{v} & Y_r K_{\phi_R} \\ 0 & \frac{N_r}{I_z} \end{bmatrix}$$

$$\hat{a}_y = \frac{v}{K_{\phi_R}} \left( \dot{\hat{\beta}} + \hat{r} \right) \quad (8)$$

wird (7) zu:

$$\begin{bmatrix} \dot{\hat{\beta}} \\ \dot{\hat{r}} \end{bmatrix} = \begin{bmatrix} Y_\beta K_{\phi_R} & -1 \\ \frac{N_\beta}{I_z} & 0 \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + \begin{bmatrix} Y_\delta K_{\phi_R} \\ \frac{N_\delta}{I_z} \end{bmatrix} \delta_L^* + \begin{bmatrix} -mK_{\phi_R} + \frac{K_{\phi_R}}{v} & Y_r K_{\phi_R} \\ 0 & \frac{N_r}{I_z} \end{bmatrix} \begin{bmatrix} a_y \\ r \end{bmatrix} + L_2 \begin{bmatrix} a_y - \hat{a}_y \\ r - \hat{r} \end{bmatrix} \quad (9)$$

Für dieses von der Geschwindigkeit unabhängige System kann man nun einen Beobachter durch geeignete Wahl von  $L_2$  entwerfen. Dazu ist jedoch wiederum die Ausgangsgleichung (2) erforderlich, die noch von der Geschwindigkeit abhängig ist. Diese Abhängigkeit kann jedoch entweder durch Addition einer bekannten Größe eliminiert werden (s.o.) oder der Beobachter wird auf folgende Weise entworfen:

Um die Geschwindigkeit zu eliminieren wird der Teil der Matrix  $L_2$ , der mit  $a_y$  multipliziert wird, durch  $v$  dividiert:

$$L_2 = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix}, \quad \begin{bmatrix} l_{11} \\ l_{21} \end{bmatrix} = \begin{bmatrix} l_{11}^* \\ l_{21}^* \end{bmatrix} \frac{K_{\phi_R}}{v} \quad (10)$$

Mit (8) und (10) wird (9) zu:

$$\begin{bmatrix} \dot{\hat{\beta}} \\ \dot{\hat{r}} \end{bmatrix} = \begin{bmatrix} Y_\beta K_{\phi_R} & -1 \\ \frac{N_\beta}{I_z} & 0 \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + \begin{bmatrix} Y_\delta K_{\phi_R} \\ \frac{N_\delta}{I_z} \end{bmatrix} \delta_L^* + \begin{bmatrix} mK_{\phi_R} + \frac{K_{\phi_R}}{v} & Y_r K_{\phi_R} \\ 0 & \frac{N_r}{I_z} \end{bmatrix} \begin{bmatrix} a_y \\ r \end{bmatrix} + \begin{bmatrix} l_{11}^* \\ l_{21}^* \end{bmatrix} \frac{K_{\phi_R}}{v} a_y - \begin{bmatrix} l_{11}^* \\ l_{21}^* \end{bmatrix} \left( \dot{\hat{\beta}} + \hat{r} \right) + \begin{bmatrix} l_{12} \\ l_{22} \end{bmatrix} (r - \hat{r})$$

und weiter zu:

$$\begin{bmatrix} 1 + l_{11}^* & 0 \\ l_{21}^* & 1 \end{bmatrix} \begin{bmatrix} \dot{\hat{\beta}} \\ \dot{\hat{r}} \end{bmatrix} = \left( \begin{bmatrix} Y_\beta K_{\phi_R} & -1 \\ \frac{N_\beta}{I_z} & 0 \end{bmatrix} - \begin{bmatrix} l_{11}^* + l_{12} \\ l_{21}^* + l_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \right) \begin{bmatrix} \hat{\beta} \\ \hat{r} \end{bmatrix} + \begin{bmatrix} Y_\delta K_{\phi_R} \\ \frac{N_\delta}{I_z} \end{bmatrix} \delta_L^* + \begin{bmatrix} -mK_{\phi_R} + (1 + l_{11}^*) \frac{K_{\phi_R}}{v} & Y_r K_{\phi_R} + l_{12} \\ l_{21}^* \frac{K_{\phi_R}}{v} & \frac{N_r}{I_z} + l_{22} \end{bmatrix} \begin{bmatrix} a_y \\ r \end{bmatrix}$$

Die Systemmatrix des Beobachters ist dann:

$$\begin{bmatrix} 1 + l_{11}^* & 0 \\ l_{21}^* & 1 \end{bmatrix}^{-1} \left( \begin{bmatrix} Y_\beta K_{\phi_R} & -1 \\ \frac{N_\beta}{I_z} & 0 \end{bmatrix} - \begin{bmatrix} l_{11}^* + l_{12} \\ l_{21}^* + l_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \right)$$

Mit

$$\begin{bmatrix} 1 + l_{11}^* & 0 \\ l_{21}^* & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{1+l_{11}^*} & 0 \\ -\frac{l_{21}^*}{1+l_{11}^*} & 1 \end{bmatrix}, \quad \begin{bmatrix} Y_{\beta} K_{\phi_R} & -1 \\ \frac{N_{\beta}}{I_z} & 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

wird die Systemmatrix des Beobachters zu:

$$F = \begin{bmatrix} \frac{1}{1+l_{11}^*} a_{11} & \frac{1}{1+l_{11}^*} (a_{12} - l_{11}^* - l_{12}) \\ -\frac{l_{21}^*}{1+l_{11}^*} a_{11} + a_{21} & -\frac{l_{21}^*}{1+l_{11}^*} (a_{12} - l_{11}^* - l_{12}) + a_{22} - l_{21}^* - l_{22} \end{bmatrix}$$

d.h. die Systemmatrix des Beobachters kann mittels der Rückführung frei gewählt werden. Um Standard-Software-Tools zum Beobachterentwurf nutzen zu können, schreibt man die Systemmatrix in der Form  $(A - LC)$  mit  $C = I$ :

$$\left( \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} - \begin{bmatrix} \frac{l_{11}^*}{1+l_{11}^*} a_{11} & \frac{l_{11}^* (a_{12} + 1) + l_{12}}{1+l_{11}^*} \\ \frac{l_{21}^*}{1+l_{11}^*} a_{11} & \frac{l_{21}^*}{1+l_{11}^*} (a_{12} - l_{11}^* - l_{12}) + l_{21}^* + l_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

Aus dem Gleichungssystem

$$\begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} = \begin{bmatrix} \frac{l_{11}^*}{1+l_{11}^*} a_{11} & \frac{l_{11}^* a_{12} + l_{11}^* + l_{12}}{1+l_{11}^*} \\ \frac{l_{21}^*}{1+l_{11}^*} a_{11} & \frac{l_{21}^*}{1+l_{11}^*} (a_{12} - l_{11}^* - l_{12}) + l_{21}^* + l_{22} \end{bmatrix}$$

können mit Hilfe von Gl. (10) die Parameter der Rückführungsmatrix berechnet werden:

$$\begin{aligned} l_{11} &= -\frac{L_{11}}{L_{11} - a_{11}} \frac{K_{\phi_R}}{v} \\ l_{12} &= \frac{-L_{12} a_{11} + L_{11} (a_{12} + 1)}{L_{11} - a_{11}} \\ l_{21} &= -\frac{L_{21}}{L_{11} - a_{11}} \frac{K_{\phi_R}}{v} \\ l_{22} &= L_{22} + \frac{L_{21}}{L_{11} - a_{11}} (a_{12} + 1 - L_{12}) \end{aligned}$$

Hiermit kann die Dynamik des Beobachters unabhängig von der Geschwindigkeit festgelegt werden, in den von der Ausgangsgröße abhängigen Term und die Rückführung  $L_2$  kann online die Geschwindigkeit eingerechnet werden. Somit wird in diesem Fall  $a_y/v$  zurückgeführt.

## 4.2 Transformation auf Beobachtungsnormalform

In diesem Abschnitt wird gezeigt, wie das Fahrzeugmodell mit Hilfe einer Zustandstransformation in die Beobachtungsnormalform überführt werden kann. Diese Vorgehensweise ermöglicht einen geschwindigkeitsunabhängigen Beobachterentwurf mit Aufschaltung nur einer Messgröße.

Das System

$$\begin{aligned} \dot{x} &= A(v)x + B(v)u + E(v)d \\ y &= C(v)x + Du \end{aligned} \quad (11)$$

wird hierbei durch die Transformationsmatrix  $T(v)$  mit  $z = T(v)x$  in das neue System

$$\begin{aligned}\dot{z} &= \tilde{A}(v)z + \tilde{B}(v)u + \tilde{E}(v)d \\ y &= \tilde{C}(v)z + \tilde{D}u\end{aligned}\quad (12)$$

überführt.

Mit zwei unterschiedlichen Transformationsmatrizen können so zwei Beobachter zur Fehlerentdeckung entworfen werden, die durch ihre Struktur direkt eine Fehlerisolation ermöglichen. Dies wird möglich, da durch die so entworfenen Beobachter Residuen generiert werden können, die nur von je einer Messgröße abhängig sind.

#### 4.2.1 Aufschaltung Querbeschleunigung

Zunächst wird ausführlich die Transformation auf Beobachtungsnormalform sowie die anschließende Umformung für den Fall der Aufschaltung der Messgröße Querbeschleunigung beschrieben. Für die Bestimmung der Transformationsmatrix  $T(v)$  wird das Modell mit nur einer Ausgangsgleichung betrachtet:

$$\begin{aligned}\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} &= \underbrace{\begin{bmatrix} \frac{Y_\beta K_{\phi R}}{m} \frac{1}{v} & \frac{Y_r K_{\phi R}}{m} \frac{1}{v} & -1 \\ \frac{N_\beta}{I_z} & \frac{N_r}{I_z} & \end{bmatrix}}_{=:A(v)} \begin{bmatrix} \beta \\ r \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{Y_\delta K_{\phi R}}{m} \frac{1}{v} \\ \frac{N_\delta}{I_z} \end{bmatrix}}_{=:B(v)} \delta_L^* + \underbrace{\begin{bmatrix} -\frac{g}{v} \\ 0 \end{bmatrix}}_{=:E(v)} \sin \alpha_x \\ a_y &= \underbrace{\begin{bmatrix} \frac{Y_\beta}{m} & \frac{Y_r}{m} \end{bmatrix}}_{=:C(v)} \begin{bmatrix} \beta \\ r \end{bmatrix} + \underbrace{\frac{Y_\delta}{m}}_{=:D} \delta_L^*.\end{aligned}$$

Mit der Transformationsmatrix

$$T(v) = \begin{bmatrix} \frac{(Y_r N_\beta - N_r Y_\beta)}{m I_z} & -\frac{Y_\beta}{m} \\ \frac{Y_\beta}{m} & \frac{Y_r}{m} \end{bmatrix}$$

kann das Modell in die Beobachtungsnormalform überführt werden:

$$\begin{aligned}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} 0 & \frac{K_{\phi R}(N_\beta Y_r - Y_\beta N_r)}{m I_z v} - \frac{N_\beta}{I_z} \\ 1 & \frac{Y_\beta K_{\phi R}}{m v} + \frac{N_r}{I_z} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \frac{K_{\phi R}(N_\beta Y_r - Y_\beta N_r) Y_\delta}{m^2 I_z v} - \frac{Y_\beta N_\delta}{m I_z} \\ \frac{Y_\beta Y_\delta K_{\phi R}}{m^2 v} + \frac{Y_r N_\delta}{m I_z} \end{bmatrix} \delta_L^* \\ &+ \begin{bmatrix} -\frac{g(Y_r N_\beta - N_r Y_\beta)}{m I_z v} - \frac{Y_\beta N_\delta}{m I_z} \\ -\frac{g Y_\beta}{m v} \end{bmatrix} \sin \alpha_x,\end{aligned}\quad (13)$$

$$a_y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \frac{Y_\delta}{m} \delta_L^*.\quad (14)$$

Der geschwindigkeitsabhängige Anteil der Systemmatrix des Modells (13),(14) kann nun als additiver Anteil ausgeklammert werden

$$\begin{aligned}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} 0 & -\frac{N_\beta}{I_z} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \frac{K_{\phi R}(N_\beta Y_r - Y_\beta N_r)}{I_z m v} \\ \frac{Y_\beta K_{\phi R}}{m v} + \frac{N_r}{I_z} \end{bmatrix} z_2 + \begin{bmatrix} \frac{K_{\phi R}(N_\beta Y_r - Y_\beta N_r) Y_\delta}{I_z m^2 v} - \frac{Y_\beta N_\delta}{m I_z} \\ \frac{Y_\beta Y_\delta K_{\phi R}}{m^2 v} + \frac{Y_r N_\delta}{m I_z} \end{bmatrix} \delta_L^* \\ &+ \begin{bmatrix} -\frac{g(Y_r N_\beta - N_r Y_\beta)}{m I_z v} - \frac{Y_\beta N_\delta}{m I_z} \\ -\frac{g Y_\beta}{m v} \end{bmatrix} \sin \alpha_x\end{aligned}$$

und der Zustand  $z_2$  mit Hilfe der neuen Ausgangsgleichung (14) substituiert werden. Zusätzlich wird eine Konstante  $k_1$  eingeführt um die Stabilität des zu entwerfenden Beobachters zu garantieren. Damit ergibt sich das neue Modell mit Aufschaltung der Messgröße Querbeschleunigung ( $a_y$ ) zu:

$$\begin{aligned} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} 0 & -\frac{N_\beta}{I_z} \\ 1 & k_1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \frac{K_{\phi R}(N_\beta Y_r - Y_\beta N_r)}{I_z m v} \\ \frac{Y_\beta K_{\phi R}}{m v} + \frac{N_r}{I_z} - k_1 \end{bmatrix} a_y + \begin{bmatrix} -\frac{Y_\beta N_\delta}{m I_z} \\ \frac{(N_\delta Y_r - Y_\delta N_r)}{m I_z} + k_1 \frac{Y_\delta}{m} \end{bmatrix} \delta_L^* \\ &+ \begin{bmatrix} -\frac{g(Y_r N_\beta - N_r Y_\beta)}{m I_z v} - \frac{Y_\beta N_\delta}{m I_z} \\ -\frac{g Y_\beta}{m v} \end{bmatrix} \sin \alpha_x \\ a_y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \frac{Y_\delta}{m} \delta_L^*. \end{aligned}$$

#### 4.2.2 Aufschaltung Gierrate

Mit dem gleichen Verfahren wie in Abschnitt 4.2.1 beschrieben, kann mit Hilfe der Transformationsmatrix

$$T(v) = \begin{bmatrix} \frac{N_\beta}{I_z} & -\frac{Y_\beta K_\phi}{m v} \\ 0 & 1 \end{bmatrix}$$

ein Modell aufgestellt werden, dessen Systemgleichung lediglich die Messgröße Gierrate ( $r$ ) aufgeschaltet wird:

$$\begin{aligned} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} 0 & -\frac{N_\beta}{I_z} \\ 1 & k_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \frac{K_{\phi R}(N_\beta Y_r - Y_\beta N_r)}{I_z m v} \\ \frac{Y_\beta K_{\phi R}}{m v} + \frac{N_r}{I_z} - k_2 \end{bmatrix} r + \begin{bmatrix} \frac{K_{\phi R}(N_\beta Y_\delta - Y_\beta N_\delta)}{m I_z v} \\ \frac{N_\delta}{I_z} \end{bmatrix} \delta_L^* \\ &+ \begin{bmatrix} -\frac{g N_\beta}{I_z v} \\ 0 \end{bmatrix} \sin \alpha_x \\ r &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}. \end{aligned}$$

Somit stehen zwei Versionen des Modells zur Verfügung, mit denen jeweils für eine der Messgrößen eine Fehlerentdeckung realisiert werden kann. Durch die Unabhängigkeit der beiden Modelle von je einer Messgröße kann zusätzlich eine Fehlerisolation erreicht werden.

## Literatur

- [1] M. Mitschke. *Dynamik der Kraftfahrzeuge Band C: Fahrverhalten*. Springer-Verlag, 1990.
- [2] H.-G. Schulz. Einspurmodell. Interner bericht, Universität Duisburg-Essen, Juli 2003.



# Verbrennungsmerkmale aus der Motordrehzahl

Thorsten Schmidt, IAV GmbH

Hermann Fehrenbach, Fraunhofer-Institut IITB

## Zusammenfassung

Neuartige Brennverfahren mit homogener Selbstzündung (HCCI) erfordern eine aufwändigere Regelung des Verbrennungverlaufes anhand von Verbrennungsmerkmalen, die durch das Motormanagement erfasst werden müssen. Durch eine Messung des Zylinderdrucks und Druckverlaufsanalyse können diese Verbrennungsmerkmale bestimmt werden. Drucksensorik für Serienfahrzeuge ist jedoch sehr kostspielig.

In diesem Artikel werden Verfahren vorgestellt, die als Verbrennungsmerkmale das indizierte Motormoment und den Brennbeginn eines aufgeladenen Dieselmotors durch Auswertung der hochaufgelösten Motordrehzahl ermitteln. Dabei ist es notwendig, die Güte des Drehzahlsignals durch eine Kompensation des geometrischen Zahnteilungsfehlers des Gebirrades zu verbessern.

Neben der kennliniengestützten Auswertung des Drehzahlsignals zeigt die modellgestützte Auswertung des hochaufgelösten Drehzahlsignals vielversprechende Ergebnisse. Der Einlernvorgang für die Erstellung des Kennfeldes könnte damit künftig entfallen.

Bisher vernachlässigte Nebenmomente aus Steuertrieb und Antriebsstrang beeinträchtigen die Genauigkeit des Verfahrens. Es werden Ansätze zur physikalischen Modellierung und zukünftigen Kompensation dieser Momente vorgestellt.

## 1 Einleitung

Das Motormoment ist die zentrale Beschreibungsgröße im Antriebsmanagement. Heutige elektronische Motormanagement-Systeme bilden den über den Pedalwertgeber erfassten Fahrerwunsch deshalb über ein internes Momentenstruktur-Modell auf die Ansteuersignale des Einspritzsystems ab.

Besondere Bedeutung erhält das Motormanagement durch die gegenwärtige Entwicklung neuer Brennverfahren mit „Homogener Verbrennung“ und Kompressionszündung, auch bezeichnet als „Homogeneous Combustion Compression Ignition“

(HCCI) [8]. Bei diesen Brennverfahren werden niedrige Verbrennungstemperaturen durch eine möglichst gleichzeitige Zündung eines Kraftstoff-Luft-Gemisches im gesamten Brennraum angestrebt. Voraussetzung für ein solches Verfahren ist eine weitgehende Homogenisierung der Zylinderfüllung, was beim Dieselmotor z.B. durch eine frühe Kraftstoffeinspritzung realisiert werden kann. Der Einspritzzeitpunkt hat bei diesen Verfahren nur noch einen schwachen Einfluss auf die Verbrennungslage. Der Zündverzug ist deutlich höher als bei konventionell direkteinspritzenden Dieselmotoren.

Eine Verbrennungssteuerung durch den Spritzbeginn als Maß für den Verbrennungsbeginn ist aus diesen Gründen nur noch sehr eingeschränkt möglich. Der Übergang zur Regelung des Verbrennungsvorganges auf Basis von gemessenen Verbrennungsmerkmalen bietet hier einen Ausweg. Ein möglicher Ansatz dafür ist ein zylinderdruckgeführtes Motormanagement [6], das eine kostspielige Brennraumdruck-Sensorik voraussetzt.

Ein kostengünstiges Verfahren zur Ermittlung des indizierten Motormoments haben die IAV GmbH, das Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB) und die Audi AG entwickelt. Die drei Partner haben zur besseren Zusammenarbeit einen Arbeitskreis gegründet. Das vom Arbeitskreis entwickelte Verfahren basiert auf der Auswertung der Motordrehzahl und benötigt außer dem Ladedruck keine weiteren Sensorsignale. Die Weiterentwicklung dieses Verfahrens ebnet den Weg zur Ermittlung von Verbrennungsmerkmalen aus der gemessenen Motordrehzahl. Erste Ergebnisse zeigen, dass diese Methode eine Basis für ein zylinderselektives Motormanagement ohne Drucksensorik sein kann.

## 2 Modellansatz

Der Modellansatz der berührungslosen Drehmomentmessung ist in [1] beschrieben. Die Drehmomentbilanz eines Einzylindermotors ergibt sich danach unter der Annahme einer hinreichend steifen Kurbelwelle und einer hinreichenden Entkopplung vom Antriebsstrang zu:

$$\Theta \ddot{\phi} + \frac{1}{2} \Theta' \dot{\phi}^2 = \tilde{M}_g + \bar{M}_g - \bar{M}_r - \bar{M}_l \quad (1)$$

mit:

$\ddot{\phi}$ : Winkelbeschleunigung,

$\dot{\phi}$ : Winkelgeschwindigkeit,

$\Theta$ : Massenträgheitsmoment der rotierenden Massen,

$\Theta'$ : Ableitung des Massenträgheitsmoments nach dem Kurbelwinkel,

$\tilde{M}_g$ : Wechselanteil des Gasdrehmoments,

$\bar{M}_g$ : Gleichanteil des Gasdrehmoments,

$\bar{M}_r$ : Reibungsdrehmoment,

$\bar{M}_l$ : Last- oder Nutzdrehmoment.

Das freie Drehmoment an der Kurbelwelle setzt sich hierbei aus einem Gleich- und einem Wechselanteil zusammen. Der Massendrehmomentanteil der oszillierenden Bauteile  $1/2 \Theta' \dot{\phi}^2$  hingegen besteht aus einem reinen Wechselanteil. Das Gasdrehmoment besteht aus dem eigentlichen Antriebsmoment in Form eines positiven Gleichanteils  $\bar{M}_g$  und einem Wechselanteil  $\tilde{M}_g$ . Das Reibungsdrehmoment und das Lastdrehmoment können in erster Näherung als in einem Arbeitspunkt (Drehzahl/Last) konstante Anteile  $\bar{M}_r$  und  $\bar{M}_l$  angenommen werden. Im stationären Fall befinden sich das Antriebsmoment und die Summe aus Reibungsdrehmoment  $\bar{M}_r$  und Lastdrehmoment  $\bar{M}_l$  im Gleichgewicht, so dass sich die vereinfachte Gleichung

$$\Theta \ddot{\phi} + \frac{1}{2} \Theta' \dot{\phi}^2 = \tilde{M}_g \quad (2)$$

ergibt. Es kann experimentell nachgewiesen werden, dass der Effektivwert des Gaswechseldrehmoments  $\tilde{M}_{g_{rms}}$  proportional zum Energieumsatz und damit proportional zu  $\bar{M}_g$  ist. Zur Bestimmung des Gaswechseldrehmoments  $\tilde{M}_g$  kann somit im stationären Fall die Summe aus freiem Drehmoment  $\Theta \ddot{\phi}$  und Massendrehmoment  $1/2 \Theta' \dot{\phi}^2$  herangezogen werden.

Im instationären Fall gilt:

$$\Delta M = \bar{M}_g - \bar{M}_r - \bar{M}_l \neq 0 \quad (3)$$

Mit Hilfe von Instationärversuchen kann gezeigt werden, dass  $\Delta M \ll \tilde{M}$  ist.  $\Delta M$  kann somit vernachlässigt werden. Demzufolge ist der Verlauf der Drehmomente innerhalb eines Arbeitsspiels als hinreichend stationär zu betrachten, so dass wie im stationären Fall von  $\tilde{M}_g$  auf  $\bar{M}_g$  geschlossen werden kann.

Um das Nettodrehmoment, den linken Teil von Gleichung (1), also die Summe aus freiem Drehmoment und Massendrehmoment, zu berechnen, werden noch  $\Theta$  und  $\Theta'$  sowie die Winkelgeschwindigkeit  $\dot{\phi}$  und die Winkelbeschleunigung  $\ddot{\phi}$  benötigt. Um  $\Theta$  und  $\Theta'$  berechnen zu können, müssen die Massen- und Geometriedaten des Motors bekannt sein:

- Massenträgheitsmomente der rotierenden Teile,
- oszillierende Massen der Kolben und Pleuel,
- Hub,
- Pleuellänge und
- Schränkung oder Desachsierung.

Der Effektivwert des Nettodrehmoments

$$M_{Nrms} = \sqrt{\frac{1}{\varphi_0} \int_0^{\varphi_0} \left( \Theta \ddot{\varphi} + \frac{1}{2} \Theta' \dot{\varphi}^2 \right)^2 d\varphi} \quad (4)$$

mit  $\varphi_0 = 720^\circ$  KW ist proportional zum Energieumsatz und damit proportional zu  $\bar{M}_l$ . Dieser Zusammenhang zwischen  $\bar{M}_l$  und  $\bar{M}_{Nrms}$  wird zum Aufbau eines Kennfelds verwendet. Gemessenen Werten der Drehzahl und den berechneten Effektivwerten des Nettodrehmoments werden gemessene Werte des Lastdrehmoments  $\bar{M}_l$  zugeordnet. Alternativ können den berechneten Effektivwerten des Nettodrehmoments auch gemessene Werte des indizierten Mitteldrucks  $P_{mi}$  zugeordnet werden. Damit ist es nach dem Einlernen des entsprechenden Kennfelds für einen bestimmten Motortyp möglich, für jedes Arbeitsspiel im laufenden Betrieb das momentane Lastdrehmoment oder den indizierten Mitteldruck anzugeben.

### 3 Erfassung der Winkelgeschwindigkeit

Ausgangspunkt für die ‚sensorlose‘ Drehmomentenbestimmung ist die hochgenaue Winkelgeschwindigkeitsmessung.

Bei winkelgesteuerten Diesel-Einspritzsystemen (Common-Rail, Pumpe-Düse) wird die Winkelgeschwindigkeit über ferromagnetische Impulsgeberräder aufgenommen. Ein solches Geberrad ist beispielsweise in Bild 1 dargestellt ist.



Bild 1: 60-2-2-Geberrad für Dieselmotoren (56 Zähne, 2 Lücken)

Bild 2 zeigt für ein solches Geberrad exemplarisch das Ausgangssignal eines induktiven Sensors, wie er zur Drehzahlerfassung eingesetzt wird. Auffällig ist der Signalverlauf in der Lücke. Der Signalverlauf zeigt hier eine stark nichtlineare Abhän-

gigkeit in Betrag und Phase von der Drehzahl. Für die weitere Signalverarbeitung ist es sinnvoll, die fehlenden Zähne in der Lücke zu interpolieren.

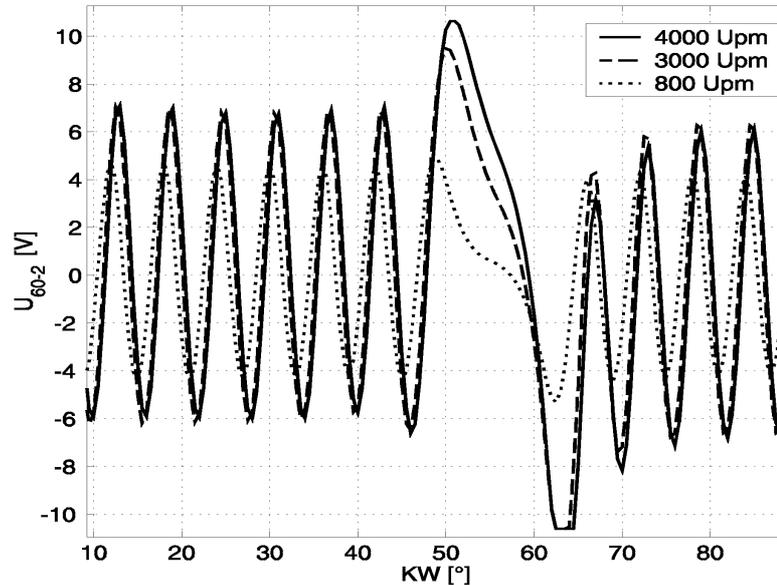


Bild 2: Induzierte Sensorspannung bei verschiedenen Drehzahlen

#### 4 Geometriefehler der Geberräder

Fertigung und Einbau der Drehgeberräder sind toleranzbehaftet. Insbesondere ein geometrischer Zahnteilungsfehler führt zu einem systematischen, für jedes Geberrad individuellen Drehzahlfehler. Diese Fehler können durchaus im Bereich von  $\pm 0,5$  °KW liegen. Solche Abweichungen führen zu nicht tolerablen Fehlern bei der Näherung der Winkelgeschwindigkeit gemäß dem Differenzenquotienten

$$\omega_{mess} = \frac{\Delta\varphi_{soll}}{\Delta T} \quad (5)$$

und folglich auch zu Fehlern bei der Bestimmung des Wechseldrehmoments. Ein Verfahren zur Kompensation dieses Drehzahlfehlers im Motorsteuergerät ist deshalb erforderlich. Ein solches Verfahren bietet auch die Option, die Qualitätsanforderungen an Geberräder zu senken, was eine kostengünstigere Fertigung ermöglicht.

#### 5 Geberradfehler-Kompensation

In [2] wurden zwei geeignete Kompensationsverfahren vorgestellt [4][5]. Das eine Verfahren nutzt die Gegenphasigkeit der Gas- und Massenmomente zur Bestimmung der Geometriefehler des verwendeten Drehgeberrads. Der große Vorteil dieses Verfahrens liegt in seiner einfachen Handhabung und darin, dass außer der Zylinder- und Zahnanzahl kein weiteres Vorwissen über den zu betrachtenden Motor notwendig ist. Das zweite Verfahren wählt einen Ansatz über die Modellierung der Kurbelwellenenergie zur Bestimmung des Geberradfehlers. Durch die Geberrad-

adaption kann der durch das Geberrad verursachte Fehler um mindestens eine Größenordnung reduziert werden.

Die vorgestellten Kompensationsverfahren wurden an einem Fünf-Zylinder-Reihenmotor überprüft. Die Referenz-Winkelgeschwindigkeit wurde mit einem optischen Sensor erfasst, wie er als Winkelmarkengeber für Indiziersysteme verwendet wird. Bei der Zylinderdruckindizierung wird dieser Geber standardmäßig am freien Ende der Kurbelwelle montiert. Messungen haben gezeigt, dass durch die Torsion der Kurbelwelle bei dieser Anordnung ein dynamischer Winkelfehler die Referenzmessung verfälscht. Deshalb wurde der Referenzmarkengeber direkt am Schwungrad montiert.

Die Ergebnisse der Geberradfehlerschätzung mit den beiden vorgestellten Verfahren im Vergleich zur Fehlerbestimmung mit Hilfe der Referenzmessung sind in Abbildung 3 zu sehen.

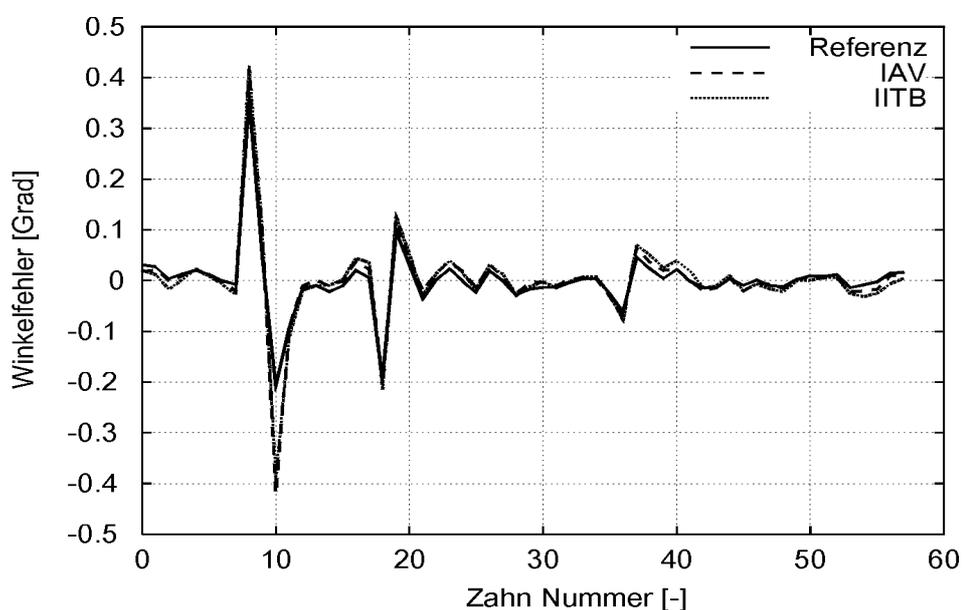


Bild 3: Vergleich der geschätzten Geberradfehler mit der Referenz

Der Verlauf des Fehlers wird erwartungsgemäß im Bereich der Zahnücke (Zahn Nr. 8-11; Lücke zwischen Nr. 9 und 10 selbst nicht dargestellt) am größten. Beide Fehlerschätzverfahren liefern vergleichbare Ergebnisse. Es fällt auf, dass in bestimmten Bereichen beide Verfahren in fast gleichem Maße von der Referenz abweichen. Die maximale Abweichung von der Referenz beträgt in dem betrachteten Fall  $0,02^\circ$  KW.

Um die Reproduzierbarkeit der Geberradfehler-Schätzung zu testen, wurden mehrere Versuche am gleichen Motor mit demselben Geberrad unternommen. Die maximalen Abweichungen zwischen den Versuche liegen unter  $0,008^\circ$  KW. Abbildung 4 zeigt beispielhaft den Verlauf des geschätzten Zahnfehlers (fehlende Zähne in der Zahnücke sind in diesem Fall interpoliert) bei fünf verschiedenen Messungen.

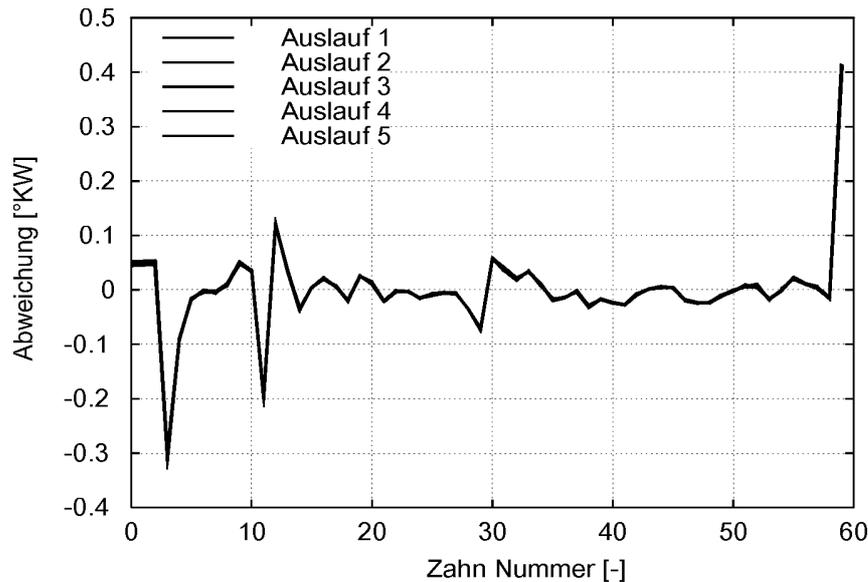


Bild 4: Reproduzierbarkeit der Geberradfehler-Schätzung

## 6 Einfluss des Ladedrucks

Das Drehmomentenschätzverfahren soll auch bei aufgeladenen Motoren zuverlässig funktionieren. Eine Aufladung führt aufgrund der höheren Zylinderfüllung zu einer höheren Amplitude des Gaswechsellmoments. Besonders transiente Vorgänge bei geregelten Turboladern können hierbei zu Problemen führen. In diesen Fällen herrscht im Betrieb nicht immer derselbe Ladedruck, wie bei dem unter stationären Bedingungen aufgenommenen Kennfeldpunkt der entsprechenden Drehzahl-Last-Kombination. Durch den unterschiedlichen Ladedruck wird in diesem Fall ein unterschiedliches Wechseldrehmoment bestimmt und damit ein falscher Lastpunkt im Kennfeld. Daher ist es für die Bestimmung des Lastmoments notwendig das entsprechende Kennfeld sowie das geschätzte Wechseldrehmoment als Eingangsgröße vom Ladedruck unabhängig zu machen. Dazu muss der Wechseldrehmomentverlauf von seinem ladedruckabhängigen Kompressionsanteil befreit werden.

Zur Schätzung des Kompressionsanteils wird die Ladedruckinformation des Motorsteuergeräts übernommen. Motorprozess-Simulationen und experimentelle Messungen haben einen linearen Einfluss des Ladedrucks auf den Drehmomenteffektivwert gezeigt. Durch die Kompensation des Ladedrucks wird auch die Abhängigkeit vom Atmosphärendruck beseitigt. Diese könnte anderenfalls z.B. im Höhenbetrieb zu größeren Schätzfehlern führen.

## 7 Erweiterter Modellansatz

Der erweiterte Modellansatz berücksichtigt die Einflüsse des Geberradfehlers und des Ladedrucks. Das Drehmomentmessverfahren kann somit auch im Fahrzeug eingesetzt werden. Hierbei kommt ausschließlich die motoreigene Sensorik zum Einsatz. Das Blockschaltbild des Drehmomentenschätzverfahrens zeigt Bild 5.

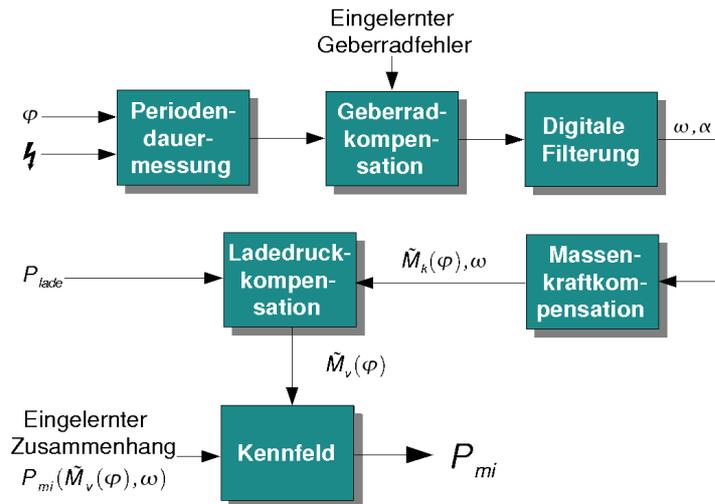


Bild 5: Schematische Darstellung des Drehmomentschätzverfahrens

Das mittels Periodendauermessung gewonnene Drehzahl-signal durchläuft zunächst die Geberrad-adaption. Der Geberradfehler muss für jeden Motor vorher nur einmal eingelernt und dann abgelegt werden.

Durch eine Tiefpassfilterung wird dieses Signal von hochfrequenten Störungen befreit. Man erhält den Verlauf der Winkelgeschwindigkeit  $\dot{\varphi}$ , und nach anschließender Differentiation den Verlauf der Winkelbeschleunigung  $\ddot{\varphi}$ . Zur Tiefpassfilterung und zur Differentiation kommen hierbei steilflankige FIR-Filter zum Einsatz.

Die Massenkraftkompensation bewirkt, dass der Momenteneinfluss durch die oszillierenden Massen kompensiert und so das reine Gaswechseldrehmoment ermittelt wird. Die Ladedruckkompensation befreit das ermittelte Wechseldrehmoment von seinem durch den Ladedruck bedingten Anteil. Hierdurch kann das Verfahren auch problemlos für aufgeladene Motoren verwendet werden. Ziel des Drehmomentmessverfahrens ist die Bestimmung des indizierten Mitteldrucks. Die Schätzung dieser Größe aus der gemessenen Drehzahl und dem Wechseldrehmoment erfolgt über ein vorher für jeden Motortyp eingelerntes Kennfeld. Dieses Kennfeld beschreibt den Zusammenhang zwischen Wechseldrehmoment, mittlerer Drehzahl und indiziertem Mitteldruck.

## 8 Aussetzererkennung

Aus dem massenkraftkompensierten Momentenverlauf  $\tilde{M}_K(\varphi)$  (siehe Bild 5) können auch zylinderspezifische Arbeitsbeiträge berechnet werden, um den Lauf des Motors beurteilen zu können. Damit kann mangelhafte Verbrennung detektiert sowie das Ausmaß schlechter Verbrennung quantifiziert werden.

Verfügt der Motor beispielsweise über gleichmäßige Zündabstände, so kann der Lauf des Motors anhand der zylinderspezifischen Wechselsarbeit  $W_i$  beurteilt werden:

$$W_i = \frac{1}{\varphi_2 - \varphi_1} \int_{\varphi_1}^{\varphi_2} \tilde{M}_K(\varphi) d\varphi \quad (6)$$

Die Integrationsgrenzen  $\varphi_1$  und  $\varphi_2$  sind je nach Zylinderanzahl und Dauer der Verbrennung so zu wählen, dass sie den Expansionsbereich des jeweiligen Zylinders eingrenzen, ohne den Einfluss der Überlagerung des Drehmoments der einzelnen Zylinder zu groß werden zu lassen.

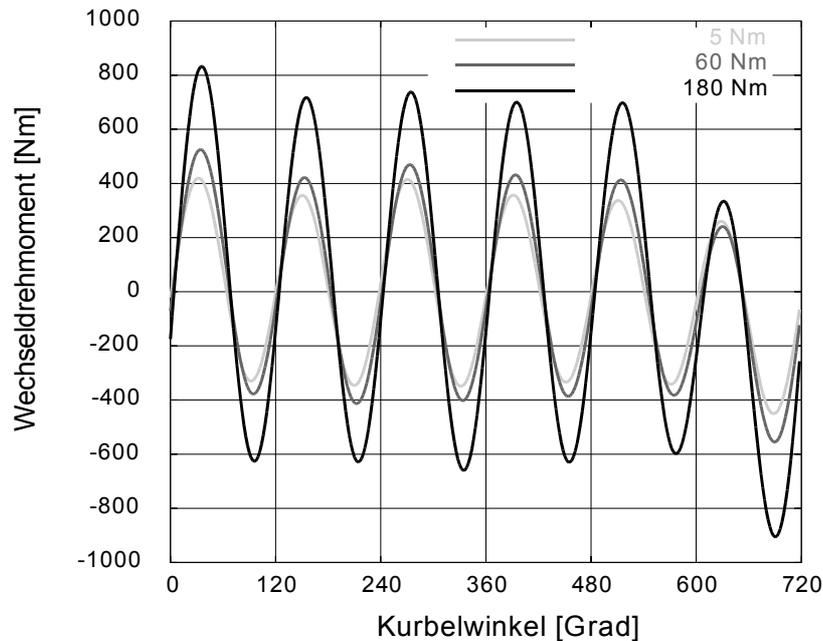


Bild 6: Wechseldrehmomentverlauf an einem Sechszylinder-Dieselmotor bei einem Zündaussetzer an Zylinder 5 (Kurbelwinkelbereich  $600^\circ$ - $720^\circ$ ) bei drei verschiedenen Lastzuständen und einer Drehzahl von 4400 Upm

Abbildung 6 zeigt den Verlauf des schargemittelten Wechseldrehmoments bei Höchstdrehzahl eines Sechszylinder-Common-Rail-Dieselmotors bei einem permanenten Zündaussetzer an einem Zylinder (Zylinder 5) für drei unterschiedliche Lasten (5 Nm, 60 Nm und 180 Nm). Bis hin zum nahezu lastlosen Betriebspunkt (5 Nm) ist der mittlere Arbeitsbeitrag von Zylinder 5 deutlich niedriger als die Beiträge der restlichen Zylinder. Eine Erkennung von Zündaussetzern kann mit dieser Methode nach bisherigen Erfahrungen auch bei hohen Zylinderzahlen erfolgen. Das beinhaltet auch die Erkennung von einzelnen, sporadisch auftretenden Aussetzern.

## 9 Modellierung des Zylinderdruckverlaufs zur Bestimmung des indizierten Mitteldrucks

Das bisher gezeigte Drehmomentenmessverfahren basiert darauf, dass der Zusammenhang zwischen indiziertem Moment und dem über die hochaufgelöste Drehzahl

bestimmten Wechseldrehmoment experimentell bestimmt und in einem Kennfeld abgelegt wird. Auch die Kompensation des Ladedrucks erfolgt auf der Basis einer eingelernten Kennlinie.

Da über den Verbrennungsablauf einerseits und über zusätzliche Messgrößen andererseits einiges Vorwissen bezüglich des Momentenverlaufs an der Kurbelwelle vorhanden ist, liegt es nahe die kompensierten Wechseldrehmomentverläufe modellgestützt auszuwerten.

Ein möglicher Ansatz für ein modellbasiertes Schätzverfahren ist in Bild 7 dargestellt.

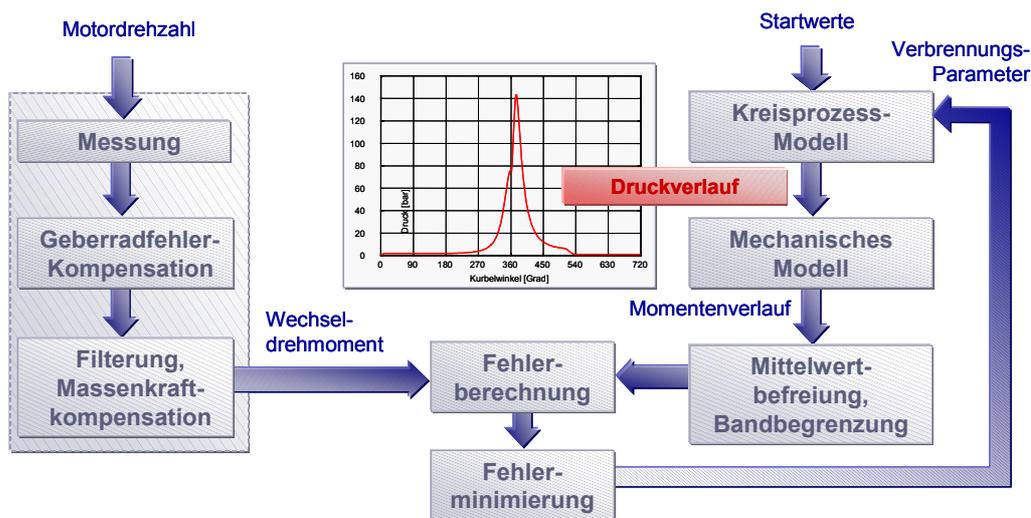


Bild 7: Modellabgleich-Ansatz

In der linken Hälfte wird aus dem gemessenen Motordrehzahlsignal das Gaswechseldrehmoment nach dem in Abschnitt 7 beschriebenen Verfahren ermittelt. Die rechte Hälfte zeigt ein Modell des Motors, mit dem die thermodynamischen und mechanischen Eigenschaften des verwendeten Motors simuliert werden können. Dabei wird aus Modellparametern der Verbrennung zunächst ein Druckverlauf über dem Kurbelwinkel berechnet. Über ein mechanisches Modell des Motors wird daraus ein Momentenverlauf an der Kurbelwelle ermittelt. Dieser Momentenverlauf wird vom Mittelwert befreit und bandbegrenzt, um ein dem Wechseldrehmoment vergleichbares Signal zu erhalten. Die Fehlerberechnung führt die Differenz zwischen beiden Momentenverläufen einem Optimierungsverfahren zu, das eine Fehlerminimierung durch Variation der Verbrennungsparameter des Motormodells durchführt. Die mit diesem Verfahren iterativ gefundenen Modell-Verbrennungsparameter entsprechen im Optimum den Parametern des realen Motors.

Die Vorteile dieses Verfahrens liegen klar auf der Hand: Ohne einen aufwändigen Einlernvorgang können aus dem Wechseldrehmomentverlauf die Druckverläufe in den einzelnen Zylindern geschätzt werden. Allerdings erfordert dieses iterative Modellabgleich-Verfahren eine Vielzahl an Berechnungen, die beim aktuellem Entwicklungsstand auf einem Motorsteuergerät heutiger Leistung noch nicht in Echtzeit

ausgeführt werden können. Die folgenden Ergebnisse sind deshalb durch eine Offline-Verarbeitung gemessener Drehzahlen ermittelt worden.

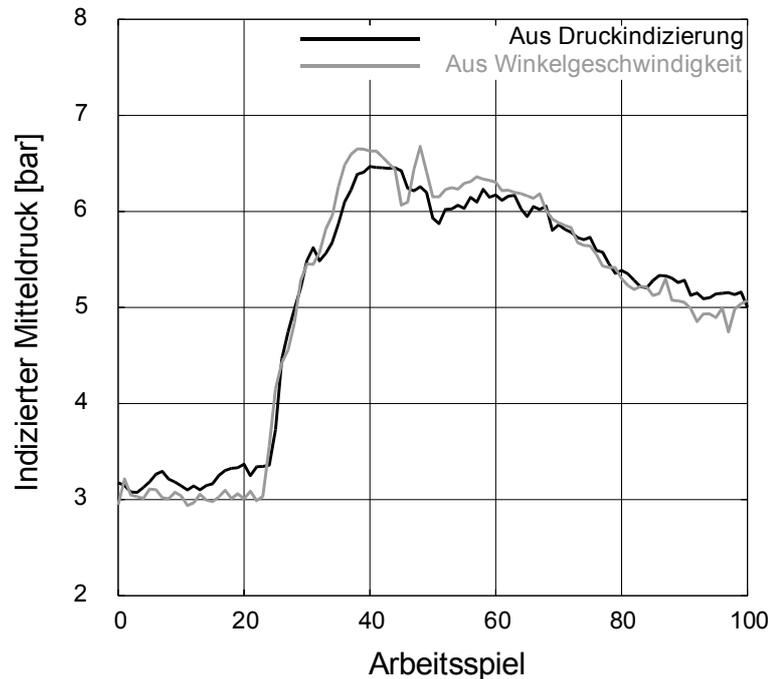


Bild 8: Modellgestützt geschätzter Verlauf des über alle Zylinder gemittelten indizierten Mitteldrucks aus der hochaufgelösten Drehzahl eines Sechs-Zylinder-Dieselmotors im Vergleich zum Verlauf des indizierten Mitteldrucks aus der Druckindizierung bei einem Instationärvorgang (Gasstoß,  $1000 \text{ min}^{-1}$ )

Die Abbildung 8 zeigt erste Ergebnisse für einen Instationärversuch, bei dem aus einer Drehzahl von  $1000 \text{ Upm}$  und einem indizierten Mitteldruck von ca. 3 bar der Mitteldruck schlagartig auf ca. 6 bar gesteigert wurde. Die Drehzahl erhöhte sich bei diesem Versuch auf  $3000 \text{ Upm}$ . Die maximale Abweichung des geschätzten Mitteldrucks gegenüber dem in einer synchron durchgeführten Druckindizierung gemessenen Mitteldrucks liegt unter  $0,5 \text{ bar}$ .

In gegenwärtige Arbeiten des Arbeitskreises werden die Prozessmodelle hinsichtlich Rechenbedarf optimiert und geeignete nichtlineare Optimierungsverfahren evaluiert.

## 10 Merkmalsbasierte Ansätze

Einen entscheidenden Einfluss auf den Motorbetrieb hat der Brennbeginn. Er beeinflusst insbesondere den Motor-Wirkungsgrad über die Lage des Verbrennungsschwerpunktes.

Als Alternative zum modellbasierten Ansatz untersucht der Arbeitskreis Verfahren zur Ermittlung des Brennbeginns aus dem nach Gleichung (2) ermittelten Gasmoment, die deutlich weniger rechentechnischen Aufwand zur Folge haben. Dafür hat es sich als vorteilhaft erwiesen, den kompressionsabhängigen Anteil aus dem Ver-

lauf des Gasmomentes zu eliminieren. Die Phasenlage des so gewonnenen Brenn-  
momentes ist ein Merkmal für den Brennbeginn.

Bild 9 zeigt im oberen Diagramm eine Messung des Brennbeginns an einem 4-  
Zylinder-Dieselmotor, wie er durch Druckverlaufsanalyse mit einem Zylinderdruck-  
Indiziersystem ermittelt wurde. Während der Messung wurde der Einspritzzeitpunkt  
sprungförmig verändert. Wie deutlich zu erkennen ist verschiebt sich der Brennbe-  
ginn entsprechend der Variation der Einspritzung. Im unteren Diagramm ist das aus  
der Phase des Brennmomentes ermittelte Brennbeginn-Merkmal aufgetragen. Dieses  
Merkmal korreliert deutlich mit dem Brennbeginn aus der Druckverlaufsanalyse.

Dieses Beispiel zeigt grundsätzlich, dass aus der Motordrehzahl ein Signal gewon-  
nen werden kann, das für eine Brennbeginn-Regelung geeignet ist. Nach derzeitigem  
Entwicklungsstand sind die Ergebnisse allerdings nur in einem eingeschränkten  
Drehzahl- und Lastbereich für eine Regelung verwertbar.

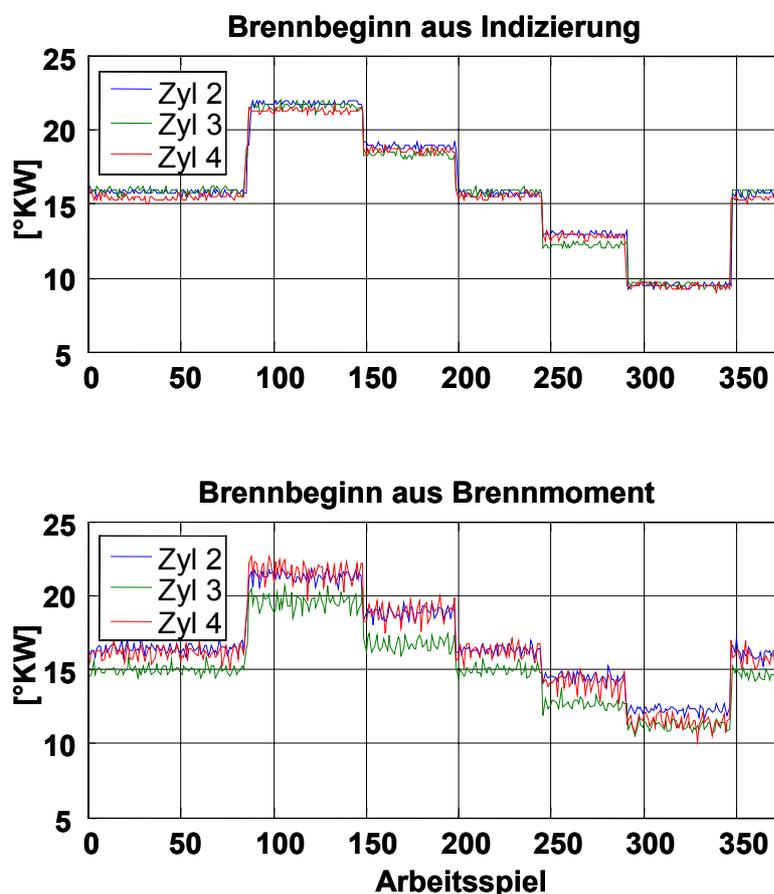


Bild 9: Brennbeginn aus der Druckindizierung (oben) und Brennbe-  
ginn-Merkmal aus Motordrehzahl (unten) an einem Vierzy-  
linder-Dieselmotor bei Variation des Einspritzzeitpunktes  
(Betriebspunkt:  $1750 \text{ min}^{-1}$ ,  $15 \text{ mg/Hub}$  Einspritzmenge)

## 11 Einfluss von Nebenmomenten

Bei der Modellierung des Gasmomentes nach Gleichung (2) werden Momente aus dem Antriebsstrang und Steuertrieb des Motors vernachlässigt. Zur Steigerung der Genauigkeit des Schätzverfahrens sollen diese Momente zukünftig berücksichtigt werden. Abbildung 10 verdeutlicht schematisch den mechanischen Aufbau eines Fahrzeugmotors.

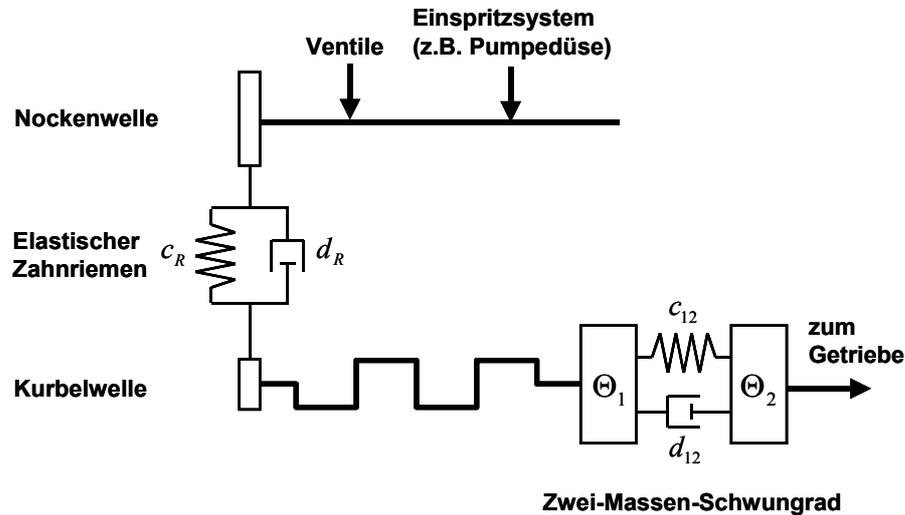


Bild 10: Modell von Steuertrieb und Antriebsstrang

Am freien Ende der Kurbelwelle befindet sich der Steuertrieb. Er dient zum Antrieb der Nockenwelle, wofür meist ein Zahnriemen zum Einsatz kommt. Verbreitet sind auch Ausführungen mit Ketten oder Stirnrädern.

Über die Nockenwelle werden die Einlass- und Auslassventile zwangsgesteuert. Bei Motoren mit Pumpe-Düse-Einspritzsystem (PDE) betätigt die Nockenwelle auch die Einspritzelemente. Die Ventile, insbesondere aber die PDE-Betätigung, erzeugen periodische Momente an der Nockenwelle. Diese Momente müssen über den elastischen Zahnriemen von der Kurbelwelle auf die Nockenwelle übertragen werden. Durch die Elastizität  $c_R$  und Dämpfung  $d_R$  des Zahnriemens handelt es sich beim Nockenwellenantrieb um ein schwingungsfähiges System, das ein frequenzabhängiges, d.h. drehzahlabhängiges Übertragungsverhalten zur Folge hat.

Bei Handschalt-Getrieben ist der Antriebsstrang über eine Kupplung mit Torsionsfeder, bei neueren Fahrzeugmodellen mit einem Zwei-Massen-Schwungrad (ZMS) an den Motor gekoppelt. Die Primärmasse  $\Theta_1$  des ZMS sitzt starr auf der Kurbelwelle und ist über gedämpfte Federelemente ( $c_{12}, d_{12}$ ) mit der Sekundärmasse  $\Theta_2$  verbunden. Aufgabe des Zwei-Massen-Schwungrades ist es, den Antriebsstrang weitgehend von der Drehunförmigkeit des Verbrennungsmotors zu entkoppeln. Auch antriebsstrangseitig bilden die elastisch gekoppelten Teilmassen ein schwingungsfähiges System mit einem drehzahlabhängigen Frequenzgang.

Zur Untersuchung der über Steuertrieb und Antriebsstrang wirkenden Nebenmomente werden bei der IAV Drehschwingungsmessungen am Versuchsfahrzeug durchgeführt. Ziel der Arbeiten ist es, mit physikalischen Modellen die Nebenmomente quantifiziert beschreiben zu können und so Ihren Einfluss auf die drehzahlbasierte Momenten- und Brennbeginn-Schätzung kompensieren zu können.

## 12 Literatur

- [1] Fehrenbach, H.; Held, W.; Zuther, F.: Drehmomentbestimmung bei Verbrennungsmotoren durch Auswertung der Kurbelwellen-Winkelgeschwindigkeit, MTZ Motorentechnische Zeitschrift, Nr. 59, 1998
- [2] Fehrenbach, H.; Hohmann, C.; Schmidt, T.; Schultalbers, W.; Rasche, H.: Verfahren zur Kompensation des Geberradfehlers im Fahrbetrieb, MTZ Motorentechnische Zeitschrift, 7-8/2002
- [3] Fehrenbach, H.; Kirsch, H.; Ruckhäberle, M.: Verfahren zur Ermittlung von Drehmomenten, Arbeiten und Leistungen an Verbrennungskraftmaschinen, Patentschrift DE 44 45 684, Fraunhofer-Gesellschaft 2000
- [4] Fehrenbach, H.; Hohmann, C.: Verfahren zur Kompensation von Geometriefehlern von Drehgeberrädern für Verbrennungsmotoren, Patentanmeldung 102 02 688.2, Fraunhofer-Gesellschaft 2002
- [5] Schmidt, Th.: Verfahren zur Kompensation des Geberradfehlers im Fahrbetrieb, Offenlegungsschrift DE 101 07 892 A1, IAV GmbH 2002
- [6] Jost, O.; Isermann, R.: Dieselmotorregelung mit Zylinderdruckmessung, Bericht zum SFB 241 „Neue integrierte mechanisch elektronische Systeme für den Maschinenbau (IMES)“, TU Darmstadt
- [7] Schmidt, Th.; Schultalbers, W.; Rasche, H.; Fehrenbach, H.: Drehzahlbasierte Bestimmung des Motormomentes, 4. Symposium „Steuerungssysteme für den Antriebsstrang“ 2003
- [8] Müller, E.; Groenendijk, A.; Raatz, T.: Homogene Dieselverbrennung - Merkmale und Entwicklungsmöglichkeiten

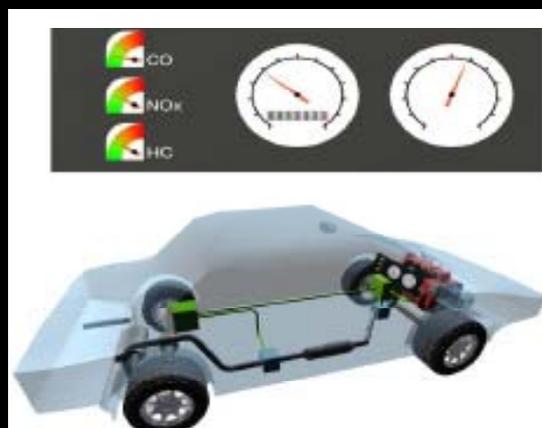


## Fibre-Optical Monitoring of Exhaust Emissions from Road Vehicles

Dr. S. Lochmann  
MSc. G. Bramann  
Dr. G. Kuka

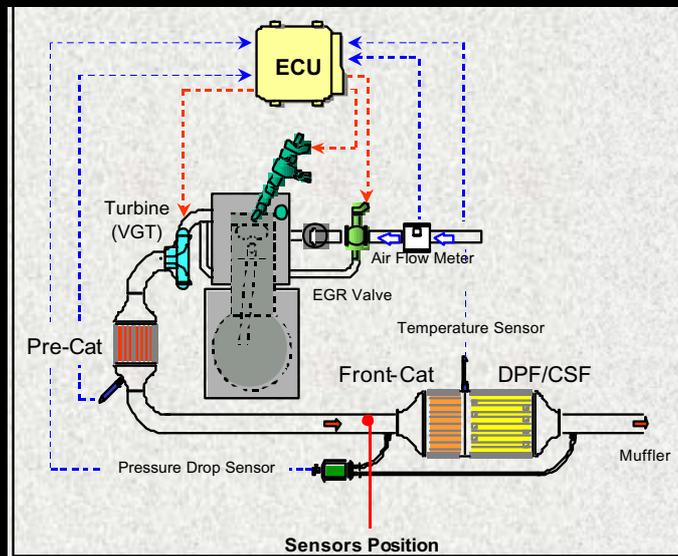
### Project: Opto-EMI-SENSE

An Optical Fibre Sensor  
Based Intelligent  
System for Monitoring  
and Control of Exhaust  
Emissions from Road  
Vehicles



## Project: Opto-EMI-SENSE

Layout of Emission Control System as installed at CRF



OPTO-EMI-SENSE

Hochschule Wismar

## Project Objective

1. Optical contaminant identification
2. Robust sensor development
3. Integration into the car information system

OPTO-EMI-SENSE

Hochschule Wismar

# Project Objective

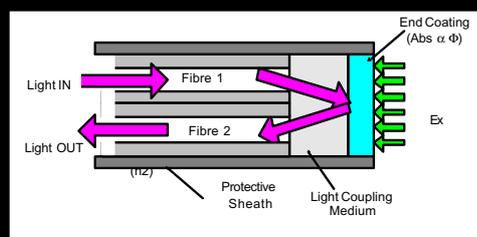
## 1. Optical contaminant identification

Nitrogen dioxide	800nm	Nitric Oxide	2.7um
Carbon dioxide	1573nm	Nitric Oxide	5.2um
Carbon Monoxide	1567nm	Nitrogen dioxide	3.4um
		Nitrogen dioxide	6.2um
		Carbon dioxide	4.3um
		Carbon Monoxide	4.7um

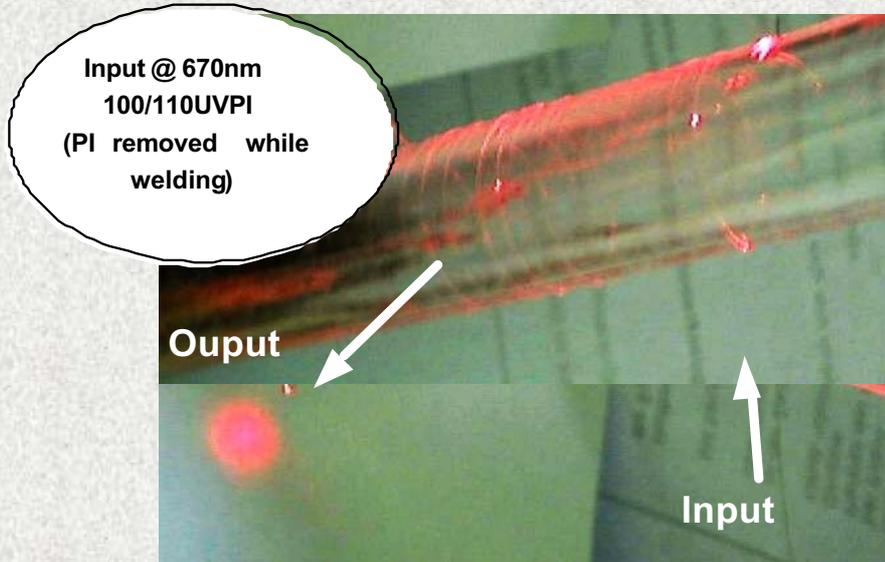
# Project Objective

## 1. Optical contaminant identification

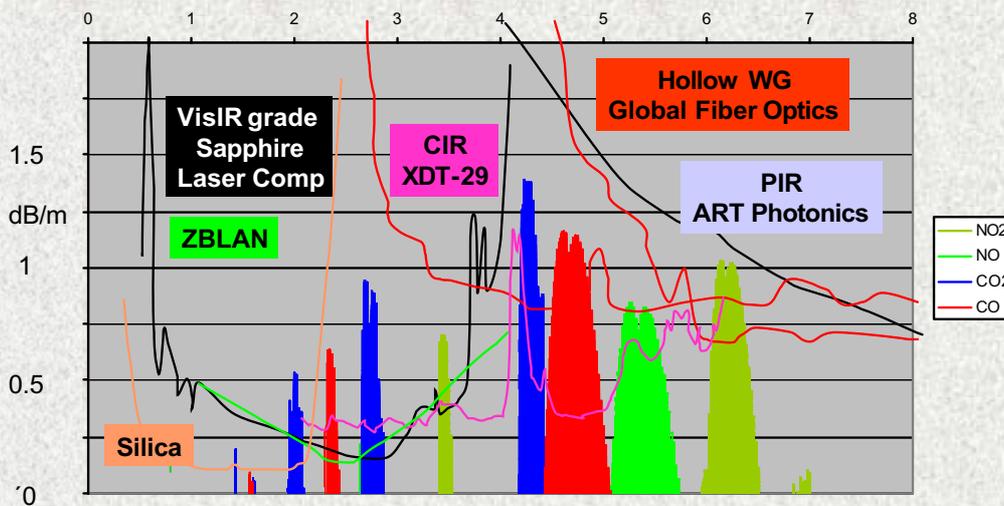
## 2. Robust sensor development



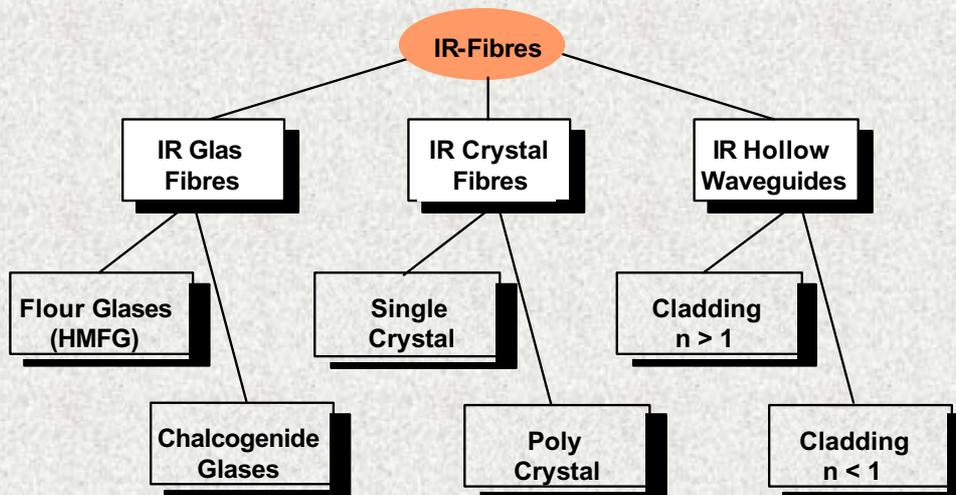
## Welded optical fibre evanescent field sensor



## Spectral Absorption and Fibre Losses



## IR Fibres ( $\lambda = 2-35\mu\text{m}$ )



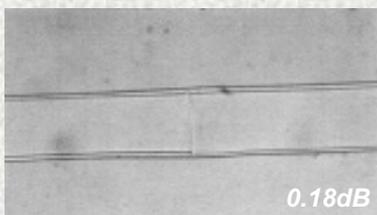
## Fibre Handling © FHG IZM Berlin

### ZBLAN Fibres



*Adapted standard cutting tools  
US support necessary!*

*Only Filament Splicers (300°C)*

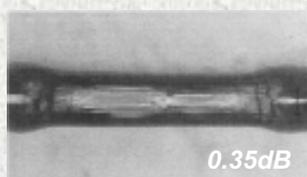


### Chalcogenite Fibres

**Cutting completely impossible!**

**Only polishing possible!**

*Only Filament Splicers (208°C)*



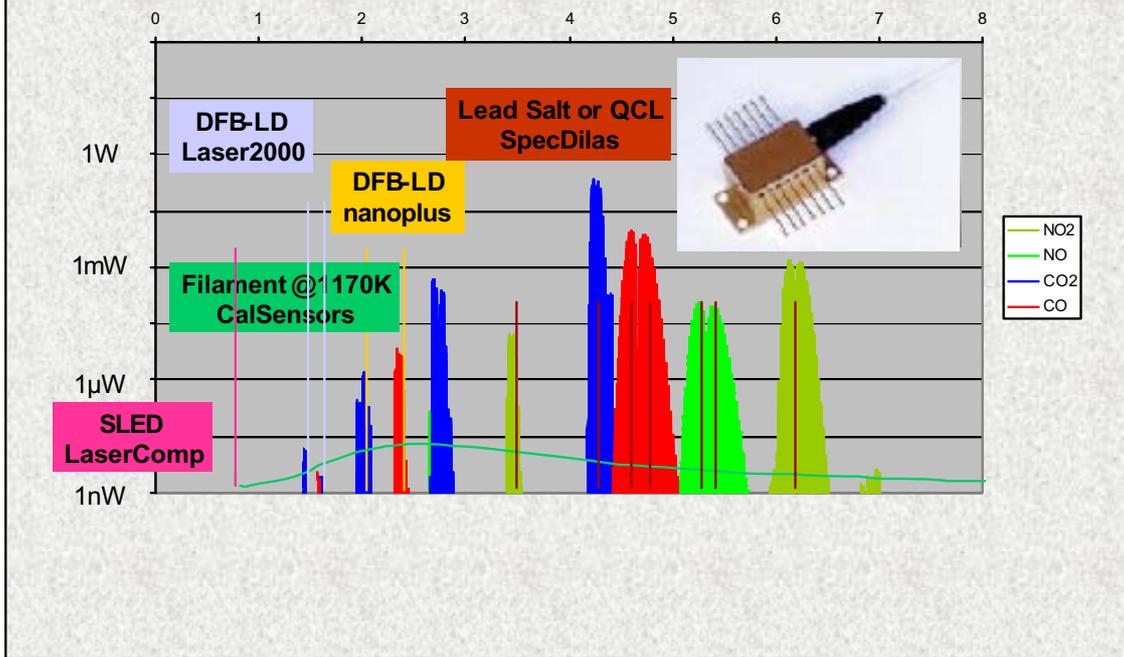
### Sapphire Fibres

**Cutting completely impossible!**

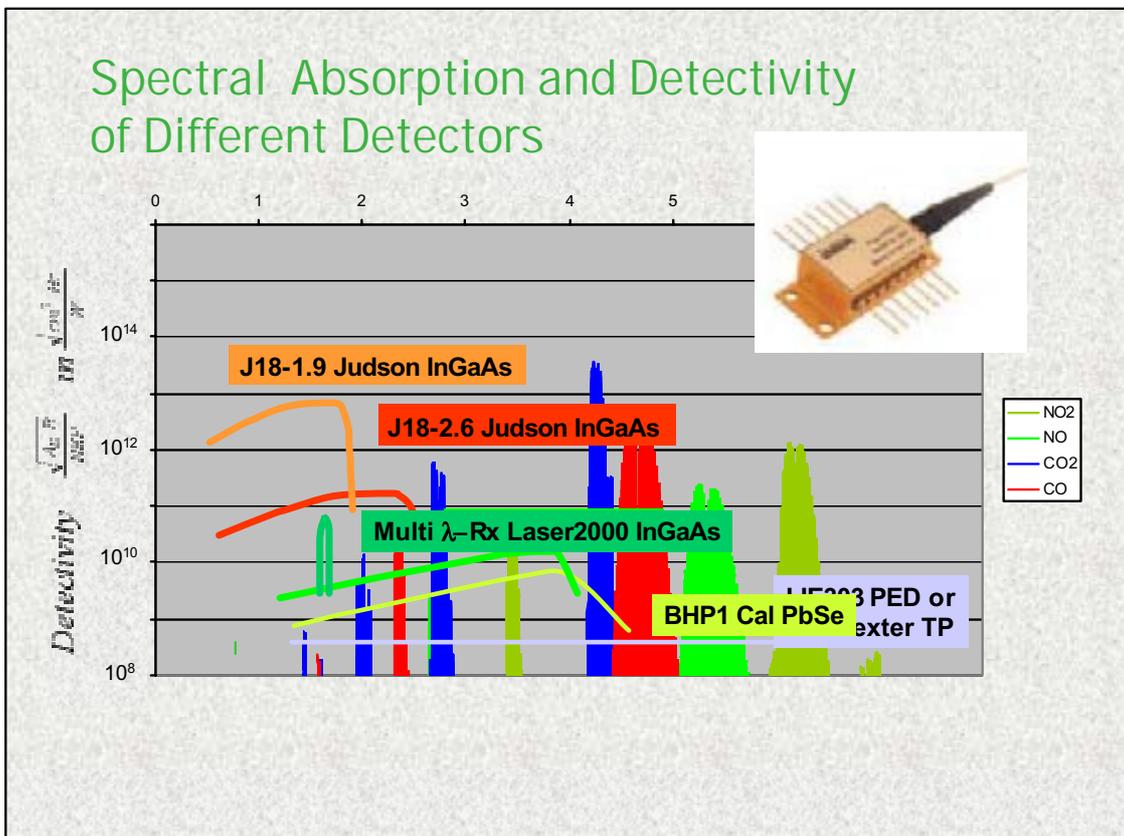
**Polishing difficult but possible!**

**Splicing completely impossible!**

## Spectral Absorption and Optical Power of Different Transmitters



## Spectral Absorption and Detectivity of Different Detectors



# **HiL-gestützte Kalibrierung mit kombinierten physikalisch/neuronalen Prozessmodellen**

Wilhelm, C.; Ayeb, M.; Winsel, Th.; Theuerkauf, H.J.,  
Institut für Elektrische Energietechnik / Antriebstechnik, Universität Kassel

## **Zusammenfassung**

Die moderne Motorenentwicklung wird einerseits durch immer kürzere Entwicklungszyklen und eine begrenzte Anzahl von Prototyp-Versuchsträgern charakterisiert, andererseits steigt die Komplexität der Fahrzeugsysteme und damit verbunden der Integrationsaufwand. Zur effizienten Unterstützung der klassischen Applikationsarbeiten, insbesondere der Anpassung der elektronischen Steuergeräte auf die jeweiligen Aggregate, wird daher die Vorausberechnung und Simulation des gesamten Antriebssystems notwendig.

Eine rein physikalisch basierte Modellierung eines Verbrennungsmotors mit einem Expertenmodell ist derzeit nicht in Echtzeit möglich. Für den Test und für die Vorbedatung des Motorsteuergerätes mit Hilfe eines Hardware-In-The-Loop (HiL)-Simulators wird jedoch ein echtzeitfähiges und hinreichend genaues Modell des möglicherweise noch nicht verfügbaren Motors und ggf. des Antriebstrangs benötigt. Hier bietet sich der Einsatz künstlicher intelligenter Systeme wie u.a. der neuronalen Netze an. In diesem Beitrag wird ein Prozessmodell vorgestellt, bei dem ein physikalisch motivierter Ansatz mit Submodellen auf Basis neuronaler Netze kombiniert wird. Geeignet trainierte neuronale Netze können das Ein-/Ausgangsverhalten des realen physikalischen Prozesses sehr genau nachbilden und in Kombination mit physikalischen Vorkenntnissen wird eine ausgezeichnete Generalisierungsfähigkeit, d.h. Reaktion auf unbekannte Messdaten, erzielt.

Damit besteht die Möglichkeit, auch extrem transiente, nichtlineare motorische Vorgänge zu beschreiben, was die Durchführung beliebiger Systemstudien gestattet. Im Rahmen dieses Vortrags wird ein solcher Ansatz vorgestellt und am Beispiel des Startverhaltens eines Referenzmotors verifiziert.

# 1 Einführung

## 1.1 Stand der Technik, Motivation und Ziele

Die Ansteuerung moderner Verbrennungsmotoren erfolgt mittels elektronischer Steuergeräte, durch die Regelfunktionen und Beherrschung transienter Vorgänge des Motors gewährleistet werden kann. Die Anpassung der Steuergeräte an den jeweiligen Motor - die Kalibrierung bzw. Applikation - geschieht heute schon in der Regel mittels HiL-Simulatoren, bei denen die realen Steuergeräte in einer Echtzeitsimulationsumgebung betrieben werden. Grundlage dazu sind echtzeitfähige Modelle, die das transiente Verhalten von Motor und Zusatzaggregaten als virtuelle Sensorsignale den Steuergeräten zur Verfügung stellen und deren Reaktion messen. Forschungsbedarf besteht bei der Modellierung transienter thermodynamischer Prozesse, wie etwa bei Beschleunigungsvorgängen oder aber bei Erwärmung eines kalten Motors (Kaltstart und Warmlauf, vgl. [ 6 ], [ 7 ], [ 9]). Hier ist man immer noch auf langwierige und kostspielige Fahrzeug- oder Prüfstandsversuche mit langen Wartezeiten angewiesen. In dieser Zeit ist teure Messtechnik und auch eine Klimazelle blockiert und nicht für andere Arbeiten nutzbar.

Daraus erwächst der Wunsch nach einem echtzeitfähigen Rechenmodell, mit dem an einem HiL-Simulator der virtuelle Versuchsträger per Mausclick in den gewünschten Startzustand gebracht wird, und so in schneller Folge unterschiedliche Datenstände des Steuergerätes bei unterschiedlichen Versuchsbedingungen getestet werden können.

In den Veröffentlichungen [ 4 ], [ 5 ], [ 3 ], [ 10 ] und [ 2 ] wurden bereits verschiedene, aber jeweils manuell von einem „Prozess-Experten“ durchzuführende Ableitungen und Realisierungen echtzeitfähiger Simulationsmodelle zur Nachbildung verbrennungsmotorischer Vorgänge behandelt. Hierbei steht die Nachbildung der sogenannten Mittelwertdynamik des Verbrennungsmotors im Vordergrund, d.h. die zeitliche Auflösung wird begrenzt auf einen repräsentativen Abtastwert pro Arbeitsspiel bzw. -zyklus. Das in [ 2 ] zur Optimierung der so genannten Beschleunigungsanreicherung verwendete echtzeitfähige Modell des Motors besteht aus der Kombination eines stationären Kennlinienmodells aus Prüfstandsmessungen am warmen Motor und eines dynamischen Modells für den Speichereffekte im Luft- und Kraftstoffpfad und dem mechanischen Beschleunigungsvorgang als quasistationäres Modell. Mit dem Kennlinienmodell konnten die durch aufwendige thermodynamische Zusammenhänge darstellbaren motorinternen Vorgänge echtzeitfähig gemacht werden. Ähnlich wird in den anderen Aufsätzen vorgegangen, nur behandeln diese das Verhalten des Verbrennungsmotors im gesamten Drehzahl-Last-

Betriebsbereich, aber wiederum nur für den betriebswarmen Motor. In dieser Arbeit wird zur Nachbildung des Kaltstarts eine teilautomatisierte Modellgenerierung vorgestellt, wobei diese Aufgabe von neuronalen Netzen erfüllt werden soll.

## 1.2 Ansatz

Hinsichtlich der beabsichtigten Echtzeitfähigkeit des Rechenmodells verbietet sich der Einsatz von ausgeprägt physikalischen Rechenverfahren, die zwar den Vorzug eines großen Gültigkeitsbereiches haben, aber aufgrund der numerisch aufwendigen Lösungsverfahren auch bei wachsender Rechenleistung heutiger Computer noch deutlich langsamer als in Echtzeit arbeiten. Der in Abb. 1 dargestellte Zielkonflikt zwischen Rechenzeit und Betrachtungstiefe kann zunächst durch eine Vereinfachung mehrdimensionaler, physikalisch orientierter Rechenmodelle entschärft werden. Weitere Vereinfachungen führen zu den sogenannten „Mittelwertmodellen“, die immer noch eine hohe Allgemeingültigkeit aufweisen.

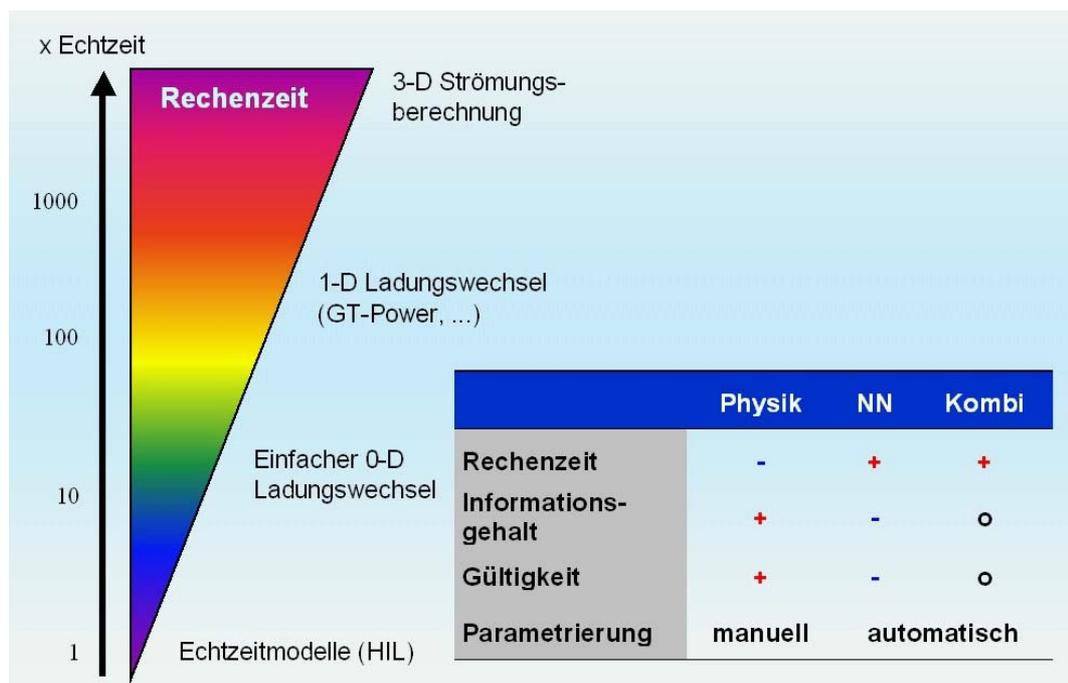


Abb. 1: Zielkonflikt zwischen Rechenzeit, Betrachtungstiefe, Gültigkeit und Erstellungsaufwand

Demgegenüber bieten Modelle, die ausschließlich auf Messdaten basieren und als reine Black-Box-Modelle aufgebaut sind, eine hohe Rechengeschwindigkeit, werden aber außerhalb des bei der Erstellung des Datenmodells abgesicherten

Parameterbereiches ungenau. Überdies werden große Mengen von Messdaten für die Parametrierung solcher Modelle benötigt. In diesem Beitrag wird eine Methode vorgeschlagen, mit der die Vorteile beider Rechenverfahren kombiniert werden.

Hierbei wird eine physikalische Modellstruktur in Teilmodelle aufgebrochen, in denen dann neuronale Netze (NN) als Approximatoren die Wirkung weniger Parameter auf den mit diesem Teilmodell abgebildeten Effekt beschreiben. Einerseits können so kompakte Netze mit verringertem Aufwand parametrierbar werden, andererseits bleibt durch die Kombination der Einzeleffekte in einer physikalischen Modellstruktur ein gewisses Maß an Allgemeingültigkeit physikalischer Modelle erhalten. Diese Art von Modelle werden Kombimodelle genannt.

### **1.3 Anforderungen**

Im Hinblick auf die spätere praktische Anwendung werden die folgenden Anforderungen an das zu entwickelnde Kombimodell für einen Verbrennungsmotor gestellt:

- Physikalische Grundstruktur des Rechenmodells mit Drehmomentberechnung aus Luft- und Kraftstoffmenge im Zylinder
- Nachbildung des quasistationären und dynamischen Motorverhaltens
- Gültigkeit für Kaltstart und Warmlauf
- Echtzeitfähigkeit
- Einfache Handhabung
- Teilautomatische Bedienung
- Verkürzung des Kalibrierprozesses

Die vorliegende Arbeit beschränkt sich zunächst auf eine Starttemperatur von 20°C.

## **2 Methodisches Vorgehen**

Im Entwicklungsprozess eines Fahrzeugantriebs werden bereits im Vorfeld der Kalibrierung vielseitige Messdaten gewonnen. Deren Nutzung als Datengrundlage für das spätere Echtzeitmodell ist wünschenswert und im realen Einsatz ohne spürbaren Mehraufwand denkbar. Im Rahmen dieser Arbeiten wird demonstriert, dass die für die Bedienung des Echtzeitmodells benötigten Messdaten mit vertretbarem Aufwand auch mit der für Fahrversuche typischen Messtechnik an Bord eines Fahrzeugs zu gewinnen sind.

Einige geforderte Zwischengrößen, z.B. die Kraftstoffmenge im Zylinder, können aber bei raschen Änderungen des Motorbetriebspunktes nicht oder nur mit großem Aufwand messtechnisch ermittelt werden, was auch für den Prüfstandsversuch gilt.

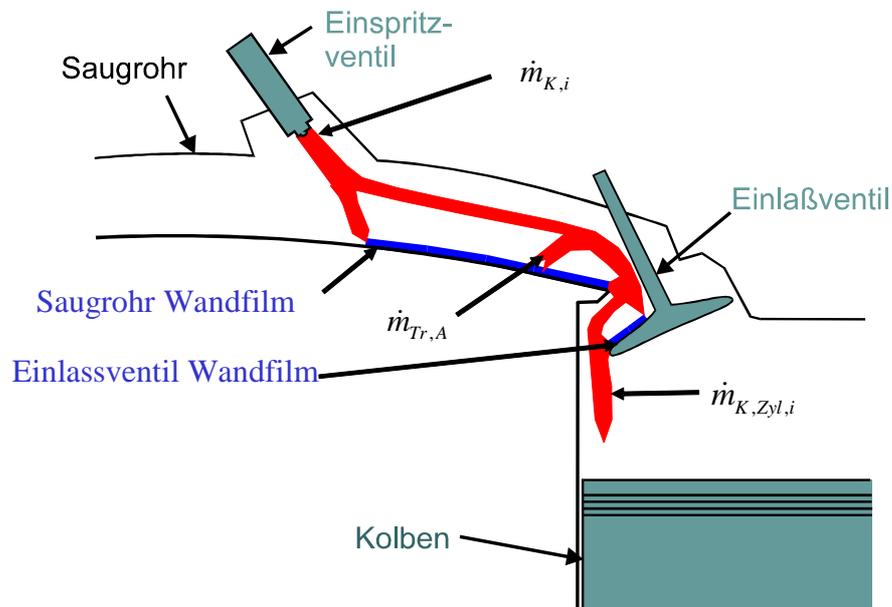


Abb. 2: Kraftstoffpfad Modell zur Nachbildung des Kraftstoffmassenflusses in die Zylinder

Deshalb wurden die dynamischen Vorgänge bei der Speicherung des eingespritzten Kraftstoffs im Wandfilm des Saugrohrs, wie in der Abb. 2 dargestellt, durch ein detailliertes physikalisches Rechenmodell (vgl. [ 6 ], [ 7 ]) des Motors berechnet. Mit Ladungswechselrechenmodellen können virtuelle Messdaten gewonnen werden, die für das Training eines dynamischen neuronalen Netzes im Kraftstoffpfad dient.

Der Einsatz eines solchen detaillierten physikalisch basierten Rechenverfahrens steht nicht im Widerspruch zu den engen zeitlichen Rahmenbedingungen einer ECU-Kalibrierung, da die zeitaufwendige Erstellung und Validierung eines solchen Rechenmodells bereits wesentlich früher im Entwicklungsprozess stattfindet. Eine wesentliche Anforderung an ein Echtzeitmodell stellt jedoch die Anpassungsmöglichkeit auf ähnliche Antriebskonzepte ohne Verwendung dieser Zwischengrößen dar.

Die im Echtzeitmodell enthaltenen neuronalen Approximatoren müssen sowohl statische als auch dynamische Ereignisse abbilden können. Hierzu werden

Messdaten benötigt, die als Trainings- bzw. Lerndaten bezeichnet werden, und die mit Hilfe von Fahrzeugversuchen oder mit Ladungswechselrechnungen erzeugt werden. Deshalb integriert das Vorgehen für die erstmalige Entwicklung eines echtzeitfähigen Neuro-Kombimodells für Kaltstart und Warmlauf experimentelle und analytische Untersuchungsmethoden.

### **3 Kombiniertes (physikalisch/neuronales) Echtzeitmodell**

Die Interpolationsfähigkeit von Neuromodellen stellt i.d.R. keine sonderliche Schwierigkeit dar, was im Vergleich zur Verwendung polynominaler Modelle bereits einen erheblichen Vorteil darstellt. Problematischer ist das Modellverhalten im Extrapolationsbereich, d.h. für Betriebsbedingungen, die gänzlich außerhalb des durch Messdaten abgedeckten Bereichs liegen. Im Gegensatz zur physikalischen Gesetzmäßigkeit ist das Verhalten eines Neuromodells außerhalb der zum Abgleich verwendeten Daten unbestimmt. Die nachfolgend vorgestellte kombinierte Modellstruktur, bestehend aus physikalisch motivierten und neuronalen Teilmodellen, stellt eine geeignete Maßnahme dar, um ein allgemeingültiges Modellverhalten - ähnlich physikalischer Modelle - zu erzielen. Neben dieser Fähigkeit zur Generalisierung ergeben sich als weitere Vorteile einer modularen, physikalisch motivierten Struktur: austauschbare bzw. wiederverwendbare Teilmodelle, physikalisch relevante Zwischengrößen und die sich hieraus ergebende Robustheit und Übersichtlichkeit bzw. Handhabbarkeit. Der aufwendige, häufig manuell durchzuführende Modellabgleich kann hierbei durch den Einsatz neuronaler Teilmodelle bzw. Verfahren automatisiert werden.

Ein echtzeitfähiges Kombimodell mit einer auf physikalischer Prozesskenntnis basierenden Grundstruktur (vgl. [ 6 ], [ 8 ], [ 9 ]) ist in Abb. 3 dargestellt, dessen Teilmodelle durch statische oder rekurrente neuronale Netze realisiert werden. Ein Modell zur Nachbildung motorischer Prozesse simuliert im Wesentlichen die Gemischbildung und den Verbrennungsprozess. Am physikalischen Vorbild orientiert sich damit auch die in Abb. 3 dargestellte Aufteilung in folgende Teilmodelle:

- Luftpfad
- Kraftstoffpfad
- Luft-/Kraftstoffgemischbildung
- Schleppmoment
- indiziertes Drehmoment

Repräsentativ für die Vorgehensweise beim Abgleich bzw. bei der Strukturierung der einzelnen Teilmodelle sei das *Luftpfad-Modell* näher erläutert: Die Parametrierung des Luftpfadmodells erfolgt unter ausschließlicher Verwendung stationärer Messdaten. Diese werden entweder direkt im Prüfstandsversuch ermittelt oder aus Messfolgen von Fahrzyklen mit dynamischen Übergangszuständen generiert.

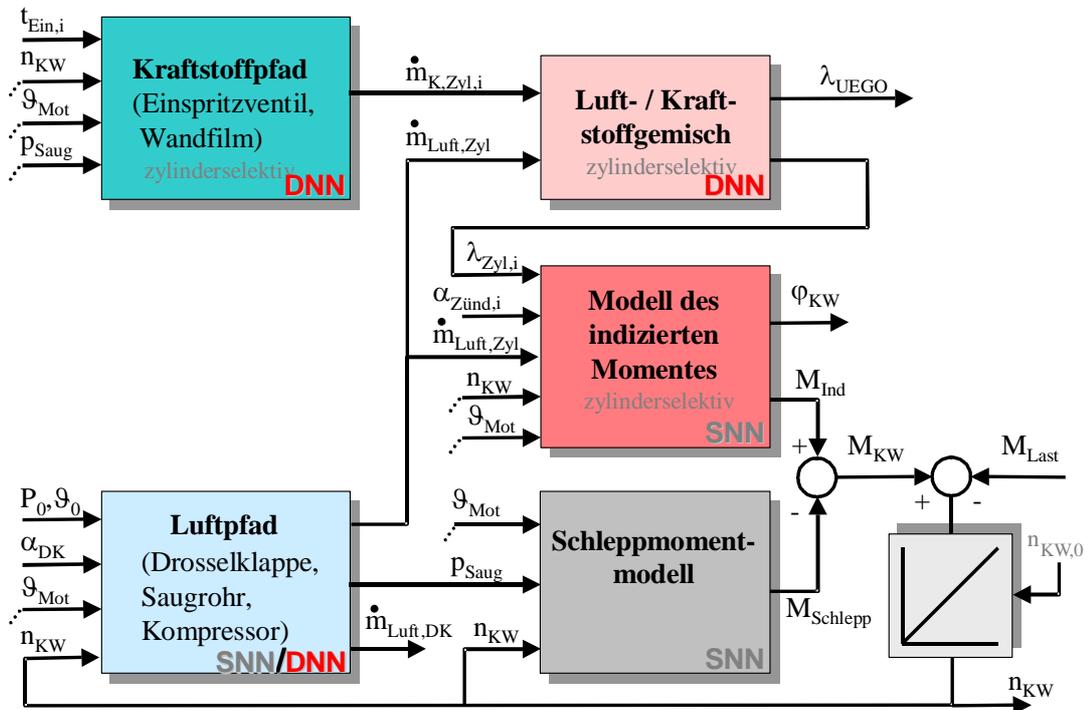


Abb. 3: Modulare Gesamtstruktur aus statischen und dynamischen neuronalen Netzen zur Nachbildung des inneren Motordrehmomentes an der Kurbelwelle

Die in Abb. 4 dargestellte Grundstruktur des echtzeitfähigen Luftpfadmodells basiert auf dem in der Literatur häufig verwendeten Saugrohr-Behältermodellansatz. Durch den Einsatz zweier statischer neuronaler Netze (SNN) konnte insbesondere die Simulationsstabilität des Luftpfad-Modells sichergestellt werden. Die üblicherweise verwendete Drosselfunktion weist im Allgemeinen Singularitäten auf, wodurch häufig ein instabiles und damit inakzeptables Simulationsverhalten beobachtet werden kann. Neben dem Luftmassenfluss in die Zylinder stehen mit dem parametrisierten Modell die virtuellen Sensorsignale Saugrohrdruck und Luftmassenfluss durch die Drosselklappe zur Verfügung. Die Gemischbildung erfolgt in einfacher Weise, anhand eines festgelegten stöchiometrischen Verhältnisses aus den Luft- und Kraftstoffmassenflüssen in die Zylinder.

**Vollautomatisierter Abgleich  
anhand stationärer Messdaten**

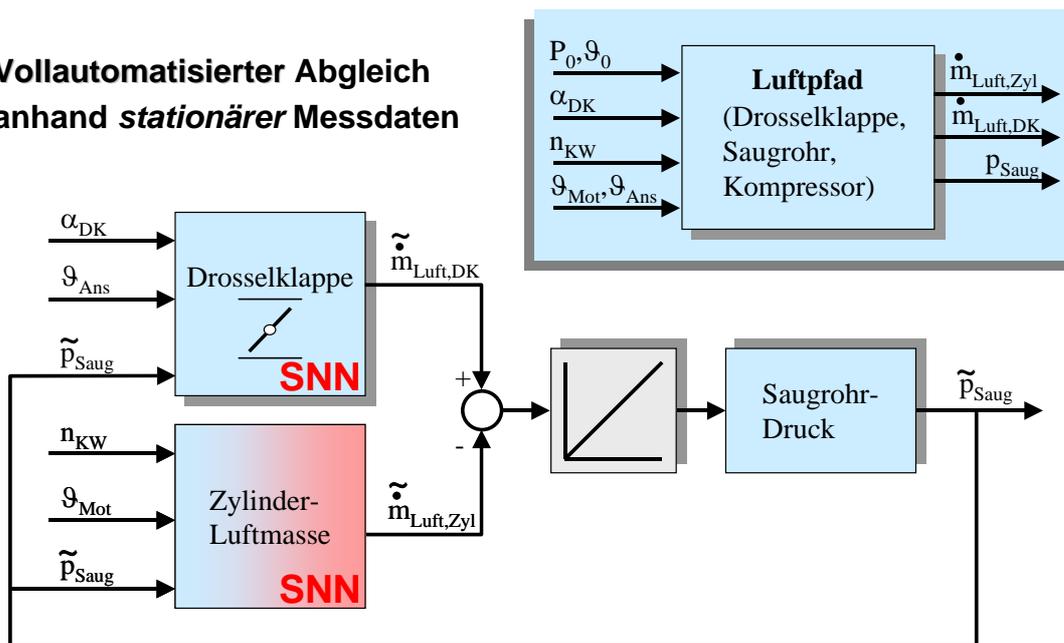


Abb. 4: Luftpfadmodell als Kombination physikalischer und statischer neuronaler Teilmodelle

**Aufteilung in die Teilprozesse:**

- Sprühnebel (im aktuellen Arbeitsspiel)
- Wandfilm (über mehrere Arbeitsspiele)

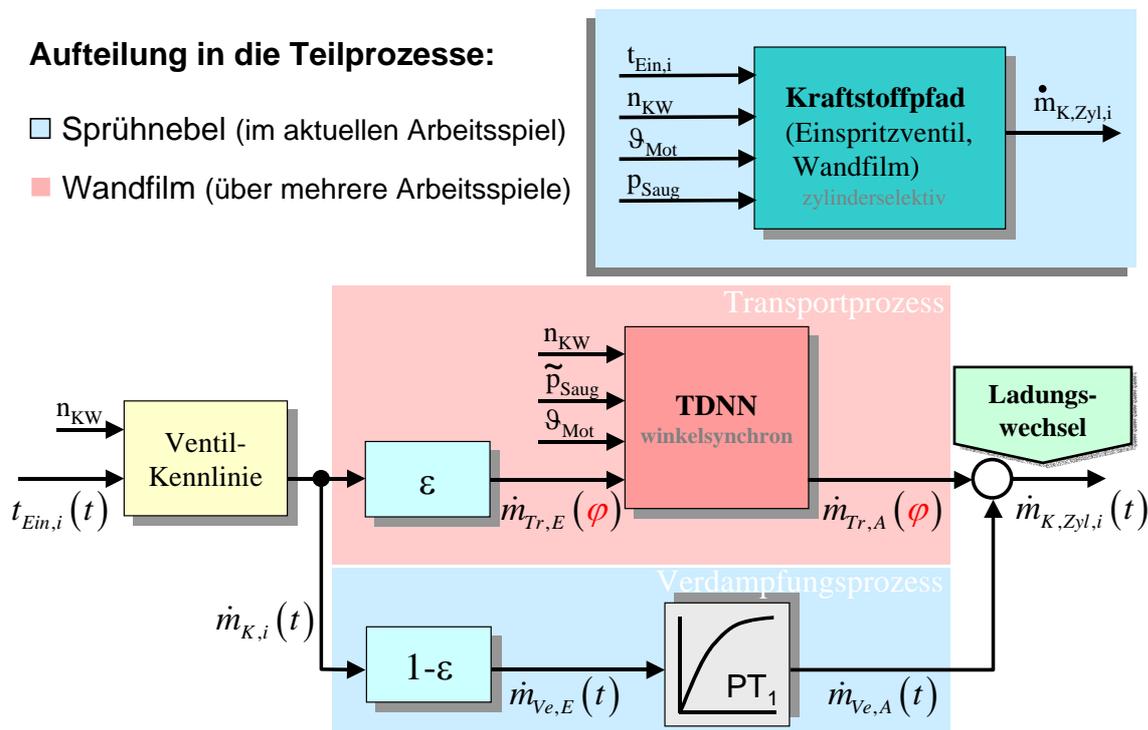


Abb. 5: Kraftstoffpfadmodell als Kombination physikalischer und dynamischer neuronaler Teilmodelle

Eine besondere Herausforderung stellt hierbei die Generierung geeigneter Sollverläufe der Kraftstoffmassenflüsse in die Zylinder dar. Anhand dieser i.d.R. nicht messbaren Verläufe erfolgt der Abgleich des, im Abb. 5 dargestellten Wandfilmmodells, welches als primärer Bestandteil des Kraftstoffpfades zu bezeichnen ist und dessen Grundstruktur auf dem Aquino-Modellansatz (vgl. [ 1 ]) basiert.

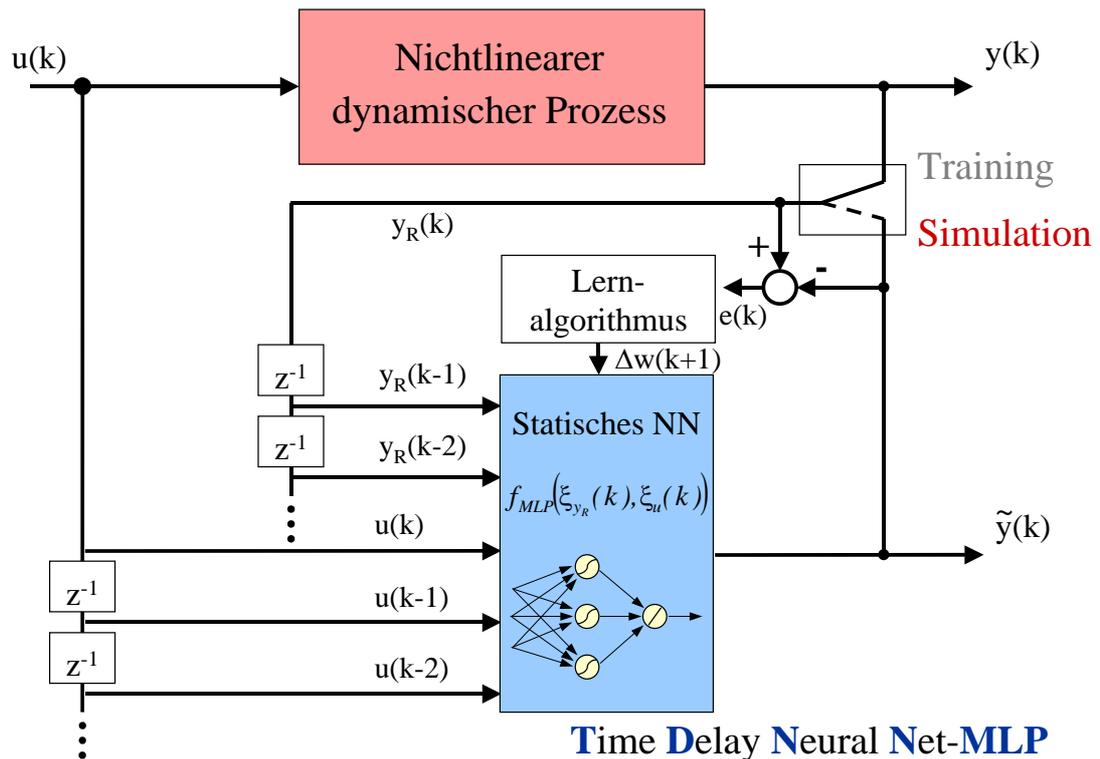


Abb. 6: Schnelles Training seriell als auch parallel zum Prozess und echtzeitfähige Simulation ohne Verwendung des Prozessausgangs, bei garantierter Stabilität

Allerdings wird hier der winkelsynchron, d.h. einmal pro Arbeitsspiel abgetastete Transportprozess mit einem rekurrenten neuronalen Netz, d.h. speziell mit dem in Abb. 6 dargestellten TDNN-MLP bei garantierter Simulationsstabilität nachgebildet (vgl. [ 8 ]). Wie in Abb. 2 und Abb. 5 dargestellt, werden die durch Einspritzung und Verdampfung verursachten zeitsynchronen Einflüsse mittels physikalisch motivierter Teilmodelle festgelegt. Somit können Eingangs- und Ausgangs-Sollwertverläufe zum Abgleich des TDNN's generiert werden. Das TDNN-MLP kann hierbei Ungenauigkeiten kompensieren, welche durch die relativ einfach strukturierten aber robusten, physikalisch motivierten Teilmodelle entstehen.

#### 4 Hardware-in-the-Loop (HiL)-Simulation

Die oben behandelten physikalisch/neuronalen Kombimodelle wurden schließlich auf einem HiL-Simulator (*CARTS*<sup>®</sup> - *Computer Aided Realtime Testsystem*) der Firma carts Real-Time Solutions GmbH implementiert und im Verbund mit einem originalen Steuergerät überprüft.

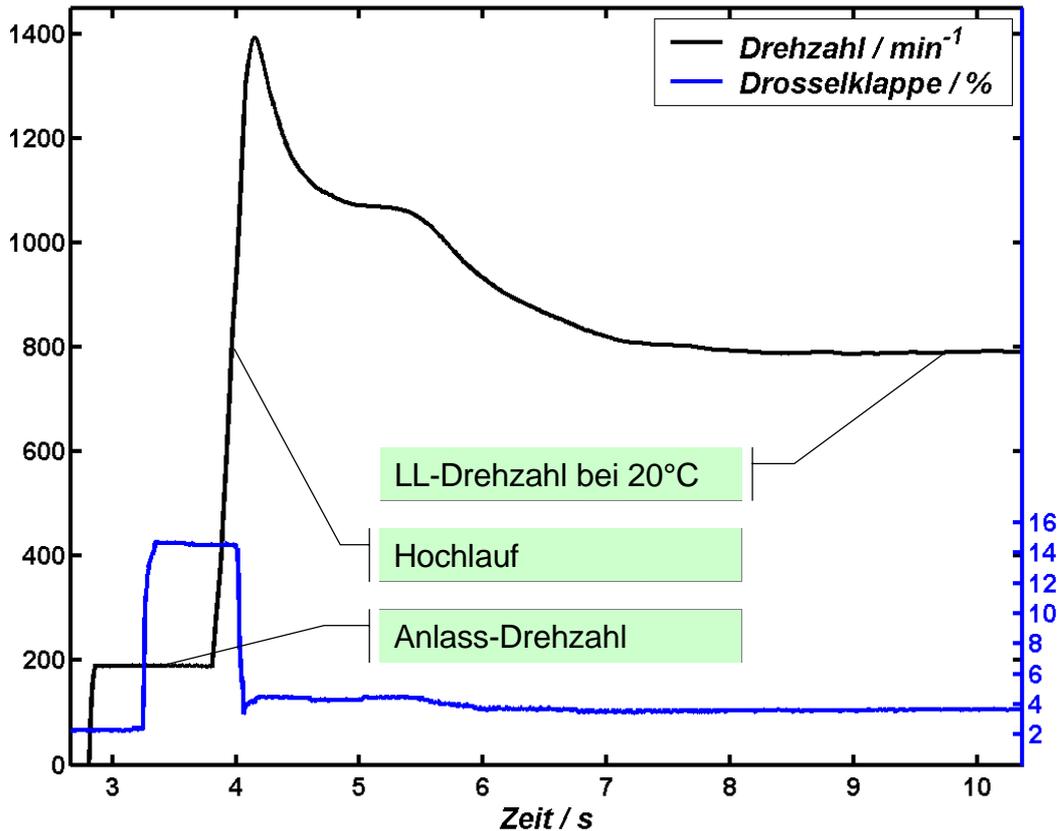


Abb. 7: HiL - Simulation eines Kaltstarts bei 20°C mit den Phasen Anschleppen, Hochlauf und Einregeln auf die Leerlaufdrehzahl

Mit diesem Aufbau können nun sehr praxisnahe Versuche durchgeführt werden, deren Umfang weit über das reine Nachrechnen von Messfahrten hinaus geht. Vielmehr lässt sich nun das Verhalten eines vollständigen Antriebstrangs im Fahrzeug in Reaktion auf beliebige Änderungen des Pedalwegs ermitteln. In Abb. 7 bis Abb. 9 sind die Verläufe verschiedener Messgrößen aus der HiL - Simulation eines Kaltstarts bei 20°C dargestellt, die mit einem CARTS-System durchgeführt wurden.

Nach dem simulierten Aufschalten der Anlasserspannung wird das Neuromodell mit Anlasserdrehzahl geschleppt und springt an, sobald ein Zylinder mit einem als zündfähig erachtetem Gemisch versorgt worden ist.

Entsprechend der Motorträgheit und dem freigegebenen Moment dreht der *Neuro-Motor* hoch und wird durch Steuergeräteeingriffe wie Zündwinkel, Drosselklappenwinkel und Einspritzmenge auf die gewünschte Leerlaufdrehzahl geregelt, solange der Pedalweggeber Leerlauf signalisiert.

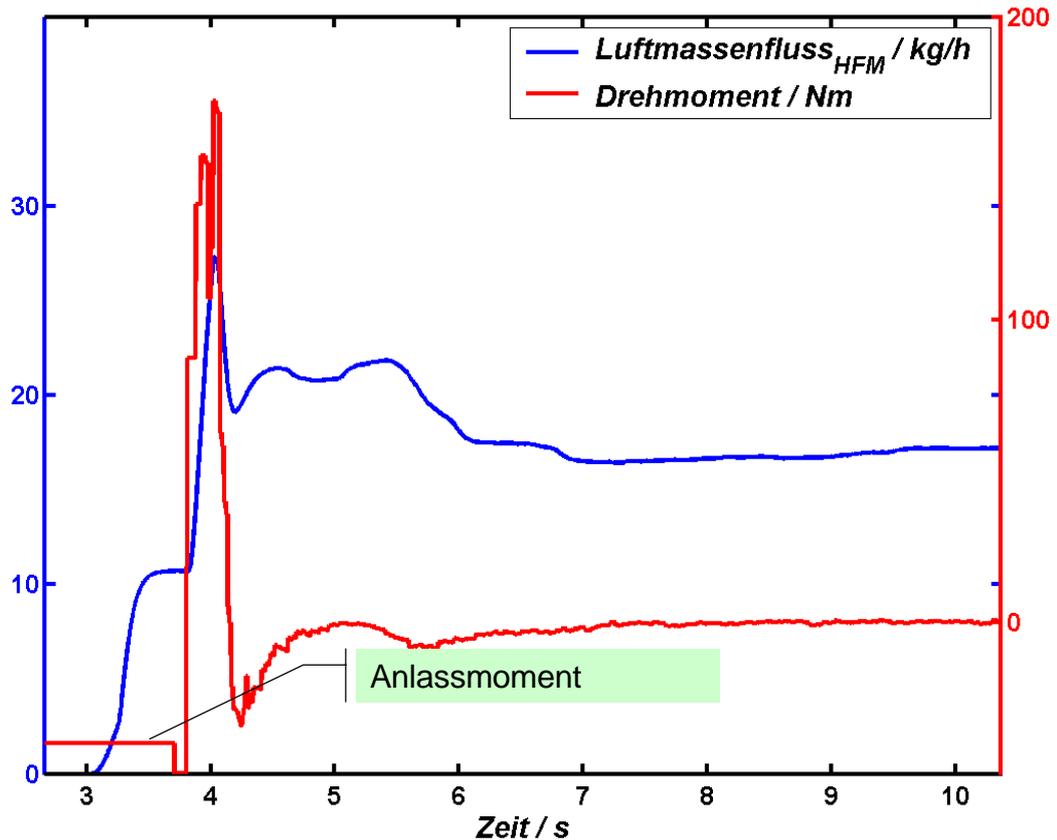


Abb. 8: HiL – simuliertes Drehmoment an der Kurbelwelle und Luftmassenfluss während des Kaltstarts aus Abb. 7

## 5 Zusammenfassung / Fazit

Mit dem Echtzeitmodell zur Nachbildung der dynamischen Vorgänge beim Start eines Verbrennungsmotors konnte bereits eine gute Genauigkeit erreicht werden. Die Erweiterung des Gültigkeitsbereichs zu niedrigeren Temperaturen, Skalierungsfragen sowie ein Emissionsmodell sind in Vorbereitung. Die konsequente Integration der dargestellten Berechnungsmethoden in die Antriebsstrangentwicklung wird zu einer deutlichen Effizienzsteigerung in der Kalibrierung führen.

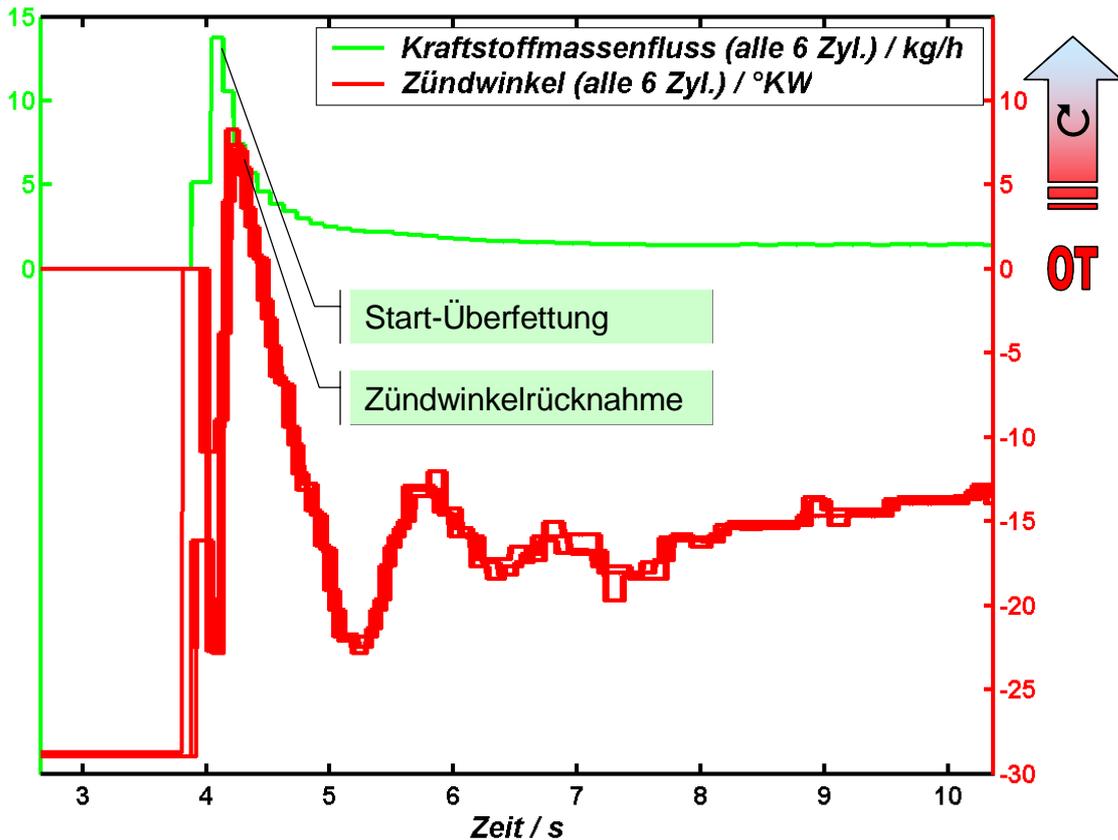


Abb. 9: Eingriffe des Motorsteuergeräts während der drei Phasen Anschleppen, Hochlauf und Einregeln auf die Leerlaufdrehzahl in Reaktion auf die HiL -Simulation des Kaltstarts

## Literatur

- [ 1 ] Aquino, C.F.:  
*Transient A/F Control Characteristics of the 5 Liter Central Fuel Injection Engine*, SAE-Paper No. 810494, 1981
- [ 2 ] Heinrich, A.:  
*Systematische Optimierung instationärer Vorgänge am Ottomotor mit Hilfe der Echtzeitsimulation*, Dissertation, Universität Kassel 1995
- [ 3 ] Lichtenthäler, D.:  
*Prozessmodelle mit integrierten neuronalen Netzen zur Echtzeit-Simulation und Diagnose von Verbrennungsmotoren*, Dissertation  
VDI-Forschritt-Berichte, Reihe 12, Nr. 454, VDI-Verlag Düsseldorf, 2001

- [ 4 ] Moskwa, J. J.:  
*Automotive Engine Modeling Real Time Control*, Dissertation  
 Massachusetts Institute of Technology Massachusetts (USA), 1988
- [ 5 ] Onder, C.-H.:  
*Modellbasierte Optimierung der Steuerung und Regelung eines  
 Automobilmotors*, Dissertation ETH Zürich, Zürich (CH), 1993
- [ 6 ] Pischinger, S; Theuerkauf, H. J.; Ayeb, M.; Lütkemeyer, G.; Schernus, C.;  
 Wilhelm, C.; Winsel, T.:  
*Erforschung eines Motormodells zur Applikationshilfe am Beispiel des  
 Kaltstart- und Warmlaufverhaltens\**  
 FVV Heft R517 (2002), Informationstagung Motoren, Nürnberg, Herbst  
 2002
- \* Das Vorhaben wird von einem Arbeitskreis unter Leitung von Dr.-Ing. P. Manz,  
 VW-AG, als Obmann begleitet. Dr. Manz und dem Arbeitskreis gebührt Dank für die  
 fachliche Unterstützung, der VFI GmbH für die Übernahme der Projektleitung und  
 den beteiligten Firmen für die finanzielle Förderung. Die Forschungsarbeiten werden  
 am Lehrstuhl für Verbrennungskraftmaschinen der RWTH Aachen (Prof. Dr.-Ing. S.  
 Pischinger) und am Institut für Elektrische Energietechnik - Antriebstechnik der  
 Universität Kassel (Prof. Dr.-Ing. H. Theuerkauf) durchgeführt.
- [ 7 ] Pischinger, Schernus, Lütkemeyer, Theuerkauf, Winsel, Ayeb:  
*Investigation of Predictive Models for Application of Engine Cold-Start  
 Behavior*, SAE International Congress, SAE Nr. 2004-01-0994, Detroit  
 (USA), 2004
- [ 8 ] Winsel, Th.:  
*Stabile neuronale Prozessmodelle: Automatisierte Generierung  
 echtzeitfähiger Modelle zur Nachbildung des dynamischen Verhaltens von  
 Verbrennungsmotoren*, Dissertation  
 VDI-Fortschritt-Berichte, Reihe 12, VDI-Verlag Düsseldorf, 2002
- [ 9 ] Winsel, Theuerkauf, Ayeb, Pischinger, Schernus, Lütkemeyer:  
*HiL-Calibration of SI Engine Cold Start and Warm-Up Using Neural Real-  
 Time Model*, SAE International Congress, SAE Nr. 2004-01-1362, Detroit  
 (USA), 2004
- [ 10 ] Woermann, R.:  
*Ein Beitrag zur Echtzeitsimulation technischer Systeme hoher Dynamik mit  
 diskreten Modellen*, Dissertation, Universität Kassel, 1994



# Verteilte Simulationen für Automotive Anwendungen

Hans-Martin Schulz, EXTESSY AG \*

## Zusammenfassung

In dem Artikel wird die Relevanz verteilter Simulationen in der modellbasierten Entwicklung für Elektroniksysteme im Automobil dargestellt. Es wird das kommerzielles Produkt EXITE zur Realisierung verteilter und hybrider Simulationen in einigen typischen Anwendungsfällen vorgestellt.

## 1 Einleitung und Motivation

Bedeutung der Simulation wird von niemandem mehr bestritten. Bekannte Effekte wie Kosten-, Zeit und Risikominimierung durch den Einsatz von Simulationen führen zu dem Wunsch, die Anwendung von Simulationen weiter auszubauen. Diesem Bedarf wird heute durch immer leistungsfähigere und vielfältigere Modellierungs- und Simulationswerkzeuge begegnet.

Eine Simulation allgemein ist die Untersuchung des dynamischen Verhaltens von modellierten Objekten durch Berechnung der Modelle mit Rechensystemen. Über das Für und Wider und die vielfältigsten Anwendungen von Simulationen sei an dieser Stelle nicht eingegangen. Fest steht, dass alle Simulationen eine Reihe von Qualitätsmerkmalen erfüllen müssen, um den erwarteten Nutzen in der praktischen Anwendung generieren zu können.

## 2 Qualitätsmerkmale von Simulationen

Die Qualität von Simulationen wird durch die Modelle als auch die Simulationsberechnung bestimmt. Während ersteres in erster Linie vom Modellierer abhängig ist, wird der zweite Aspekt durch den Werkzeughersteller des Simulationswerkzeugs vorrangig bestimmt.

Die Forderung Exaktheit des zu berechnenden Modells wird durch die zugrunde liegenden mathematischen Verfahren bestimmt. Exaktheit des Modells bezieht sich auf die Widerspiegelung des realen Objektes entsprechend dem Untersuchungsziel und seine Angemessenheit für die durchzuführende Simulation, bspw. der zeitlichen Rahmenbedingungen für die Simulation

---

\*InnovationsCampus, Major-Hirst-Str. 11, 38442 Wolfsburg, Germany, h.schulz@extessy.com

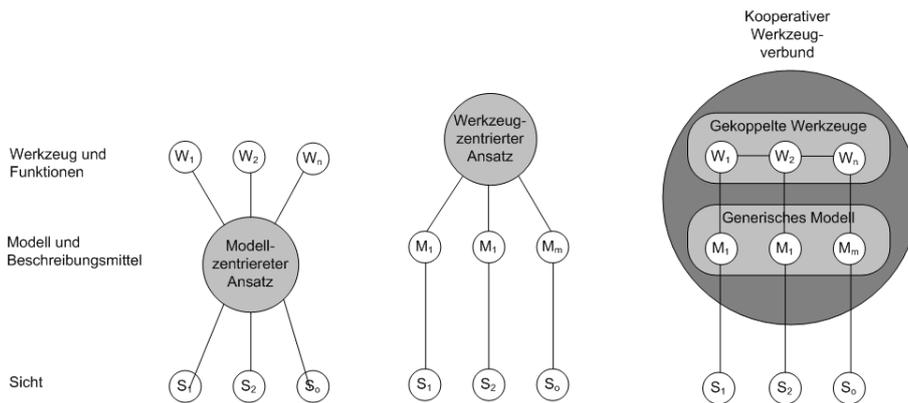


Abbildung 1: Kopplungsebenen: Sicht, Modell oder Werkzeug (aus [BS02])

Durch die Berechnung auf IT-Systemen wird wiederum eine numerische Näherung der Modelle durchgeführt, welche die Exaktheit zum realen Objekt weiter reduziert. Ein wesentlicher Einflussfaktor ist hierbei verfügbare Rechenleistung, da diese die bei einer akzeptablen Rechenzeit erzielbare Näherung bestimmt.

Die Usability wird durch eine Vielzahl, oftmals sehr individuelle Faktoren bestimmt. Essenziell sind aber neben der oben genannten Rechenleistung die Verfügbarkeit geeigneter Modellierungswerkzeuge und Simulationsumgebungen. Die Werkzeuge müssen eine effiziente und fehlerfreie Modellierung ermöglichen und die Simulationsumgebung soll eine Simulation ohne zusätzliche Aufwendungen ermöglichen, d.h. ohne umfassende Arbeiten nur zur Vorbereitung und Durchführung der Simulation auskommen.

In der industriellen Praxis ist insbesondere die Verfügbarkeit der Modelle ein wesentliches Entscheidungskriterium. Modelle im notwendigen Umfang stellen für sich bedeutende Investitionen dar, so dass eingeführte Werkzeuge und damit erstellte Modelle die Basis der modellbasierten Entwicklung sind. Im Automotive gibt es einen wachsenden Markt für Modelle, welcher sich neben den Werkzeugen etabliert hat. Diese Angebote effektiv zu nutzen, ohne die Flexibilität für neue Modellierungstechnologien zu verlieren ist sicherlich ein wettbewerbsrelevanter Vorteil für jeden Anwender von Modellen und Simulationen im Automotive.

### 3 Verteilte Simulation

#### 3.1 Integration und Verteilung von Simulationen

Im Automotive zu simulierende Systeme stellen enorme Anforderungen hinsichtlich der obigen Qualitätskriterien. Ursachen sind sowohl die unterschiedlichen Abstraktionsebene der Modelle aufgrund des tief gestaffelten Entwicklungsprozesses, der Gesamtumfang der Systeme sowie die Vielfältigkeit der zu modellierenden Aspekte (bspw. zeitkontinuierliches vs. zeitdiskretes Verhalten).

Um allen diesen Anforderungen gerecht zu werden, d.h. alle notwendigen Sichten und Funktionen abzubilden, lassen sich verschiedene Ansätze einer Werkzeugkopplung anwenden (vgl. [BS02]), wovon die Integration von Simulationen eine Variante ist.

**Modellintegrierender Ansatz** Alle Sichten und Funktionen werden auf ein Modell überführt. Werden verschiedenen Werkzeuge für die Modellierung verschiedener Sichten eingesetzt, so müssen diese gemeinsam das gleiche Modell unterstützen.

**Funktions- und werkzeugintegrierender Ansatz** Die unterschiedlichen Sichten werden in getrennten Modellen erfasst, die gemeinsam in einem Werkzeug weiterverarbeitet werden. Beispiele hierfür sind Simulink<sup>1</sup> oder UML-Werkzeuge, die verschiedenste Modelle, bspw. Zustandsautomaten und Blockschaltbilder bzw. Sequenzdiagramme, in gemeinsam verarbeiten.

**Kooperativer Werkzeugverbund** Die Integration der Modelle erfolgt auf Werkzeugebene durch die Kopplung der Werkzeuge. Die darzustellenden Sichten verbleiben jedoch in den einzelnen Werkzeugen.

Im Automotive-Context führen die ersteren beiden Varianten zu nicht hinnehmbaren Einschränkungen der Modellierungsmächtigkeit bzw. zu Einschränkungen in der Nutzbarkeit der Modelle. Beispiele hierfür sind:

- In der UML gibt es keine Notation für zeitkontinuierliches Verhalten und Signale.
- Hardwarenahe Sprachen (SystemC, VHDL etc.) erlauben sowohl die Modellierung diskreten und zeitkontinuierlichen Verhaltens, sind aber hinsichtlich der Berechnungszeiten vergleichsweise langsam und lassen keine abstrakte Modellierung zu.

Im Hinblick auf die oben genannten Qualitätskriterien für Simulationen erscheint der Ansatz des Kooperativen Werkzeugverbundes als sehr vielversprechend und wurde zur Grundlage der Entwicklung des Produktes **EXITE**.

### 3.2 Zielstellungen der Verteilten Simulation

Da die mathematische Grundlage der einzelnen Werkzeuge fest steht, konzentrieren sich die Werkzeughersteller insbesondere auf Verbesserungen im Sinne der

- Erleichterung der Erstellung der Modelle durch
  - Prozessunterstützung (Teamwork / verteilte Entwicklung) und
  - neue Notationen (bspw. Automotive Profile der UML oder SysML[sys04]), des
- Handlings der Modelle hinsichtlich
  - Partitionierung
  - Nutzung der jeweils optimalen Modellierung für den zu untersuchenden Teilaspekt des Gesamtmodells

---

<sup>1</sup>® The MathWorks Inc.

- Effiziente Simulationsvorbereitung und schnelle Durchführung der Berechnung

In diesem Sinne kann die verteilte Simulation mit **EXITE** als eine Ausbaustufe entsprechend folgender Ausbaustufen einer Werkzeugumgebung, hier beispielhaft Simulink, dargestellt werden.

Level	Product / Module	Focus of improvement	Comment
0	Simulink	Model based development	General purpose modelling and simulation
1	Simulink accelerator	Simulation speed	
2	Simulink block libraries	Modelling process	Definition of Re-Usesable components, which can be edited independently from the models that use them
3	RTW	Simulation speed	Compiled code executes much faster.
4	Model referencing	Modelling process	Separately compiled components reduce compilation and allow distributed engineering with information hiding.
5	Distributed Simulation Toolbox	Simulation speed + Modelling Process	First Level of <b>EXITE</b> : Distributed, parallel simulation/modelling with Simulink models.
6	<b>EXITE</b>	Simulation speed + Modelling Process	Fastest parallel simulation and hybrid modelling: Any models can be connected.

### 3.2.1 Paradigmen

Zur Realisierung einer verteilten Simulation sind 5 wesentliche Paradigmen zu erfüllen, welche hier nur insofern kurz genannt werden, als sie für die Wahl der Architektur der Co-Simulation **EXITE** von Bedeutung sind. Weitergehende Informationen siehe [BS02].

**Kooperation** Die Kooperation bezeichnet die zu übergeordneten Steuerungsaufgaben erforderlichen Aufgaben der Werkzeugkopplung. Sie dient dazu, die in den verteilten Werkzeugen verfügbaren Funktionen in einer zweckbestimmten, koordinierten Art und Weise dem Nutzer verfügbar zu machen.

**Kommunikation** Die Kommunikation zwischen den beteiligten Simulationen ist die Basis für die darauf aufbauende Synchronisation und Koordination.

**Synchronisation** Die Synchronisation stellt sicher, dass in der Kommunikation keine Nachrichten verloren gehen.

**Koordination** Die Koordination bezeichnet die zeitliche Abstimmung der an der Co-Simulation beteiligten Simulationen bezüglich der statischen und dynamischen Konsistenz der beteiligten Modelle.

**Information** Die Information bezeichnet das Bereitstellen der zu kommunizierenden Information in einem für die jeweilige Simulation geeigneten Format und Schnittstelle.

### 3.3 EXITE

Die Wahl der Architektur einer Lösung für einer Werkzeugkopplung mit dem Ziel der verteilten Simulation muss eine Reihe von Kompromissen eingehen, die hier kurz umrissen werden.

Aus Sicht der Simulation betreffen die Kompromisse die 4 Dimensionen Performanz, Mächtigkeit des Protokolls, Flexibilität hinsichtlich der Topologie der aufzubauenden verteilten Simulation und funktionaler Einfluss der Kopplung, letzteres insbesondere bedingt durch die Exaktheit der Abtastung zeitkontinuierlicher Signale zwischen den Simulationsteilnehmern.

Bei der Erstellung von EXITE standen die Anforderungen im Mittelpunkt, die in einer von MATLAB/Simulink<sup>2</sup> dominierten Welt vorrangig sind.

EXITE besteht aus einer Reihe von Komponenten, welche die geforderten Paradigmen erfüllen. Zentrale Komponente ist der EXITE-Kernel. Er realisiert für die Co-Simulation die Synchronisation und Koordination. Er baut auf einer Middleware auf, welche die Kommunikation liefert. Für EXITE wird meist CORBA<sup>3</sup> verwendet, wodurch nicht nur die Co-Simulation auf einem Rechner sondern auch eine verteilte Simulation über Rechnergrenzen hinweg möglich wird.

Die Plug-Ins für die verschiedenen Simulationswerkzeuge realisieren einerseits die Information als auch teilweise die Kooperation, soweit das verbundene Simulationswerkzeug eine externe Steuerung erlaubt.

Für die Funktionsweise von zentraler Bedeutung ist die EXITE-API. Diese geht zugunsten einer sehr hohen Performanz nur von einer Art der zu übertragenden Daten aus kontinuierlichen Signalen, die in äquidistanten Zeitschritten zwischen den beteiligten Modellen im Wert abgeglichen und zeitlich synchronisiert werden. Alle Datentypen werden in einer einheitlichen Repräsentation (als double) übertragen.

Des weiteren ist die Architektur von EXITE so ausgelegt, dass die zeitliche Synchronisation und die Kommunikation miteinander integriert sind, was ein einheitliches,

---

<sup>2</sup>® The MathWorks Inc.

<sup>3</sup>Common Object Request Broker Architecture, ® Object Management Group

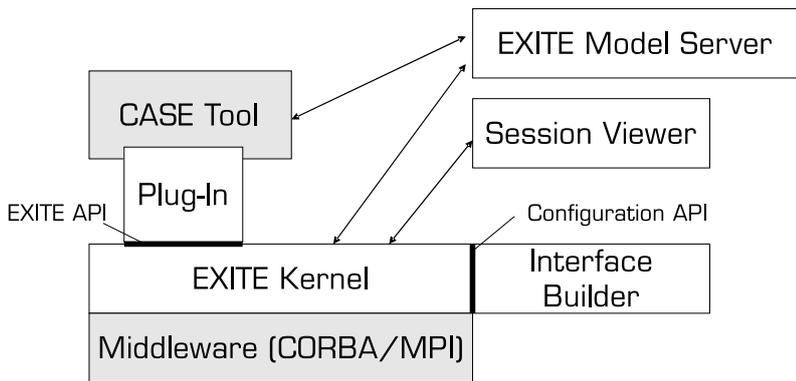


Abbildung 2: Komponenten der **EXITE**-Architektur

schlankes Protokoll auf Ebene des Kernels sicherstellt. Weiterhin erfolgt so die Synchronisation immer paarweise zwischen den Modellen, so dass keine zentrale Instanz zu einem Flaschenhals für die Performanz führen kann.

In der Folge müssen nun aber alle beteiligten Simulationen gemeinsam eine Baumstruktur formen (Simulationen als Knoten und **EXITE**-Verbindungen als Kanten des Baumes), um Verklemmungen zu vermeiden, die bei der Synchronisation im Falle von Schleifen in der Systemstruktur entstehen können. Gepaart mit dem Fakt, dass jede Abtastung eines zeitkontinuierlichen Signals zu einem Fehler führt, summieren sich diese bei tief gestaffelten Hierarchien verteilter Modelle auf, was zu funktionalen Rückwirkungen auf die Simulation haben kann.

Um einen solchen Synchronisationsfehler auszuschließen und trotzdem die Funktionen hinsichtlich hybrider und verteilter Simulation nutzen zu können, kann die **EXITE**-API auf sequenziellen Betrieb umgestellt werden, d.h. zwei Modelle kommunizieren im Sinne eines „function calls“ und führen die Berechnungen sequenziell, nicht parallel aus.

Für eine Co-Simulationen sind neben **EXITE** auch alternative Lösungen verfügbar. Der bekannteste Ansatz ist in der HLA (High Level Architecture)[CW04] zu finden. HLA ist ein Standard des DoD und definiert eine Architektur zur Modellierung und Simulation. Zentraler Bestandteil ist die RTI (Run Time Infrastructure) Interface Definition, welche die notwendigen Funktionen zu einer allgemeinen, verteilten Simulation spezifiziert. Sowohl die möglichen Typen der ausgetauschten Informationen sind völlig und auch das dynamische Kommunikationsverhalten ist sehr flexibel. Es lassen sich so beliebige Simulationen koppeln.

## 4 Beispielhafte Anwendungen

### 4.1 Überwindung von Abstraktionsstufen

Im Verlaufe eines Entwicklungsprozesses laufen parallele Entwicklungsströme, bspw. der einzelnen Steuergeräte, der Testumgebungen und -modelle sowie der Umgebungsmodelle, vgl. Abb. 3. In diesem Kontext besteht für die Durchführung von Simulationen die Herausforderung, Modelle unterschiedlicher Abstraktionsebenen miteinander zu vernetz-

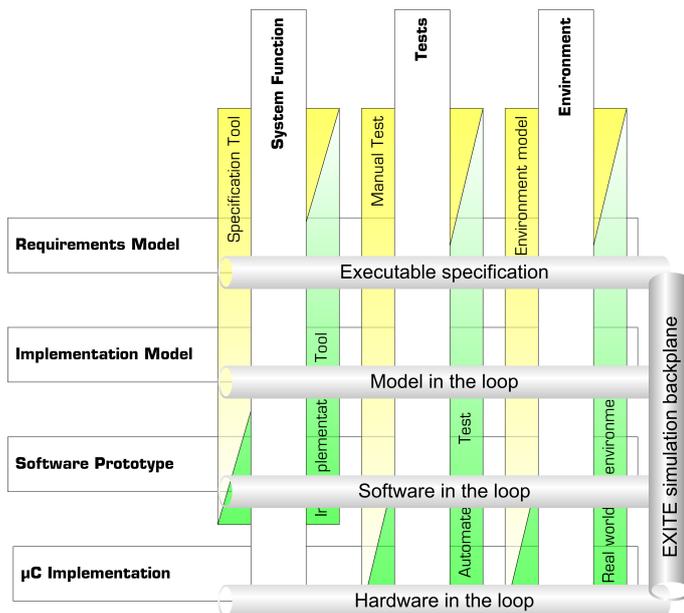


Abbildung 3: Co-Simulation als Verbindung von Abstraktionen und Domänen

ten.

Die Systemgrenzen und Schnittstellen sind durch die gegebene Anwendung definiert und lassen sich in Form von Schnittstellenklassen in **EXITE** definieren, welche dann von den unterschiedlichen Modellen genutzt werden können. So lassen sich die einzelnen Komponenten eines Systems schrittweise und unabhängig voneinander entwickeln und in einer einheitlichen, stabilen Experimentierumgebung in Form einer Co-Simulation simulativ untersuchen und validieren.

In der aktuellen Praxis erfolgt diese vollständige Integration erst in der Hardware-in-the-Loop-Simulation, welche mit bedeutenden Aufwendungen für die Simulationsimplementierung und Einschränkungen für die Simulationsdurchführung behaftet ist, da alle Systemteile die Ausführungsbedingungen und Codegenerierung für den HIL<sup>4</sup> unterstützen müssen.

## 4.2 Hochkomplexe Gesamtfahrzeugsimulationen

Die zunehmende Verfügbarkeit von Simulationsmodellen nahezu aller Komponenten eines Fahrzeugs lässt die Realisierung kompletter, virtueller Fahrzeuge in Simulationen zu einem schlagkräftigen Werkzeug werden. Um aufwendige, mit Hardwareaufbauten verbundene Tests ablösen zu können, müssen alle beteiligten Simulationsmodelle eine die Extrapolation der Untersuchungsergebnisse auf das reale System erlaubende Exaktheit aufweisen. Neben den hochkomplexen, physikalischen Modellen eines Gesamtfahrzeugs betrifft das in zunehmendem Umfang auch die Abbildung seiner umfangreichen, verteilten Steuerungssysteme. Hier bedeutet Realitätsnähe oftmals die Forderung nach der Nutzung der implementierungsnaher Modelle der Zielsoftware der Steuergeräte.

<sup>4</sup>Hardware-in-the-Loop Simulator

Die Berechnung solcher Modelle erfolgt in der Praxis am HIL bzw. durch Zusammenführung der Modelle auf einem PC, meist durch das Einbinden externen Codes aus Teilmodellen in ein Mastermodell. So aufgebaute, umfangreiche Gesamtsimulationen lassen sich nur durch umfangreiche Maßnahmen im Finetuning der Modelle in akzeptabler Zeit berechnen.

Die parallele Simulation bietet hier die Möglichkeit, nahezu beliebige Ressourcen für die Simulation parallel zu nutzen und so die Simulationszeit bedeutend zu verringern. Der Kommunikationsaufwand zwischen den vernetzten Rechnern steht dem entgegen, kann jedoch durch eine angemessene Partitionierung der Verteilung der Modelle auf die Rechner soweit reduziert werden, dass er nicht zu substantiellen Performance-Verlusten führt. In der Praxis hat sich eine Verteilung entsprechend der Struktur der vernetzten Steuergeräte im Automobil bewährt.

Die Architektur von EXITE ist so gewählt, dass es in der Kommunikation keinen Flaschenhals gibt, d.h. weder im Datenaustausch noch in der Synchronisation der vernetzten Modelle.

### **4.3 Beherrschung von Architekturen**

In der zunehmende Komplexität der elektronischen Systeme in Fahrzeugen verlangt nach neuen Ansätzen zur Beherrschung der komplexen Architekturen. Schlagwörter sind hierbei eine flexible Funktionspartitionierung und die Bewertung von Fahrzeugarchitekturen. In diesem Zusammenhang werden verschiedene Werkzeuge zur getrennten Modellierung von Software- und Hardware-Architektur und dem darauf aufbauenden Mapping zwischen diesen eingeführt.

Eine Bewertung solcher Fahrzeugarchitekturen kann nun sowohl anhand statischer Berechnungen als auch simulativ erfolgen. Ersteres Verfahren erlaubt bereits in frühen Projektphasen Näherungsergebnisse hinsichtlich nicht-funktionaler Aspekte, bspw. Kosten und Gewicht der elektronischen Gesamtsystems, als auch funktionaler Aspekte, bspw. der mittleren Buslast, maximaler Antwortzeiten [Kie96]. Die Simulation kann dann zur Validation und genaueren Betrachtung einzelner funktionaler Fragestellungen dienen.

Während die statische Berechnungen für komplexeste Systeme durchgeführt werden können, stellen sich für deren Simulation wachsende Anforderungen, für die eine verteilte Simulation einen bedeutenden Performanzgewinn darstellen kann, wenn der Fakt der Abbildung verteilter Systeme die Komplexität dominiert. Im Fall einer geschichteten Simulation einzelner Komponenten, bspw. der Ausführung von Software auf simulierter Hardware, läßt einer Verteilung keine Perfomanzsteigerung erwarten.

### **4.4 Know-How-Schutz**

Der Umfang der Software in den Steuergeräten verlangt nach der Nutzung verteilter Entwicklungsressourcen, welche durch Zusammenarbeit mehrerer Firmen realisiert wird. Neue Softwareplattformen auf Steuergeräten unterstützen dies durch eine Modularisierung und Kapselung der einzelnen Softwarekomponenten. Bei der Erprobung solcher Software ist auf den Schutz der Intellectual Property der Beteiligten zu achten. Insbesondere ist in der Regel kein Austausch von Modellen untereinander möglich.

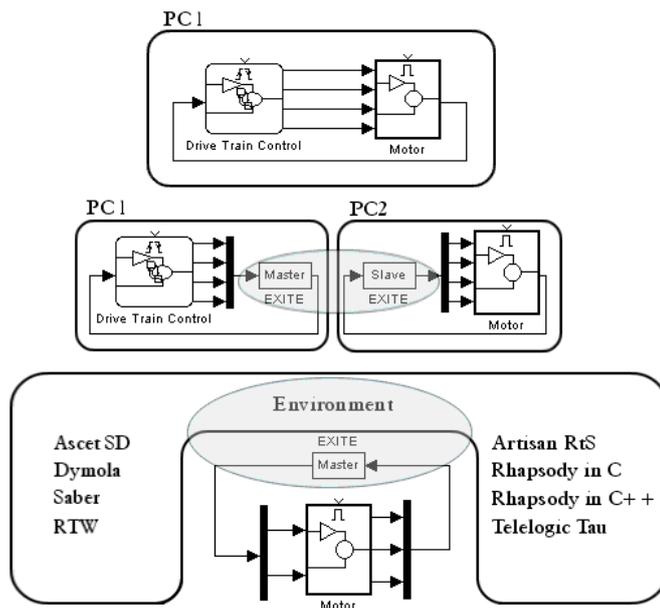


Abbildung 4: Single-Threaded zur verteilten zur hybriden Simulation

Die verteilte Simulation hilft hier, an den vordefinierten Schnittstellen gemeinsam Simulationen zu realisieren, ohne eine Gefährdung der IP herbeizuführen. Eine solche Zusammenführung findet in der Regel erst mit der Generierung und Integration von Quellcode statt. Die verteilte Simulation ermöglicht hier wesentlich früher eine Zusammenarbeit, bspw. indem OEMs virtuelle Prototypen der Lieferanten testen und bewerten können [BSW04].

## 4.5 Hybride Simulation

Bei der Entwicklung werden in der Regel verschiedenartigste Modelle und Werkzeuge eingesetzt, deren exakte Simulation unterschiedliche Berechnungsverfahren bedingen. Der Ansatz der Integration solcher Modelle in einem einzigen Code bzw. der der Einbindung externer Modelle in Form von Codefragmenten in ein Mastermodell führt unter Umständen zu einer Verfälschung der Rechenergebnisse bzw. zu suboptimalen Berechnungsverfahren, wie Praxistests anhand von Simulink und Dymola beweisen [RRS03].

Im Vergleich dazu werden durch die verteilte Simulation nur Simulationsdaten ausgetauscht; die Berechnungen der einzelnen Modelle bleiben in ihrer ursprünglichen Umgebung erhalten. Dies sichert eine optimale Nutzung der gebotenen Performanz.

## 4.6 Co-Simulation zu Code-Integration

In den vorangegangenen Abschnitten wurde auf die verteilte Simulation als Werkzeug für frühe Phasen eines Entwicklungsprozesses dargestellt. In der Simulation erzielte Ergebnisse müssen Aussagen über das Verhalten der modellierten Objekte zulassen. Im Automotive gilt dies insbesondere auch für die Simulation von Funktionssoftware der Steuergeräte. Dieses ist gegeben, wenn die verwendeten Modelle ohne weitere Änderung in die

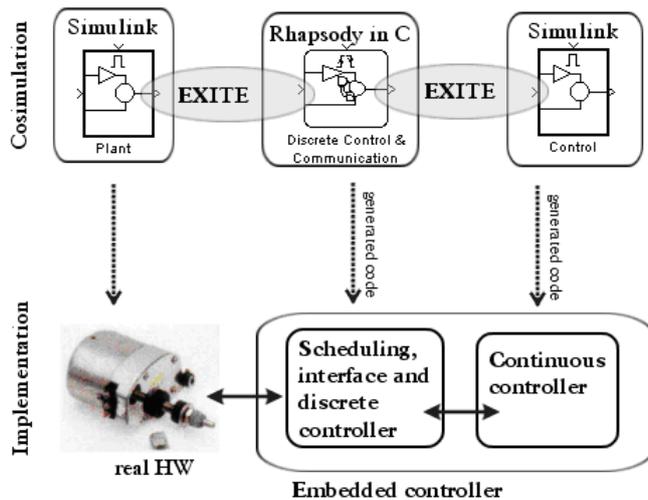


Abbildung 5: Co-Simulation zu Code-Integration

Codeentwicklung einfließen und die Ausführungsumgebung sowohl in der Simulation als auch auf dem Target (-steuergerät) aus Sicht der Funktionssoftware gleich sind.

Die erste Bedingung ist von der Art der verwendeten Zielhardware, der Modelle und der Codegeneratoren abhängig, was hier nicht weiter vertieft werden soll. Für die zweite muss die in der Co-Simulation durch **EXITE** realisierte Integration der Einzelmodelle auf dem Target auf Code-Ebene realisiert werden: a) durch eine direkte Codeintegration oder b) durch ein multithreaded Betriebssystem, dass die Kopplung der einzelnen Softwarethreads realisiert.

Mit **EXITE** wurde eine Entwicklungsumgebung für Simulink und Rhapsody<sup>5</sup> in C aufgebaut, welche auf dem IDF (Interrupt driven Framework) von Rhapsody basiert. Die **EXITE**-Plug-Ins für Rhapsody und Simulink sind so gestaltet, dass sie eine Codegenerierung erlauben, wurden aber hinsichtlich des Zieltargets targetspezifisch angepasst. Somit können nun die Modelle in der Co-Simulation mittels **EXITE** untersucht und dann ohne weitere Änderung dann auf dem Target im IDF verwendet werden, wobei der Simulink-Code in den Rhapsody-Code integriert wird.

## 5 Zusammenfassung und Ausblick

Die verteilte Simulation ist eine Antwort auf die Anforderung, Simulationen zur Optimierung von Kosten, Risiken und Qualität in weiter steigendem Umfang einzusetzen. Durch die Schaffung universeller Integrationsplattformen wird dieses Ziel der aktuellen, heterogenen Werkzeuglandschaft im Automotive bereits heute realisierbar. Durch Zusammenspiel mit zukünftige Integrationsplattformen auf Target-Ebene, wie sie bspw. im AUTOSAR-Konsortium erarbeitet werden, entsteht so ein lückenloser, modellbasierter Entwicklungsprozess.

<sup>5</sup>® I-Logix, Inc.

## 6 Literatur

### Literatur

- [BS02] G. Bikker and M. Schroeder. *Methodische Anforderungsanalyse und automatisierter Entwurf sicherheitsrelevanter Eisenbahnleitsysteme mit kooperierenden Werkzeugen*, volume 529 of *Fortschrittsberichte VDI, Reihe 12*. VDI Verlag, Düsseldorf, Germany, 2002.
- [BSW04] J. Bortolazzi, S. Steinhauser, and T. Weber. Entwicklung der kfz-elektronik - schwerpunkt software. In Gerhard Walliser, editor, *Elektronik im Kraftfahrzeugwesen: Steuerungs-, Regelungs- und Kommunikationssysteme*, pages 398–413. Expert Verlag, Renningen, 2004.
- [CW04] James O. Calvin and Richard Weatherly. An introduction to the high level architecture (hla) runtime infrastructure (rti). <http://dss.ll.mit.edu/dss.web/96.14.103.RTI.Introduction.html>, 2004.
- [Kie96] Kiefer. *Methodische Partitionierung und Parametrierung von Feldbussen*, volume 428 of *Fortschrittsberichte VDI, Reihe 10*. VDI Verlag, Düsseldorf, Germany, 1996.
- [RRS03] Felix Richert, Joachim Rückert, and Axel Schloßer. Vergleich von modelica und matlab anhand der modellbildung eines dieselmotors. *at - Automatisierungstechnik*, 51(6):247–354, 2003.
- [sys04] Systems modeling language: Sysml. <http://www.sysml.org>, 2004.





# Modellbasierte Auslegung von Fahrfunktionen

Dr.-Ing. Jan Peter Blath  
IAV GmbH, Gifhorn



## Modellbasierte Auslegung von Fahrfunktionen

### Gliederung

- 1 Einleitung**
- 2 Modellbildung Motor und Antriebsstrang**
- 3 Auslegung von Fahrfunktionen**
- 4 Zusammenfassung und Ausblick**



## 1 Einleitung

*Beispiele für Fahrfunktionen sind:*

- Ruckeldämpfer
- Fahrgeschwindigkeitsregler
- Beschleunigungsregler
- Leerlaufregler
- Anfahrregler

*Anforderungen betreffen:*

- Komfort
- Wirkungsgrad
- Robustheit
- Stöerausgleich



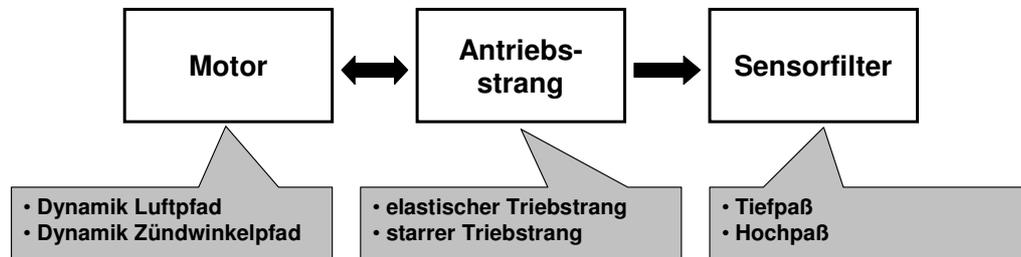
## Gliederung

- 1 Einleitung
- 2 **Modellbildung Motor und Antriebsstrang**
- 3 Auslegung von Fahrfunktionen
- 4 Zusammenfassung und Ausblick



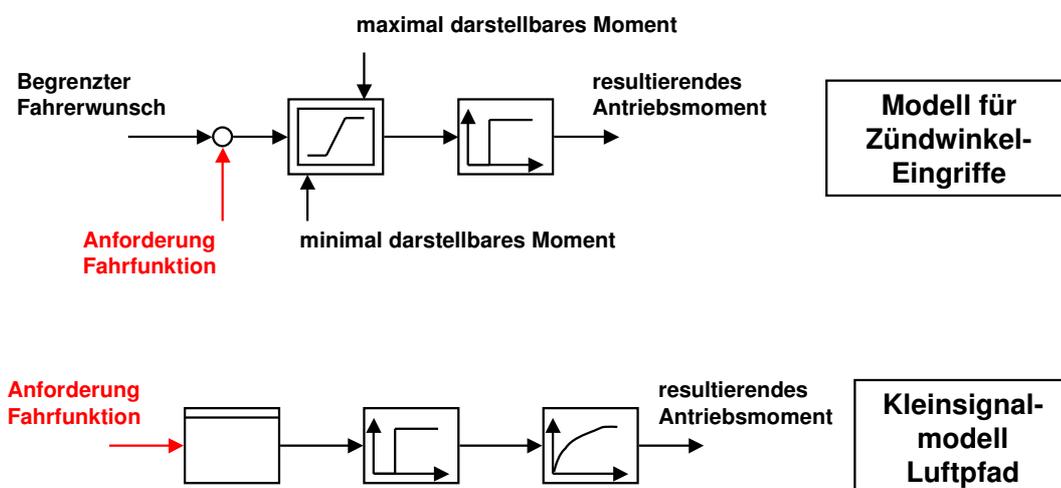
# Modellbasierte Auslegung von Fahrfunktionen

## 2 Modellbildung – Übersicht Gesamtsystem



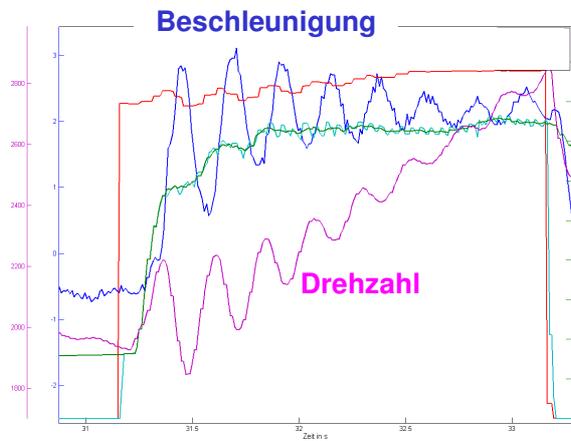
# Modellbasierte Auslegung von Fahrfunktionen

## 2 Modellbildung – Motormodelle





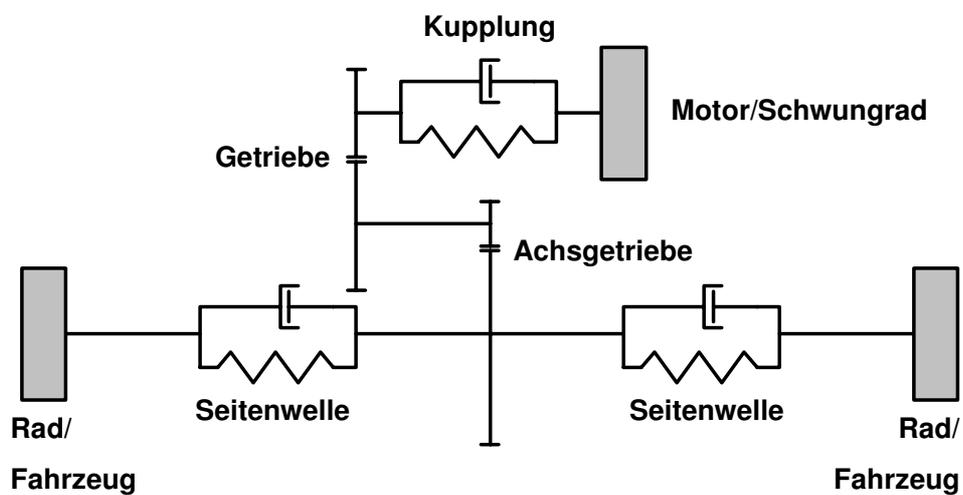
## 2 Modellbildung – Elastischer Antriebsstrang



Ruckelschwingungen beim positiven Lastwechsel



## 2 Modellbildung – Elastischer Antriebsstrang





## 2 Modellbildung – Bewegungsgleichungen elast. Antriebsstrang

Schwungrad:

$$J_1 \ddot{\varphi}_1 = M_M - M_{Verl} - c_K (\varphi_1 - i_G i_A \varphi_2) - d_K (\dot{\varphi}_1 - i_G i_A \dot{\varphi}_2)$$

Ausgang Achsgetriebe:

$$(i_G^2 i_A^2 J_2 + i_A^2 J_3 + J_4) \ddot{\varphi}_4 = i_G i_A c_K (\varphi_1 - i_G i_A \varphi_2) + i_G i_A d_K (\dot{\varphi}_1 - i_G i_A \dot{\varphi}_2) - c_S (\varphi_4 - \varphi_5) - d_S (\dot{\varphi}_4 - \dot{\varphi}_5)$$

Rad/Fahrzeug:

$$J_5 \ddot{\varphi}_5 = c_S (\varphi_4 - \varphi_5) + d_S (\dot{\varphi}_4 - \dot{\varphi}_5) - r F_{Wid}$$



## 2 Modellbildung – Vereinfachtes Modell elast. Antriebsstrang

Vereinfachung und Linearisierung führt zu

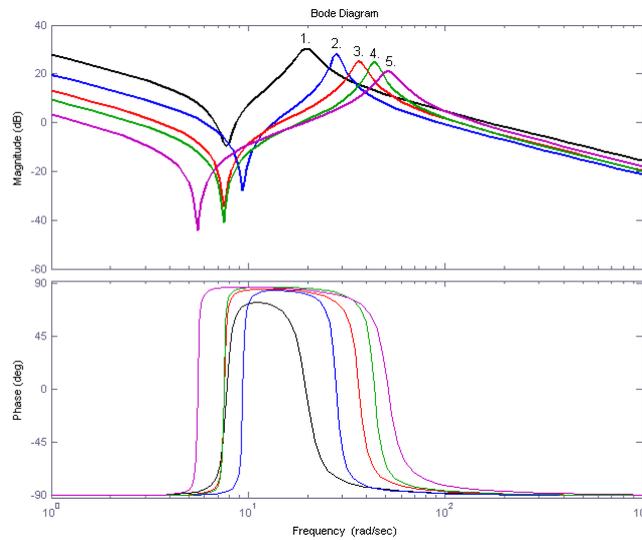
$$N_{Motor}(s) = V \cdot \frac{m}{s} \cdot \frac{b_2 s^2 + b_1 s + 1}{\frac{1}{\omega_R^2} s^2 + b_1 s + 1} \cdot (M_M(s) - M_{Verl}(s))$$

Skalierung      Starre Kopplung      Einfluß Elastizität

Beschreibung des positiven Lastwechsels durch nur 4 Parameter → zuverlässige Identifikation mit „Bordmitteln“



## 2 Modellbildung – Frequenzgänge elast. Antriebsstrang



Bode-Diagramme identifizierter Modelle



## 2 Modellbildung – Bewegungsgleichung starrer Antriebsstrang

Bewegungsgleichung:

$$J_{ges}(i_G, i_A) \dot{v} = r i_G i_A (M_M - M_{Verl}) - r^2 \left[ \frac{1}{2} c_w A_{Fzg} \rho_L v^2 + m_{Fzg} (c_{r1} + c_{r2} v) + m_{Fzg} g \sin \chi \right]$$

Luftwiderstand

Rollwiderstand

Steigungswiderstand

Linearisierung ergibt:

$$\delta V(s) = \frac{b_0}{s + a_0} \delta U(s) - \frac{d_0}{s + a_0} \delta \chi(s)$$

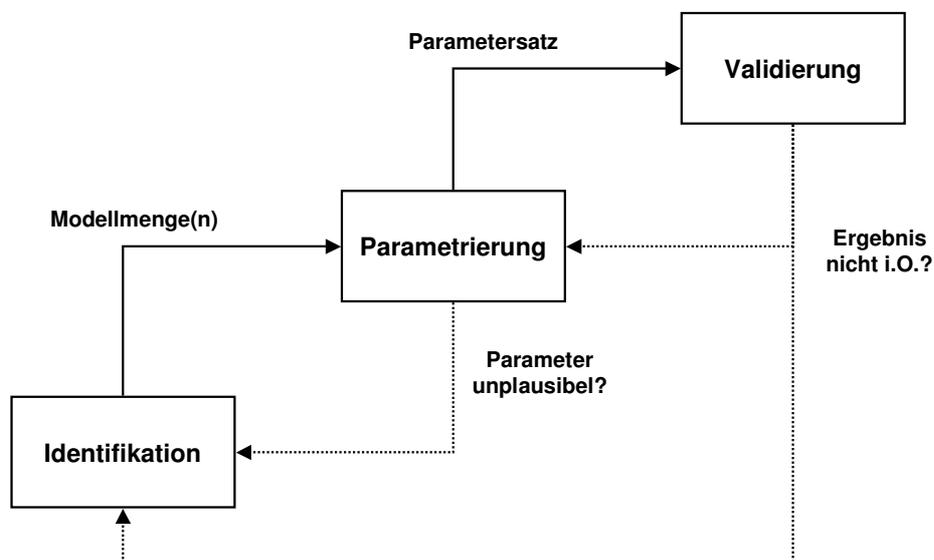


## Gliederung

- 1 Einleitung
- 2 Modellbildung Motor und Antriebsstrang
- 3 Auslegung von Fahrfunktionen**
- 4 Zusammenfassung und Ausblick

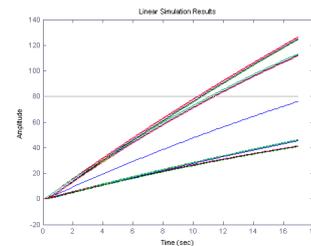
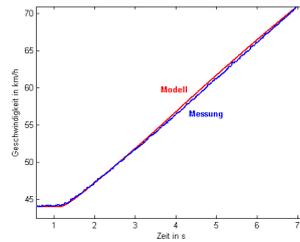
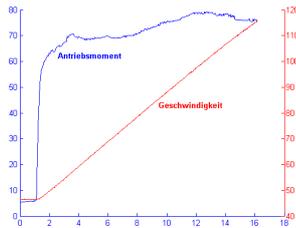
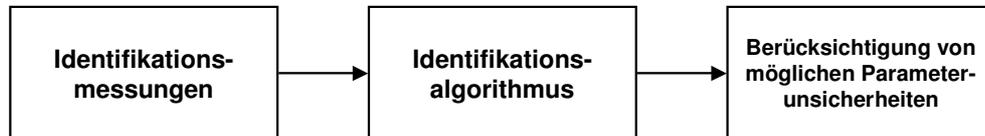


## 3 Auslegung von Fahrfunktionen – Vorgehensweise

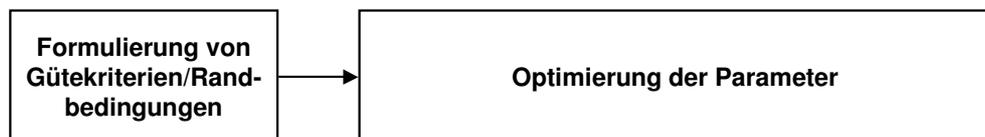




## 3 Auslegung von Fahrfunktionen – Identifikation



## 3 Auslegung von Fahrfunktionen – Parametrierung



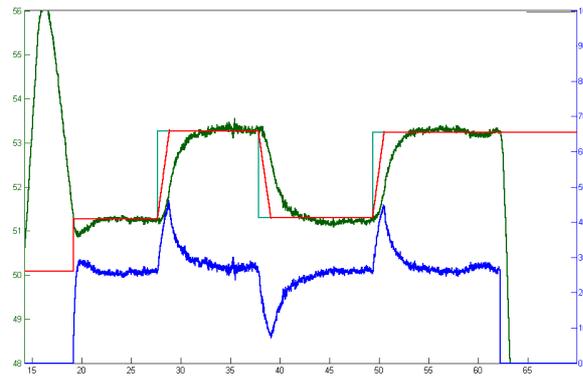
- Stabilität
- Überschwingen
- max. Stellamplitude
- Wirkungsgrad
- Dämpfung
- ...

- Optimierung mit Nebenbedingungen (z.B. Methode der Penalty-Funktion)
- Optimierung vektorieller Gütekriterien

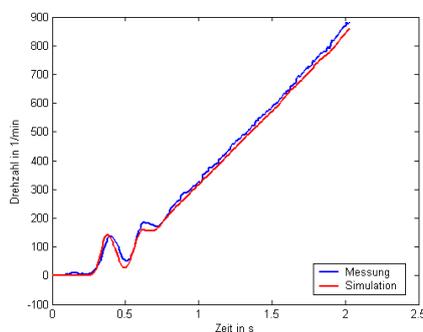


## 3 Auslegung von Fahrfunktionen – Validierung

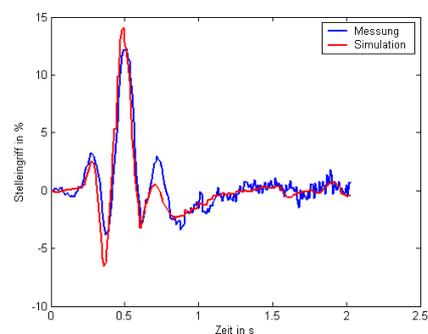
Abfahren eines Fahrprofils mit zu bewertenden Aktionen



## 3 Auslegung von Fahrfunktionen – Validierung



Motordrehzahl

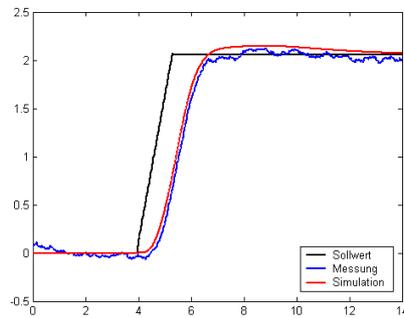


Stelleingriff

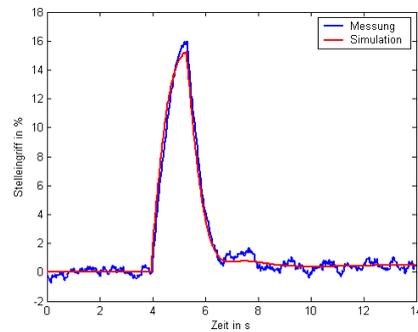
Vergleich Messung/Simulation am Beispiel Ruckeldämpfer



## 3 Auslegung von Fahrfunktionen – Validierung



Fahrzeuggeschwindigkeit



Stelleingriff

### Vergleich Messung/Simulation am Beispiel Fahrgeschwindigkeitsregler



## Gliederung

- 1 Einleitung
- 2 Modellbildung Motor und Antriebsstrang
- 3 Auslegung von Fahrfunktionen
- 4 **Zusammenfassung und Ausblick**



### 4 Zusammenfassung und Ausblick

1. Zweck der betrachteten Fahrfunktion legt problemangepaßte Modellierung nahe (Ruckeldämpfer: elastisches Modell; Fahrgeschwindigkeitsregler: starres Modell)
2. Identifizierbarkeit mit begrenztem Sensoraufwand ist zu beachten (Anzahl und Position Drehzahlsensoren, Gültigkeit des berechneten Moments)
3. Optimierungsansätze für die Auslegung oft zielführend, da Gütekriterien und Randbedingungen systematisch Berücksichtigung finden
4. Nutzung des durch Modellierung erworbenen Prozeßwissens zur Entwicklung neuer Steuergerätefunktionen möglich



# Potenzial zur Verbesserung der Klopfregelung in Ottomotoren

Ulf Lezius, Wolfgang Drewelow, Matthias Schultalbers

## 1 Einleitung

Mit dem Begriff Klopfen wird ein Verbrennungsphänomen in Ottomotoren bezeichnet, bei dem sich Teile des Benzin-Luft-Gemisches von selbst entzünden, ohne von der regulären Flammenfront erfasst worden zu sein. Der schlagartige Stoffumsatz bei diesem Selbstzündungsvorgang führt zur Anregung von Resonanzschwingungen im Brennraum. Diese Schwingungen überlagern sich dem normalen Druckverlauf im Zylinder und sind das typische Merkmal einer klopfenden Verbrennung.

Die Ursachen für dieses Verhalten sind komplex. Wesentliche Einflussfaktoren sind hohe Temperaturen und Drücke. Diese wiederum werden durch eine Vielzahl von Randbedingungen, konstruktiven Parametern und Stellgrößen festgelegt (Abbildung 1).

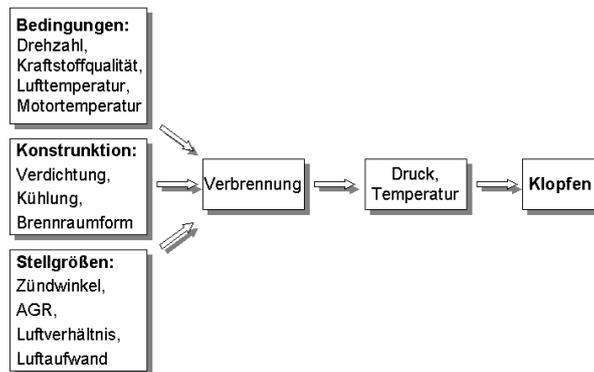


Abbildung 1: Prinzipielle Kausalkette bei der Entstehung von klopfender Verbrennung

Klopfende Verbrennungen schädigen den Motor mechanisch und müssen daher verhindert werden. Zu diesem Zweck sind moderne Ottomotoren mit Klopfregelungen ausgestattet.

Im vorliegenden Beitrag wird kurz die typische Funktionsweise von Klopfregelungen erläutert. Darauf aufbauend werden Ansätze für verbesserte Verfahren zur Klopfregelung betrachtet. Mit Hilfe von Messdaten wird das Potenzial zur Verbesserung der Klopfregelung gezeigt.

## 2 Grundsätzliche Funktionsweise von Klopfregelungen

Klopfregelungen erfüllen im Wesentlichen zwei Aufgaben:

- Erkennung und Verhinderung von unkontrolliert auftretendem Klopfen
- Betrieb des Motors nahe der Klopfgrenze zur Leistungsmaximierung

Die Klopfregelung basiert üblicherweise auf der Erkennung klopfender Arbeitsspiele. Für diese Klopfenerkennung werden z.B. Messungen des Zylinderdrucks, des Körperschalls oder Ionenstromsignale verwendet ([2]). Die entsprechenden Messsignale werden über Hochpass- oder Bandpassfilter gefiltert, wobei die Filterfrequenzen auf die typischen Resonanzfrequenzen des Brennraums

abgestimmt sind. Aus den gefilterten Signalen werden Kenngrößen zur Einschätzung der Klopfstärke abgeleitet, z.B.

$$I = \max(|s(i)|) \quad \text{Betragsmaximum} \quad (1)$$

$$I = \frac{1}{k} \cdot \sum_{i=0}^{k-1} |s(i)| \quad \text{Betragsmittel} \quad (2)$$

$$I = \frac{1}{k} \cdot \sum_{i=0}^{k-1} s(i)^2 \quad \text{quadratisches Mittel} \quad (3)$$

Die Kenngrößen werden in einem für das Auftreten von Klopfen relevanten Zeitfenster berechnet. Zur Detektion von klopfenden Verbrennungen und gleichzeitigen Bildung eines Klopfmaßes wird die gebildete Kenngröße mit einer Schwelle verglichen (4). Bei der Regelung werden lediglich Anteile der Klopfintensität verwendet, die diese Schwelle überschreiten.

$$\Delta I = \begin{cases} I - I_0 & , \quad I \geq I_0 \\ 0 & , \quad \text{sonst} \end{cases} \quad (4)$$

Als Stellgröße der Klopfregelung dient der Zündwinkel  $\alpha_{ign}$ . Eine Spätverstellung des Zündwinkels reduziert die Klopfneigung, führt aber gleichzeitig zur Verringerung des erzeugten Drehmoments und des Wirkungsgrades. Die Gleichung

$$\alpha_{ign}(n) = \alpha_{ign}(n-1) + \Delta\alpha_{adv} - \Delta\alpha_{ret} \cdot \Delta I(n) \quad (5)$$

beschreibt einen typischen Algorithmus zur Klopfregelung, wie er in aktuellen Motorsteuerungen Verwendung findet. Der aktuelle Zündwinkel  $\alpha_{ign}(n)$  ergibt sich direkt aus dem Zündwinkel des vorhergehenden Arbeitsspiels  $\alpha_{ign}(n-1)$ . In jedem Regelungsschritt wird der Zündwinkel um einen festgelegten Betrag  $\Delta\alpha_{adv}$  nach früh in Richtung der Klopfgrenze verschoben. Nimmt die Klopfstärke  $\Delta I$  einen Wert größer als Null an, wird der Zündwinkel zur Reduzierung der Klopfneigung um den Wert  $\Delta\alpha_{ret} \cdot \Delta I(n)$  nach spät verschoben.

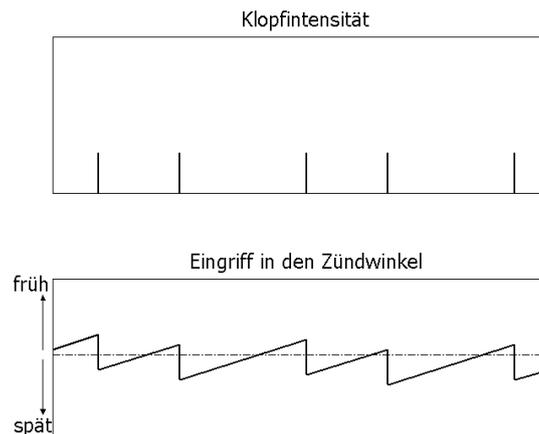


Abbildung 2: Verlauf der Stellgröße einer Klopfregelung

Mit diesem Prinzip der Grenzwertregelung ergibt sich der in Abbildung 2 dargestellte typische Verlauf der Stellgröße. Die Struktur der Regelung ist in Abb. 3 veranschaulicht.

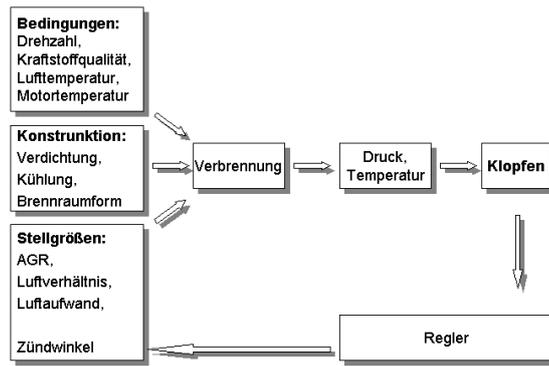


Abbildung 3: Typische Klopfregelung

### 3 Idee für eine Klopfregelung in zylinderdruckgeführten Motorsteuerungen

Die oben erläuterten Prinzipien der Klopfregelung beruhen im Wesentlichen auf der Auswertung der Klopfintensität. Nachteilig dabei ist, dass unterhalb der Klopfschwelle keine Information über den Abstand zur Klopfgrenze benutzt wird. Der Regler muss sich deshalb wieder „blind“ an die Klopfgrenze herantasten. Erfolgsversprechender ist ein Ansatz, der in der Regelung ein Abstandsmaß zur Klopfgrenze verwendet. Das hier vorgeschlagene Konzept geht davon aus, dass zur Ermittlung eines solchen Abstands der Zylinderdruckverlauf herangezogen werden kann (Abbildung 4). Dabei sind unterschiedliche Merkmale des Zylinderdrucks denkbar, die als Maß für den Abstand zur Klopfgrenze benutzt werden können. Beispiele für solche Maße sind:

- Bewertung der Form des Druckverlaufs (z.B. Maximum des Zylinderdruckverlaufs, maximaler Anstieg des Zylinderdruckverlaufs)
- reaktinskinetische Maße (z.B. [1])

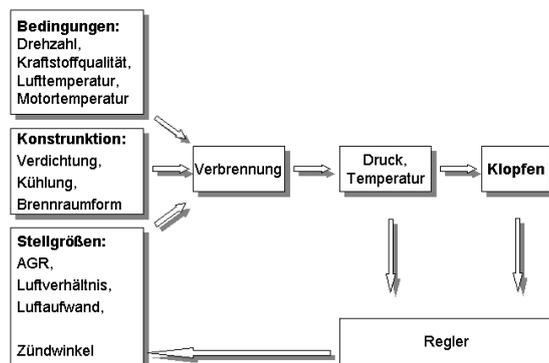


Abbildung 4: Konzept für eine erweiterte Klopfregelung

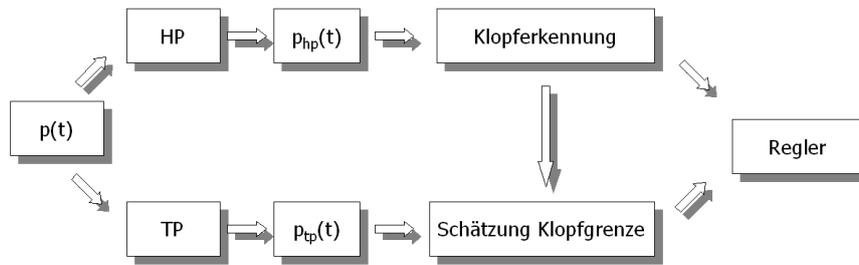


Abbildung 5: Struktur der Klopfregelung unter Verwendung von Zylinderdruckmessungen

In Abbildung 5 ist eine denkbare Struktur zur Realisierung des neuen Regelungsansatzes dargestellt. Das gemessene Zylinderdrucksignal wird hier auf zwei getrennten Pfaden verarbeitet. Ein hochpassgefiltertes Zylinderdrucksignal wird für die Erkennung klopfender Arbeitsspiele genutzt. Das über ein Tiefpass gefilterte Signal enthält zwar keine auswertbaren Klopf-schwingungen, kann aber mit statistischen, thermodynamischen oder reaktionskinetischen Methoden ausgewertet werden, um den Abstand der analysierten Arbeitsspiele zur Klopfgrenze zu ermitteln. Beide Informationen (aktuelle Klopfstärke, Abstand zur Klopfgrenze) werden dem Regler zugeführt. Dieser Regler nimmt den Stelleingriff in den Zündzeitpunkt vor [3].

Im Rahmen von Voruntersuchungen wurden Messungen an einem mit Zylinderdrucksensoren ausgestatteten Motor VW 1,6l 85kW FSI vorgenommen. Insbesondere wurde untersucht, ob Kenngrößen wie das Maximum des Zylinderdruckverlaufs und ein reaktionskinetisches Maß nach Franzke als Abstand zur Klopfgrenze benutzt werden können.

Es wurden in einem konstanten Arbeitspunkt des Motors (1500 U/min, Vollast) verschiedene Zündwinkel eingestellt. Aus den gemessenen Zylinderdrücken wurde für jedes Arbeitsspiel des Motors die Klopfstärke und der maximale Zylinderdruck berechnet.

Die Abbildung 6 zeigt, wie sich die Häufigkeitsverteilung der Klopfstärke in Abhängigkeit vom Zündwinkel verhält. Dabei wurde die Klopfintensität nach Formel 1 als das Betragsmaximum des hochpassgefilterten Zylinderdrucksignals berechnet. Dargestellt sind die Häufigkeitsverteilungen für drei verschiedene Zündwinkel. Die Zündwinkel von  $0^\circ$  und  $2,25^\circ$  vor dem oberen Totpunkt können in diesem Fall als nicht klopfende Betriebspunkte angesehen werden. Ein eingestellter Zündwinkel von  $5,25^\circ$  ruft klopfende Arbeitsspiele hervor. Es wird deutlich, dass zwischen zwei Zündwinkeln, die kein Klopfen verursachen ( $0,00^\circ$  und  $2,25^\circ$ ), kein wesentlicher Unterschied in den Häufigkeitsverteilungen der Klopfintensitäten besteht. Erst bei tatsächlichem Klopfen (Zündwinkel  $5,25^\circ$ ) wird ein deutlicher Unterschied zu den nicht klopfenden Zündwinkeln sichtbar.

Die Abbildung 7 zeigt das Verhalten des Zylinderdruckmaximums bei der Variation des Zündwinkels. Es liegen die selben Messdaten zu Grunde wie in der Abbildung 6. Im Gegensatz zur Klopfintensität zeigt diese Darstellung eine deutliche Verschiebung der Lage der Häufigkeitsverteilung bei einer Veränderung des Zündwinkels um die Klopfgrenze herum.

Neben einfachen Merkmalen des Zylinderdrucks wie dem oben gezeigten Zylinderdruckmaximum, können auch reaktionskinetische Kriterien Informationen über den Abstand zur Klopfgrenze liefern. In [1] wird ein solches Kriterium beschrieben:

$$N(\alpha) = \frac{1}{\omega} \cdot \int_{\alpha_{es}}^{\alpha} p^a \cdot e^{-\frac{b}{T_{uv}}} d\alpha \quad (6)$$

Dabei wird mit Hilfe der Temperatur der unverbrannten Gase  $T_{uv}$  und des Zylinderdrucks  $p$  ein abstraktes reaktionskinetisches Niveau  $N$  berechnet. Die Variablen  $a$  und  $b$  bezeichnen zwei

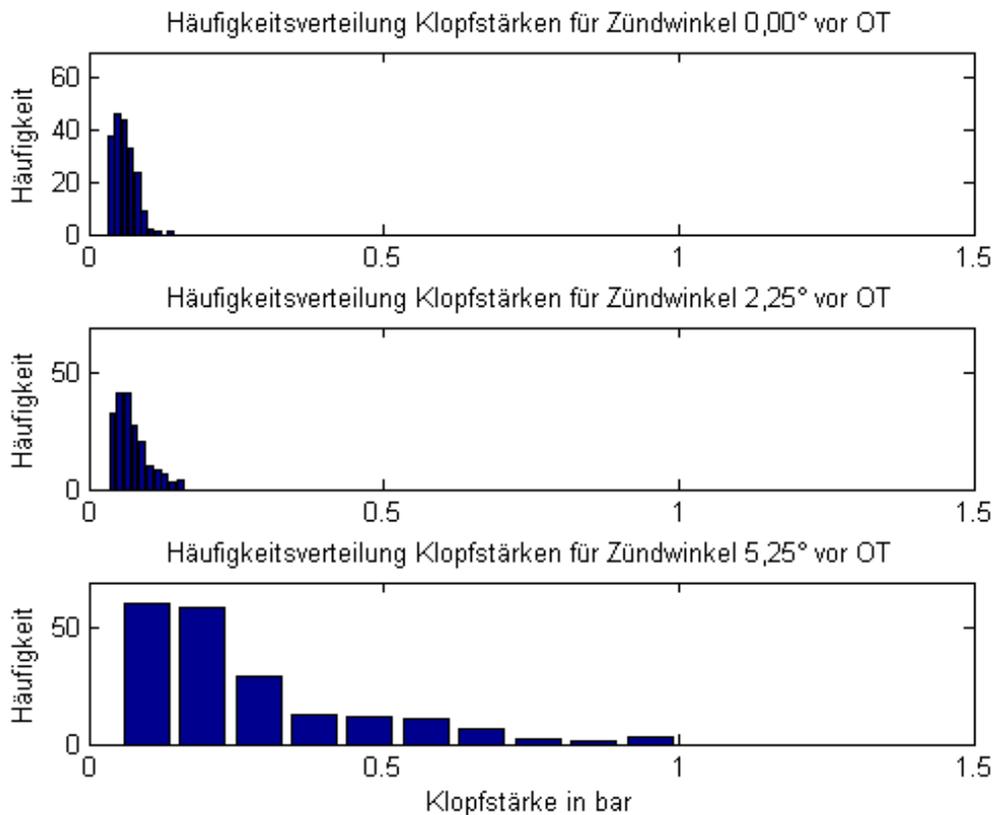


Abbildung 6: Häufigkeitsverteilungen der Klopfstärke bei verschiedenen Zündwinkeln

verschiedene, günstig zu wählende Parameter,  $\omega$  ist die Rotationsgeschwindigkeit der Kurbelwelle. Die Integration wird vom Kurbelwinkel, bei dem die Einlassventile schließen  $\alpha_{es}$ , bis zum aktuellen Kurbelwinkel  $\alpha$  ausgeführt

Wird ein bestimmtes kritisches Reaktionsniveau vor einem kritischen Zeitpunkt bzw. Kurbelwinkel erreicht, so kann die Schlussfolgerung gezogen werden, dass die Voraussetzungen für eine Selbstzündung des Kraftstoff-Luft-Gemisches im Zylinder gegeben sind.

Es ist also offensichtlich möglich, vom Zylinderdruckverlauf auf den Abstand zur Klopfgrenze für jedes einzelne Arbeitsspiel zu schließen. Diese Information ermöglicht eine Weiterentwicklung bestehender Klopfregelungen. Ist der Abstand zur Klopfgrenze bekannt, kann der Zündwinkel schnell und gezielt verstellt werden, selbst wenn kein Klopfen auftritt.

#### 4 Abschätzung des Verbesserungspotenzials

Mit Hilfe von Messdaten soll das Verbesserungspotenzial abgeschätzt werden. Dabei soll der Betrieb eines Ottomotors XXXXXXXXXX bei eingeschalteter Klopfregelung analysiert werden.

Die Abbildung 8 zeigt die aufgetretenen maximalen Zylinderdrücke und die dazugehörigen Klopfintensitäten. Zu erkennen ist, dass Klopfen erst beim Überschreiten eines bestimmten Wertes des maximalen Zylinderdrucks auftritt. Mit wachsendem Zylinderdruckmaximum steigt die Wahrscheinlichkeit und auch die Schwere auftretender Klopfereignisse. Es ist also möglich, mit Hilfe des Zylinderdruckmaximums einen Schwellwert für das Auftreten vom Klopfen zu definieren. Der Vorteil ist darin zu sehen, dass das Zylinderdruckmaximum auch unterhalb der Klopfschwelle aussagekräftige Messwerte liefern kann. Allerdings ist diese Klopfgrenze nicht konstant. Sie hängt von den Betriebsbedingungen und den aktuellen Werten der Stellgrößen des Motors ab.

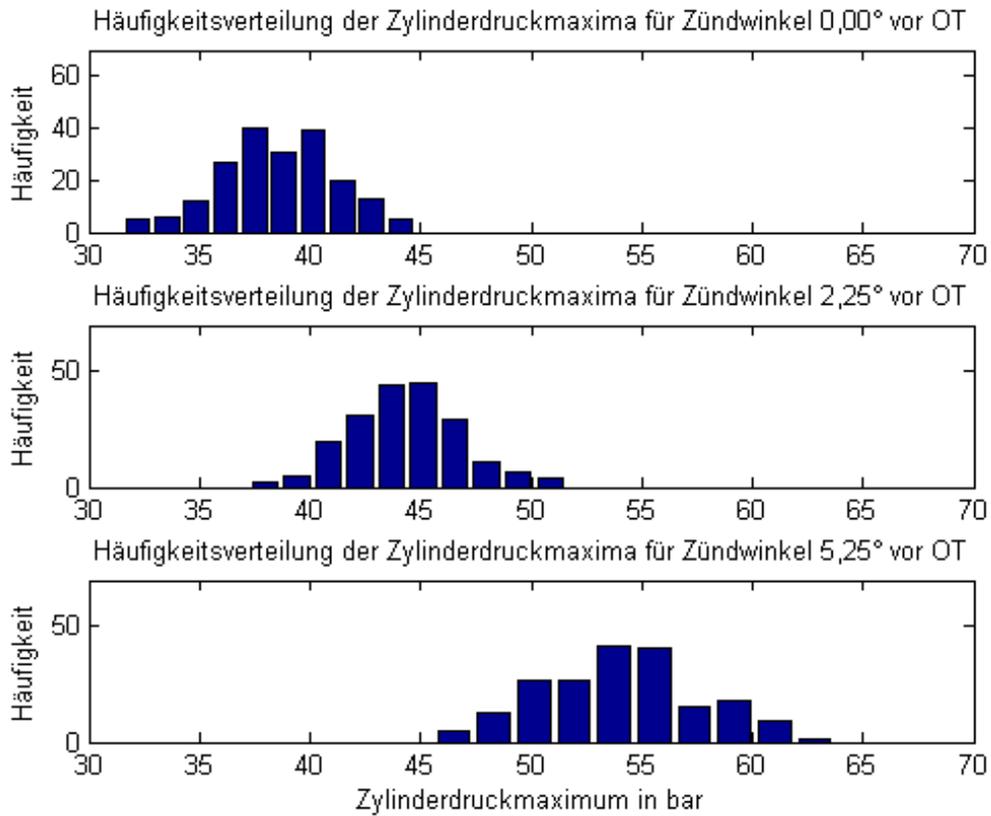


Abbildung 7: Häufigkeitsverteilungen der Zylinderdruckmaxima bei verschiedenen Zündwinkeln

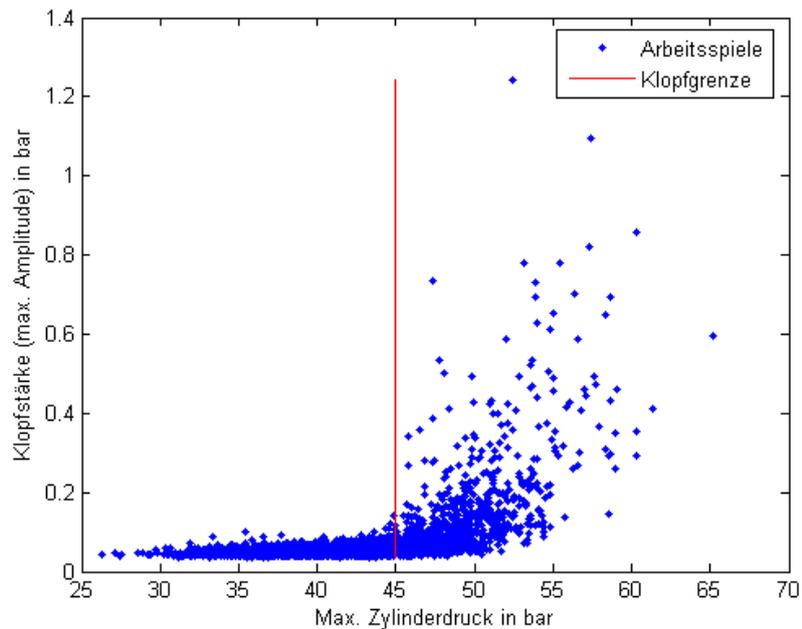


Abbildung 8: Festlegung einer Klopfgrenze für den Zylinderdruck

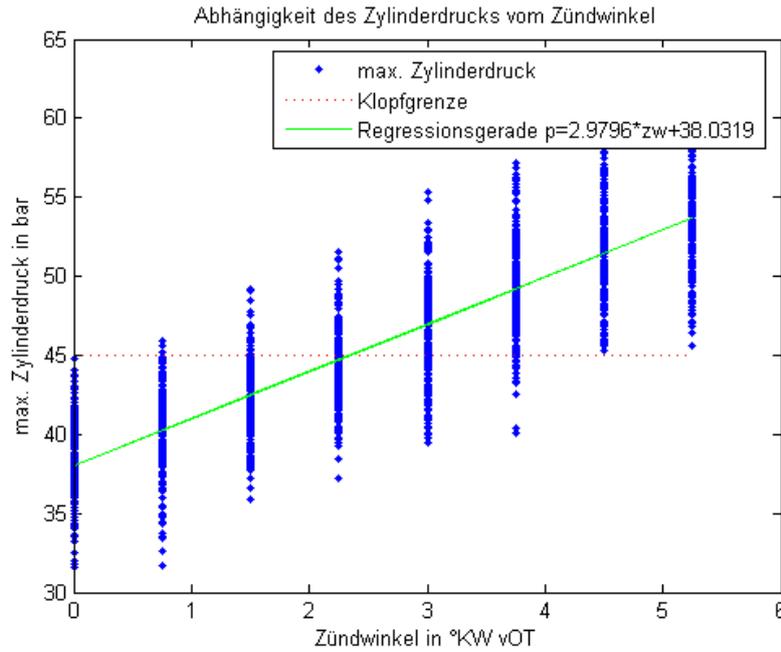


Abbildung 9: Abhängigkeit des Zylinderdruckmaximums vom Zündwinkel

In der Abbildung 9 ist die Abhängigkeit des maximalen Zylinderdrucks vom Zündwinkel dargestellt. Dabei wurden mehrere Zündwinkel eingestellt und konstant belassen. Die sich ergebenden Zylinderdruckmaxima wurden über mehrere Arbeitsspiele aufgezeichnet. Eine Frühverstellung des Zündwinkels führt im Mittel zu höheren Zylinderdrücken. In der Nähe der Klopfgrenze wurde ein weitgehend linearer Zusammenhang zwischen Zündwinkel und Zylinderdruckmaximum beobachtet. Dieser Zusammenhang kann durch eine Regressionsgerade  $p_{max} = m \cdot \alpha_{ign} + n$  angenähert werden.

In einem konstant gehaltenen Arbeitspunkt ist es also möglich, für jedes einzelne Arbeitsspiel den Abstand zur Klopfgrenze zu schätzen. Die Differenz zwischen dem Zylinderdruckmaximum  $p_{max}$  und dem Schwellwert  $p_{KG}$  wird als  $\Delta p_{max}$  bezeichnet.

$$\Delta p_{max} = p_{max} - p_{KG} \quad ; \quad p_{KG} = f(n_{mot}, OZ, T_{Ansaug}, T_{mot}, \lambda, r_l, AGR) \quad (7)$$

Mit Hilfe der Steigung der Regressionsgeraden kann diese Druckdifferenz in eine Zündwinkel-differenz  $\Delta \alpha_{ign}$  umgerechnet werden.

$$\Delta \alpha_{ign} = \frac{1}{m} \cdot \Delta p_{max} \quad (8)$$

Die Abbildung 10 zeigt die Zeitverläufe verschiedener aus dem Zylinderdruck berechneter Größen. Während der Messung lief der verwendete Motor bei 1800 U/min und Vollgas. Die in diesem Fall aktive serienmäßige Klopfregelung des Motors basiert auf dem Körperschallsignal und einem Regelalgorithmus ähnlich zu dem in Abschnitt 2 beschriebenen. Das oberste Diagramm in Abbildung 10 zeigt den Verlauf der aus dem Zylinderdruck ermittelten Klopfintensität. Es wird deutlich, dass die Annäherung an die Klopfgrenze nicht zuverlässig aus der Klopfintensität ermittelt werden kann. Einige Klopfereignisse treten ohne Vorwarnung auf. Im mittleren Diagramm wird der dazugehörige Verlauf des maximalen Zylinderdrucks dargestellt. Während die Klopfregelung den Zündwinkel nach früh verstellt, zeigt sich deutlich die

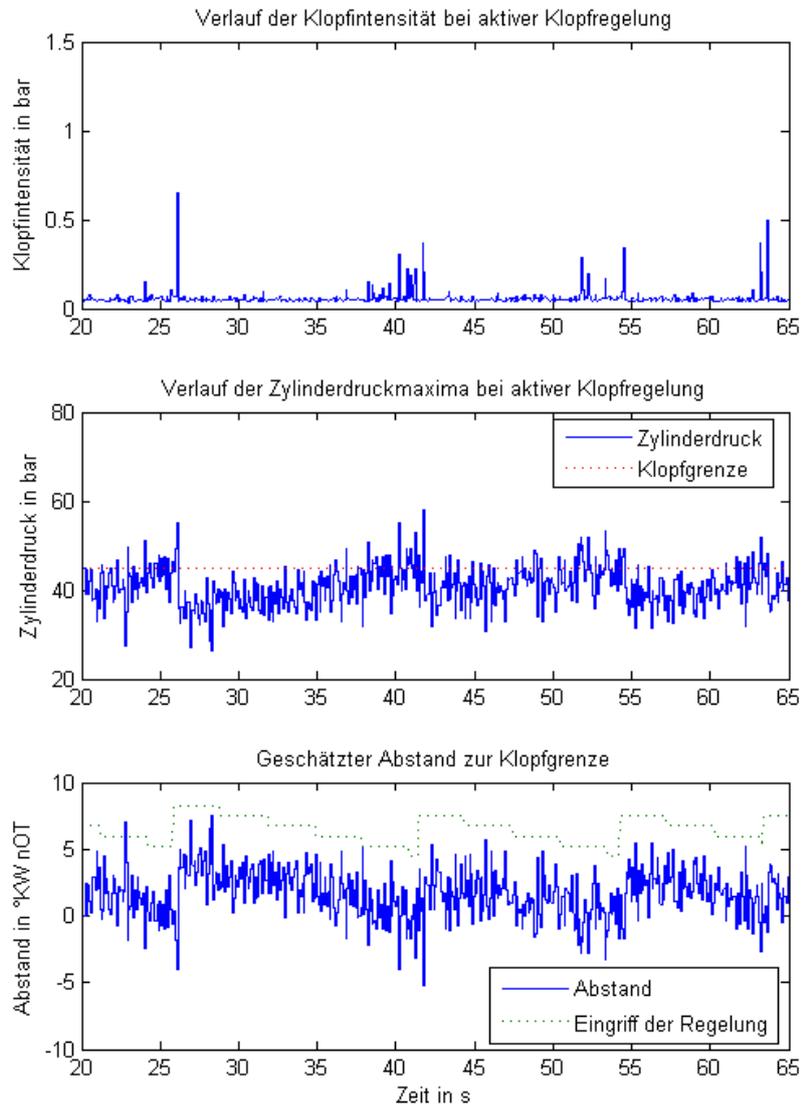


Abbildung 10: Zeitverläufe bei aktiver Klopfregelung

Annäherung an die Klopfgrenze und das Überschreiten derselben. Nach aufgetretenen Klopfereignissen verstellt die Klopfregelung den Zündwinkel schlagartig um einen größeren Betrag nach spät, was sich ebenfalls im Zylinderdruck widerspiegelt. Das untere Diagramm zeigt den nach (8) geschätzten Abstand zur Klopfgrenze ausgedrückt in  $^{\circ}$  KW nach dem oberen Totpunkt. Zum Vergleich wurde der Stelleingriff der Klopfregelung ebenfalls in diesem Diagramm dargestellt.

Es zeigte sich, dass der Motor während dieser Messung bei aktiver Klopfregelung teilweise unterhalb der Klopfgrenze betrieben wurde. Deshalb konnte das maximal mögliche Drehmoment nicht erreicht werden. Der Mittelwert aller ermittelten Abstände betrug  $0,87^{\circ}$  KW. Dieser Wert kann als ein Anhaltspunkt für das Potenzial zur Verbesserung bestehender Klopfregelalgorithmen dienen.

## 5 Zusammenfassung

In den vorliegenden Ausführungen wurde ein Ansatz entwickelt, mit Hilfe der Zylinderdruckindizierung den Abstand einzelner Arbeitsspiele zur Klopfgrenze zu schätzen. Damit wird es ermöglicht, Klopfregelungen zu entwerfen, die den Zündwinkel möglichst dicht an der Klopfgrenze zu betreiben. Solche Regelungen haben das Potenzial, Wirkungsgrad und Leistung von Ottomotoren weiter zu steigern.

## Literatur

- [1] FRANZKE, D. E.: *Beitrag zur Ermittlung eines Klopfkriteriums der ottomotorischen Verbrennung und zur Vorausberechnung der Klopfgrenze*, Technische Universität München, Dissertation, 1981
- [2] KIENCKE, Uwe ; NIELSEN, Lars: *Automotive Control Systems for Engine, Driveline and Vehicle*. 1. Auflage. Berlin : Springer Verlag, 2000
- [3] LEZIUS, Ulf ; DREWELow, Wolfgang ; SCHULTALBERS, Matthias: Vergleich verschiedener Konzepte für die Klopfregelung. In: *11. Symposium Maritime Elektronik und Elektrotechnik — Arbeitskreis Steuerungs und Regelungstechnik*. Rostock, 2004, S. 53–57
- [4] MÜLLER, N. ; ISERMANN, R.: Zylinderdruck-basiertes Motormanagement beim Ottomotor. In: *at* (2003), 8, Nr. 51, S. 352–360
- [5] ROBERT BOSCH GMBH (Hrsg.): *Ottomotor-Management*. 2. Auflage. Braunschweig/Wiesbaden : ViewegVerlag, 2003

## Verfasser

Dipl.-Ing. Ulf Lezius  
Dr.-Ing. Wolfgang Drewelow  
Universität Rostock  
Institut für Automatisierungstechnik  
Richard-Wagner-Str. 31 / H8  
18119 Warnemünde  
Tel.-Nr.: +49-(0)381-498-3548  
E-Mail: ulf.lezius@stud.uni-rostock.de

Dipl.-Ing. Matthias Schultalbers  
IAV GmbH  
Betrieb Gifhorn  
AE-MO  
Nordhoffstr. 5  
38518 Gifhorn  
Tel.-Nr.: +49-(0)5371-805-1456  
E-Mail: matthias.schultalbers@iav.de



# Modellierung und Simulation des ottomotorischen Verbrennungsprozesses mittels Ein- und Zweizonenmodell

ASIM – Workshop 2004  
Wismar

Janssen, C.

Universität Rostock, Lehrstuhl für Technische Thermodynamik

Prof. Dr.-Ing. habil E. Hassel  
Dr.-Ing. J. Nocke  
Dipl.-Ing. A. Schulze  
M.Eng. N. Chindaprasert

Dipl.-Ing. M. Schultalbers / IAV GmbH  
Dr. W. Gottschalk / IAV GmbH



Modellierung und Simulation des ottomotorischen  
Verbrennungsprozesses mittels Ein-  
und Zweizonenmodell



## Gliederung

- Einleitung und Motivation
- Modellansatz
  - Energiebilanz
  - Innere Energie
  - Wärmeübergang
  - Brennraumaufteilung
  - Temperaturbestimmung
  - Chemisches Gleichgewicht
  - $\text{NO}_x$ -Berechnung
- Ergebnisse
- Zusammenfassung und Ausblick

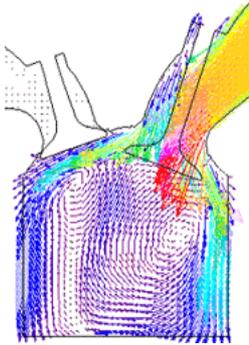


Modellierung und Simulation des ottomotorischen  
Verbrennungsprozesses mittels Ein-  
und Zweizonenmodell



## Einleitung und Motivation

Bestimmung und Optimierung folgender Größen durch Analyse des Zylinderdruckverlaufes

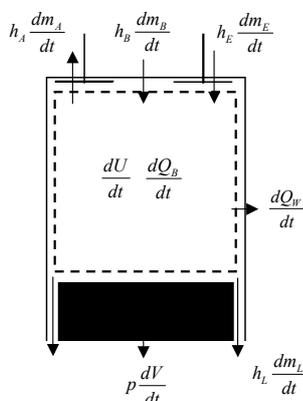


- Verbrennungsschwerpunkt
- Kraftstoffverbrauch
- Lambda
- Zündverzug
- Brennverlauf
- Abgas-/Prozesstemperaturen
- Schadstoffemissionen ( $\text{NO}_x$ , CH, CO)

## Modellansatz - Energiebilanz

Energiebilanz über den Brennraum (1. Hauptsatz)

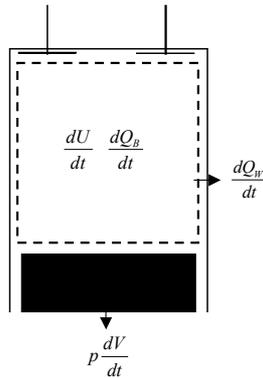
$$\frac{dU}{dt} = \frac{dQ_B}{dt} - \frac{dQ_W}{dt} - p \frac{dV}{dt} + h_E \frac{dm_E}{dt} - h_A \frac{dm_A}{dt} - h_L \frac{dm_L}{dt} + h_B \frac{dm_B}{dt}$$



Terme:

1. Chemischgebundene Energie aus Kraftstoff
2. Wärmestrom über Brennraumwand
3. Abgegebene Arbeit
4. Ein- und austretende Enthalpieströme
5. Enthalpiestrom aus Blowby-Verlusten
6. Enthalpiestrom durch Kraftstoffeinspritzung

## Modellansatz



Vereinfacht (Einzonenmodell):

$$\frac{dU}{dt} = H_U \frac{dm_B}{dt} - \frac{dQ_W}{dt} - p \frac{dV}{dt}$$

Voraussetzungen und Annahmen

- Geschlossenes System (kein Massenstrom)
- Homogene Kraftstoffverteilung
- Homogene Druckverteilung
- Homogene Temperaturverteilung
- Vollständige Verbrennung

## Bestimmung der Inneren Energie

Ansatz nach Zachariasansatz<sup>1</sup>

- Realgaseigenschaften werden berücksichtigt
- Dissoziation wird berücksichtigt
- Gaszusammensetzung bleibt unberücksichtigt (Wasserluftgehalt und AGR fehlen)
- Schnelle Berechnung, da Polynomansatz

Ideale Mischung realer Gase<sup>2</sup>

- Realgaseigenschaften werden eingeschränkt berücksichtigt
- Dissoziation wird eingeschränkt berücksichtigt
- Gaszusammensetzung wird berücksichtigt (Wasserluftgehalt und AGR)
- Langsame Berechnung (Polynome und Tabellenwerte)
- Flexibel (Einsatz auch im Zweizonenmodell möglich)
- Einfaches Chemiemodell (Bruttoreaktion)

<sup>1</sup> Dissertation F. Zacharias, TU Berlin, 1966    <sup>2</sup> VDI 4670

## Bestimmung des Wärmeüberganges

(Newton)  $\dot{Q}_W = \alpha * A * (T_W - T_B)$

$\alpha$  – Wärmeübergangskoeffizient (Ermittlung über Modell)

$A$  – Zylinderinnenfläche (aus Vorausberechnung)

$T_W$  – Wandtemperatur (fixiert)

$T_B$  – Brennraumtemperatur (aus Berechnung)

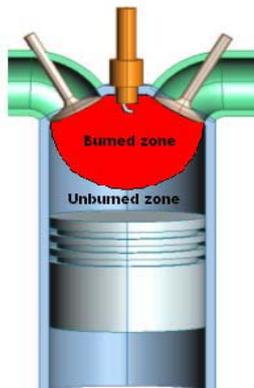
Verschiedene halbempirische Modellansätze zur Berechnung von  $\alpha$ :

Hohenberg, Woschni, Eichelberg, Nusselt

$$\alpha = 130V^{-0.06} p^{0.8} T^{0.4} (c_m + 1.4)^{0.8}$$

## Brennraumaufteilung

Bestimmung der thermischen  $\text{NO}_x$ -Emissionen nur möglich, wenn die Temperatur hinter der Flammenfront bekannt ist. Daher ist die Trennung in eine Reaktions- (rot) und Frischluftzone notwendig.



$\text{NO}_x$  Entstehung durch

- Thermisches  $\text{NO}_x$
- Promptes  $\text{NO}_x$  (geringer Anteil)
- Kraftstoff  $\text{NO}_x$  (nicht vorhanden)

Annahmen:

- Homogene Druckverteilung
- Massenverteilung aus Einzonenmodell
- Energieübertragung zwischen den Zonen nur durch turbulente Vermischung<sup>3</sup>

<sup>3</sup> Dissertation G. Heider, TU München, 1996

## Temperaturbestimmung der Reaktionszone

Temperaturdifferenz<sup>1</sup>

$$T(\varphi)_1 - T(\varphi)_2 = B(\varphi) \cdot A^*$$

Indizes

1 - Verbrannt

2 - Unverbrannt

A\* - Motorspezifischer Faktor

VB - Verbrennungsbeginn

AOE - Auslass öffnet

Charakterisierungsfunktion<sup>3</sup>

$$B(\varphi) = 1 - \frac{\int_{\varphi_{VB}}^{\varphi} [p(\varphi) - p_0(\varphi)] m_1(\varphi) d\varphi}{\int_{\varphi_{VB}}^{\varphi_{AOE}} [p(\varphi) - p_0(\varphi)] m_1(\varphi) d\varphi}$$

Temperaturdifferenz zwischen beiden Zonen zu Beginn der Verbrennung maximal und gegen Ende Null.

<sup>3</sup> Dissertation G. Heider, TU München, 1996



Modellierung und Simulation des ottomotorischen  
Verbrennungsprozesses mittels Ein-  
und Zweizonenmodell



## Temperaturbestimmung der Reaktionszone

Ideales Gas

$$p_Z = p_1 = p_2 \quad V_Z = V_1 + V_2 \quad m_Z = m_1 + m_2$$

$$p_Z V_1 = m_1 R_1 T_1 \quad p_Z V_2 = m_2 R_2 T_2$$

$$T_2 = \left( \frac{p_Z V}{m_2 R_2} - \frac{m_1 R_1}{m_2 R_2} \cdot B(\varphi) A \right) \cdot \left( 1 + \frac{m_1 R_1}{m_2 R_2} \right)^{-1}$$

Mittlere Temperatur aus Einzonenmodell

$$T_M = \frac{T_1 c p_1 m_1 + T_2 c p_2 m_2}{c p_1 m_1 + c p_2 m_2}$$

$$T_2 = \frac{T_M \cdot (c p_1 m_1 + c p_2 m_2) - B(\varphi) \cdot A \cdot c p_1 m_1}{(c p_1 m_1 + c p_2 m_2)}$$



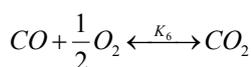
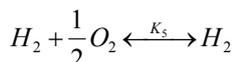
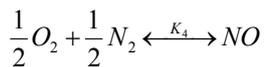
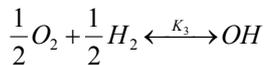
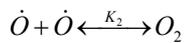
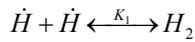
Modellierung und Simulation des ottomotorischen  
Verbrennungsprozesses mittels Ein-  
und Zweizonenmodell



## Chemisches Gleichgewicht

Berechnung des chemischen Gleichgewichts für 10 Spezies<sup>4</sup>

CO<sub>2</sub>, H<sub>2</sub>O, N<sub>2</sub>, O<sub>2</sub>, CO, H<sub>2</sub>, H, O, OH, NO



Bestimmung der Reaktionskonstanten über Approximation der JANAF-Tabellen (NASA)

$$\log(K_i) = A \ln(T/1000) + B/T + C + DT + ET^2$$

$$(600 \text{ [K]} < T < 4000 \text{ [K]})$$

Reduktion auf nichtlineares Gleichungssystem mit 4 Unbekannten. Lösung über Partialdruckverhältnisse mittels Newton-Verfahren

<sup>4</sup> SAE-Paper, Olikara, C & Borman, G.L. 1975 - A computer program for calculating properties of equilibrium...

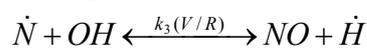
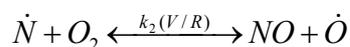
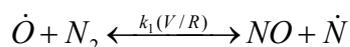


**Modellierung und Simulation des ottomotorischen  
Verbrennungsprozesses mittels Ein-  
und Zweizonenmodell**



## NO<sub>x</sub>-Berechnung

NO<sub>x</sub>-Bestimmung basiert auf dem erweiterten Zeldovich-Mechanismus



Geschwindigkeitskonstante über Arrhenius-Beziehung

$$k_{i(V/R)} = 1e^{-3} * A * T^B * e^{\frac{E*4.19}{R*T}}$$

$$K_{1V} = k_{1V}[N_2][O]$$

$$K_{1R} = k_{1R}[NO][N]$$

$$K_{2V} = k_{2V}[O_2][N]$$

$$K_{2R} = k_{2R}[NO][O]$$

$$K_{3V} = k_{3V}[N][OH]$$

$$K_{3R} = k_{3R}[NO][H]$$

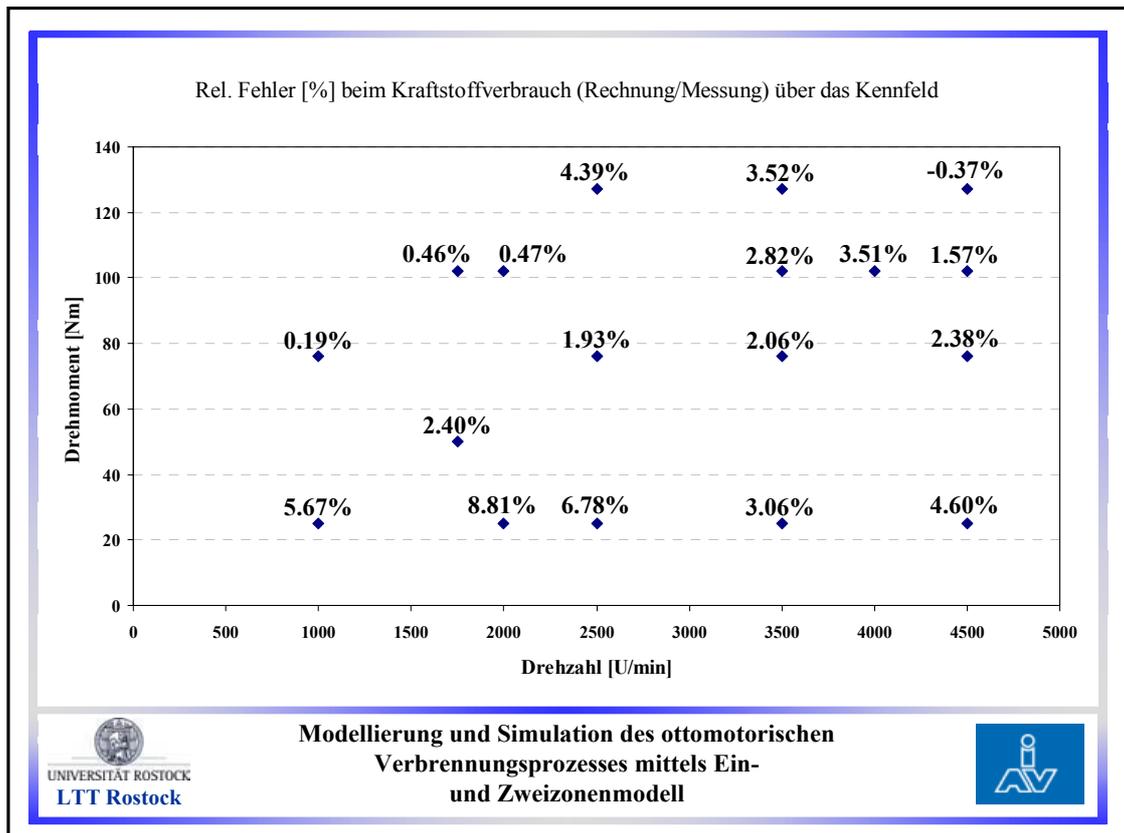
$$\Longrightarrow \frac{d[NO]}{dt} = K_{1V} - K_{1R} + K_{2V} - K_{2R} + K_{3V} - K_{3R}$$

<sup>5</sup> Dissertation Frank Weberbauer, TU München, 2001 – Thermodynamische Analyse und Vorausberechnung der ...



**Modellierung und Simulation des ottomotorischen  
Verbrennungsprozesses mittels Ein-  
und Zweizonenmodell**





- ### Zusammenfassung und Ausblick
- Entwicklung eines Einzonenmodells mit verschiedenen Untermodellen
    1. Berechnung der Innere Energie über Zacharias-Ansatz und mittels idealer Mischung realer Gase (VDI 4670)
    2. Verschiedene Modelle zur Bestimmung des Wandwärmeeüberganges
    3. Hohe Präzision bei der Bestimmung des Kraftstoffverbrauchs
    4. Geringe Rechenzeiten
    5. Schnelle Messwertverarbeitung und Diagnose möglich
  - Erweiterung auf ein Zweizonenmodell
    1. Aufteilung des Brennraumes in Reaktions- und Frischluftzone
    2. Bestimmung des chemischen Gleichgewichts in der Reaktionszone
    3. Berechnung der Stickoxidemissionen
- 
- Anpassung des Zweizonenmodells hinsichtlich der Stickoxidemissionen
  - Entwicklung eines neuen Ansatzes zur Bestimmung der Temperaturdifferenz bei Verbrennungsbeginn.
  - Erweiterung der Emissionsmodelle auf HC und CO
- Modellierung und Simulation des ottomotorischen Verbrennungsprozesses mittels Ein- und Zweizonenmodell**
- UNIVERSITÄT ROSTOCK  
LTT Rostock
-



# Two-Zone combustion model in S.I. engine using polytropic index

N. Chindaprasert, M.Eng.

Lehrstuhl für Technische Thermodynamik

Prof. Dr.-Ing. E. Hassel

Dr.-Ing. Jürgen Nocke

Dipl.-Ing. Christoph Janssen

IAV GmbH

Dipl.-Ing. H. Rabba

Dipl.-Ing. M. Köller



LTT  
Rostock

Two-Zone Combustion Model



## Outline

- Motive
- Goal of model
- Model description
- Results
- Conclusion



LTT  
Rostock

Two-Zone Combustion Model



## Motive

### Motives for 2 zone model

1. NOx model with good accuracy
2. Low calculating time
3. Extension of 1-zone (more detailed chemistry)
4. 1-zone model can not calculate NOx



LTT  
Rostock

Two-Zone Combustion Model



## Goals of the model

### The goals of the two zone model

- Nitrogen oxides (NOx)
- Mass fraction burned
- Temperatures of burned gas and unburned gas

(Mass fraction burned is ratio of burned mass and total mass)



LTT  
Rostock

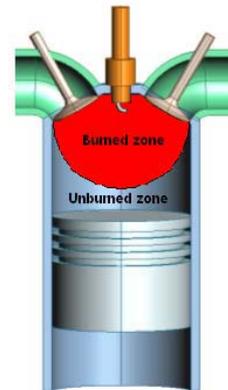
Two-Zone Combustion Model



## Model description

Two Zone combustion model divide the combustion chamber into two regions

- Burned zone
- Unburned zone



The calculation is based on

- Energy Equation
- Mass balance
- Chemical Equilibrium
- Ideal gas

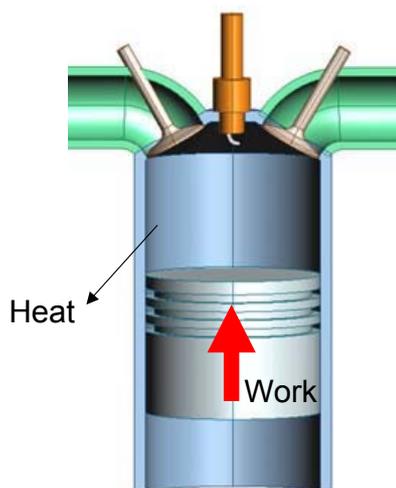


LTT  
Rostock

Two-Zone Combustion Model



## Compression process



There is only one zone,  
Unburned zone. (unburned mixture)

We use the Pressure-time data from  
the pressure sensor and the polytropic  
exponent to find the temperature

$$(P_i / P_{i+1})^{(n-1)/n} = (T_i / T_{i+1}) \quad (1)$$

$$n = -(dP/dV) \cdot (V/P) \quad (2)$$

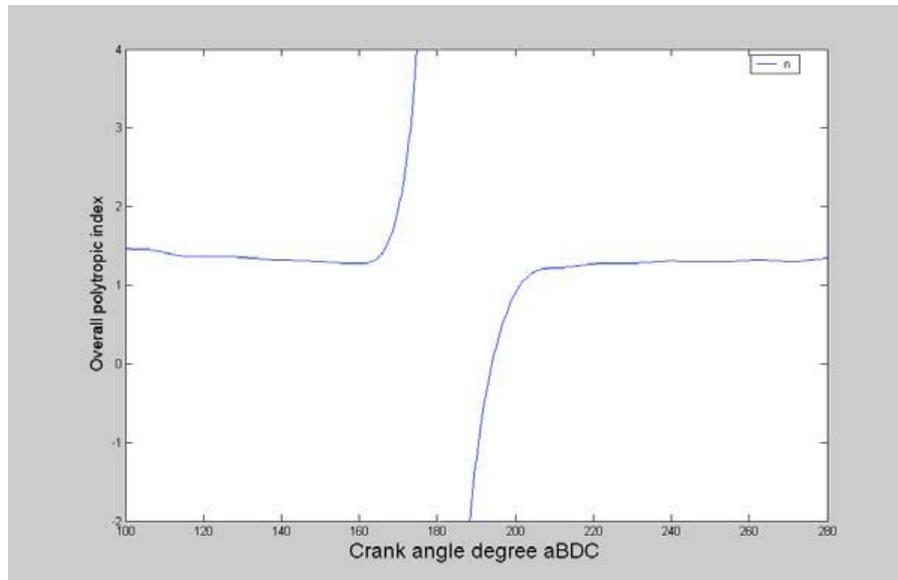


LTT  
Rostock

Two-Zone Combustion Model



## Model description



LTT  
Rostock

Two-Zone Combustion Model



## Combustion Model description

During the combustion process

$$P \cdot V = m_u \cdot R_u \cdot T_u + m_b \cdot R_b \cdot T_b$$

and  $m = m_b + m_u$  we can rewrite to

$$P \cdot V = m \cdot R_u \cdot T_u + m_b \cdot (R_b \cdot T_b - R_u \cdot T_u) \quad (3)$$

if we know P from the experiment, the only unknown of equation (3) are

$$m_b, T_b, T_u$$



LTT  
Rostock

Two-Zone Combustion Model



## Combustion Model description

For a constant mass closed system in equilibrium is undergoing a reversible polytropic process

$$dQ = m \cdot C_n \cdot dT \quad (4)$$

$C_n$  is specific heat

$$C_n = C_v \cdot (\gamma - n) / (1 - n) \quad (5)$$

$$\gamma = \frac{C_p}{C_v}$$

Heat transfer through the boundary

$$dQ = h_q \cdot A \cdot (T - T_w) \cdot dt \quad (6)$$



LTT  
Rostock

Two-Zone Combustion Model



## Combustion Model description

Combination equation (4) and (6) yields the following:

$$h_q \cdot A \cdot (T - T_w) \cdot dt = m \cdot C_n \cdot dT \quad (7)$$

Across a small time interval  $dt$ , equation (7) can be integrated by the trapezoidal rule

$$\left( h_{q,i+1} \cdot A_{i+1} \cdot (T_{i+1} - T_w) \right) / \left( h_{q,i} \cdot A_i \cdot (T - T_w) \right) = C_{n,i+1} / C_{n,i} \quad (8)$$



LTT  
Rostock

Two-Zone Combustion Model



## Combustion Model description

assumption

- the sum of the change in the value of the heat transfer coefficient  $\delta h_q$  and the external heat transfer area  $\delta A$  is small
- the changes in mass across the small time interval  $\delta m$  are too small compared to the changes of  $T$ ,  $\gamma$  and  $C_v$

$$(T_{z,i+1} - T_w) / (T_{z,i} - T_w) = C_{n_z,i+1} / C_{n_z,i} \quad (9)$$

Using the relationship of  $T_z$  and  $n_z$  for any zones across the interval of time

$$(P_i / P_{i+1})^{n_z - 1} = (T_{z,i} / T_{z,i+1})^{n_z} \quad (10)$$

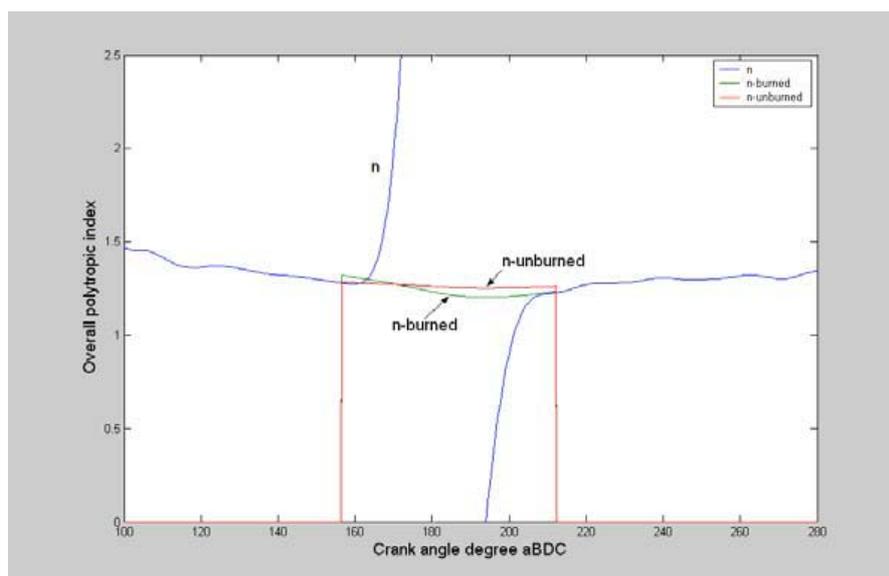


LTT  
Rostock

Two-Zone Combustion Model



## Result from 2 Zone



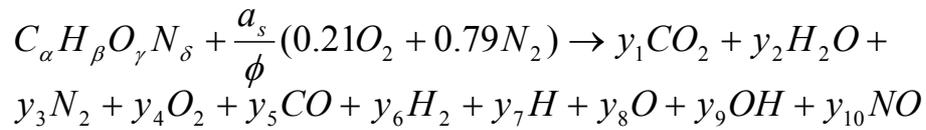
LTT  
Rostock

Two-Zone Combustion Model



## Equilibrium Combustion Products

### Combustion Equation



### Balance C,H,O,N

$$C: \quad \alpha = (y_1 + y_5)N$$

$$H: \quad \beta = (2y_2 + 2y_6 + y_7 + y_9)N$$

$$O: \quad \gamma + \frac{2a_s}{\phi} = (2y_1 + y_2 + 2y_4 + y_5 + y_8 + y_9 + y_{10})N$$

$$N: \quad \delta + \frac{3.76a_s}{\phi} = (2y_3 + y_{10})N \quad \text{where N is the total number of moles}$$



LTT  
Rostock

Two-Zone Combustion Model



## Equilibrium Combustion Products

$$\sum_{i=1}^{10} y_i - 1 = 0$$

Olikara and Borman (1975) have curve fitted the equilibrium constants  $K_i(T)$  to JANAF table

$$\log_{10} K_i(T) = A_i \ln\left(\frac{T}{1000}\right) + \frac{B_i}{T} + C_i + D_i T + E_i T^2$$



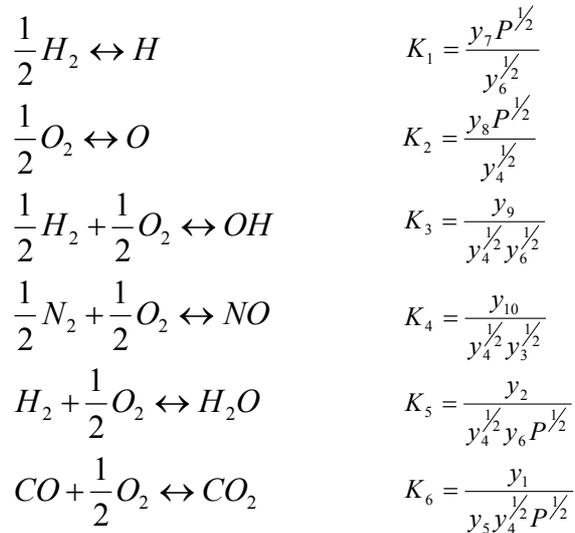
LTT  
Rostock

Two-Zone Combustion Model



## Equilibrium Combustion Products

Equilibrium constant  $K_i$  from



LTT  
Rostock

Two-Zone Combustion Model



## Equilibrium Combustion Products

From balancing C,H,O,N and equilibrium constant  
There are 11 equations with 11 unknowns

This set of 11 equations are nonlinear and solved  
by Newton Raphson iteration



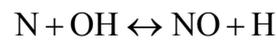
LTT  
Rostock

Two-Zone Combustion Model



## Kinetics of Nitric Oxide Formation

- NO is formed by the following reactions which make up the extended Zeldovich mechanism (Heywood, 1988):



- The rate of formation of NO can therefore be written as:-

$$\frac{d[\text{NO}]}{dt} = k_1^+ [\text{O}][\text{N}_2] + k_2^+ [\text{N}][\text{O}_2] + k_3^+ [\text{N}][\text{OH}] - k_1^- [\text{NO}][\text{N}] - k_2^- [\text{NO}][\text{O}] - k_3^- [\text{NO}][\text{H}]$$



LTT  
Rostock

Two-Zone Combustion Model



## Kinetics of Nitric Oxide Formation

- Equilibrium value
 
$$R_1 = k_1^+ [\text{O}]_e [\text{N}_2]_e = k_1^- [\text{NO}]_e [\text{N}]_e$$

$$R_2 = k_2^+ [\text{N}]_e [\text{O}_2]_e = k_2^- [\text{NO}]_e [\text{O}]_e$$

$$R_3 = k_3^+ [\text{N}]_e [\text{OH}]_e = k_3^- [\text{NO}]_e [\text{H}]_e$$

- Following a simplification using the approximation of steady-state N concentration

$$\frac{d[\text{N}]}{dt} = 0$$

- the following expression is obtained

$$\frac{d[\text{NO}]}{dt} = \frac{2R_1 \{1 - ([\text{NO}]/[\text{NO}]_e)^2\}}{1 + ([\text{NO}]/[\text{NO}]_e)R_1/(R_2 + R_3)}$$

This final expression can be integrated to give the Nitric Oxide concentration at time t



LTT  
Rostock

Two-Zone Combustion Model



# The results of 2-zone model

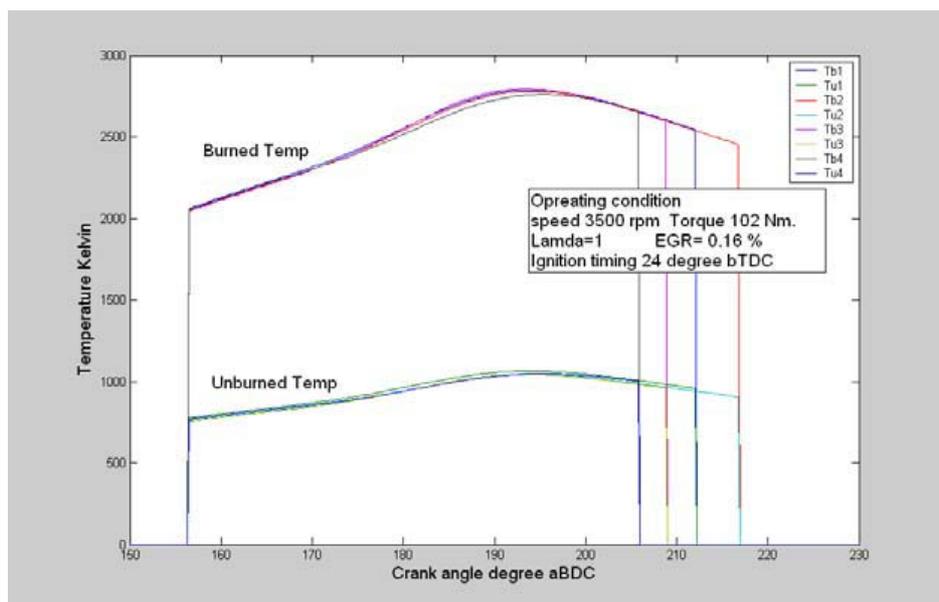


LTT  
Rostock

Two-Zone Combustion Model



## Temperature from 2 Zone

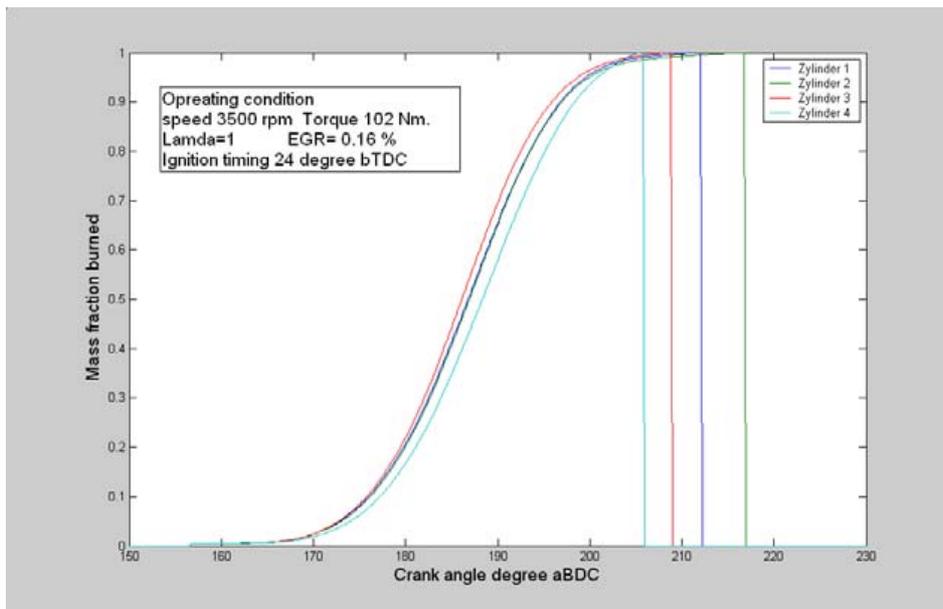


LTT  
Rostock

Two-Zone Combustion Model



## Result of mass fraction burned from 2 Zone

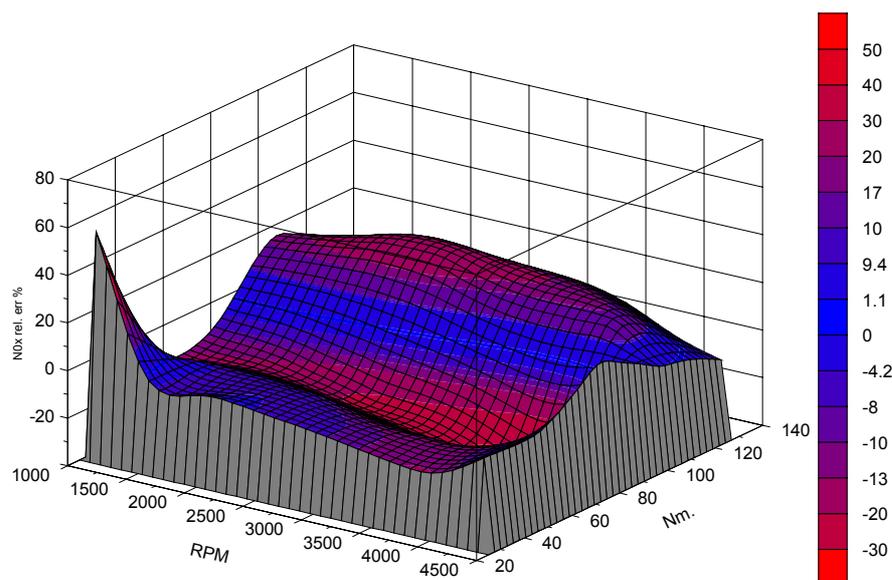


LTT  
Rostock

Two-Zone Combustion Model



## NOx rel.error %, normal condition

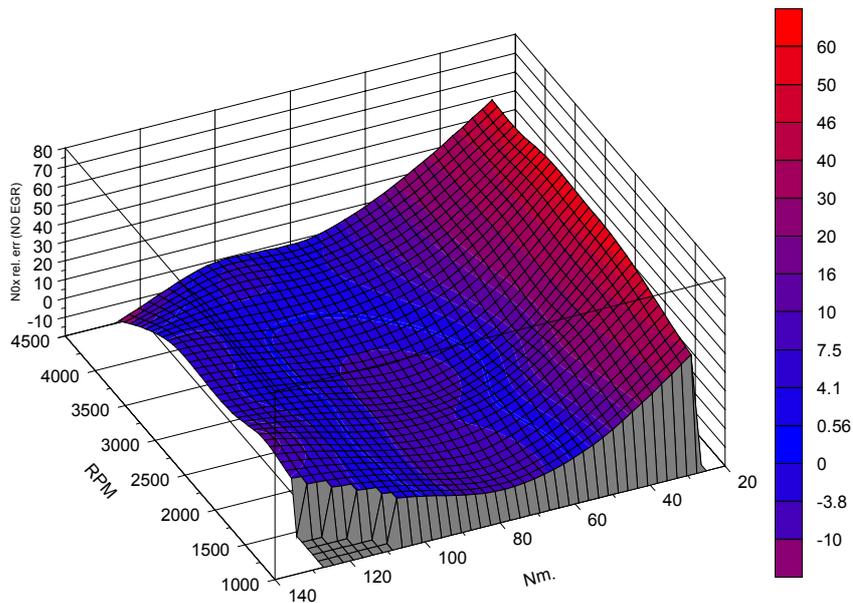


LTT  
Rostock

Two-Zone Combustion Model



# NOx rel.error %, NO EGR

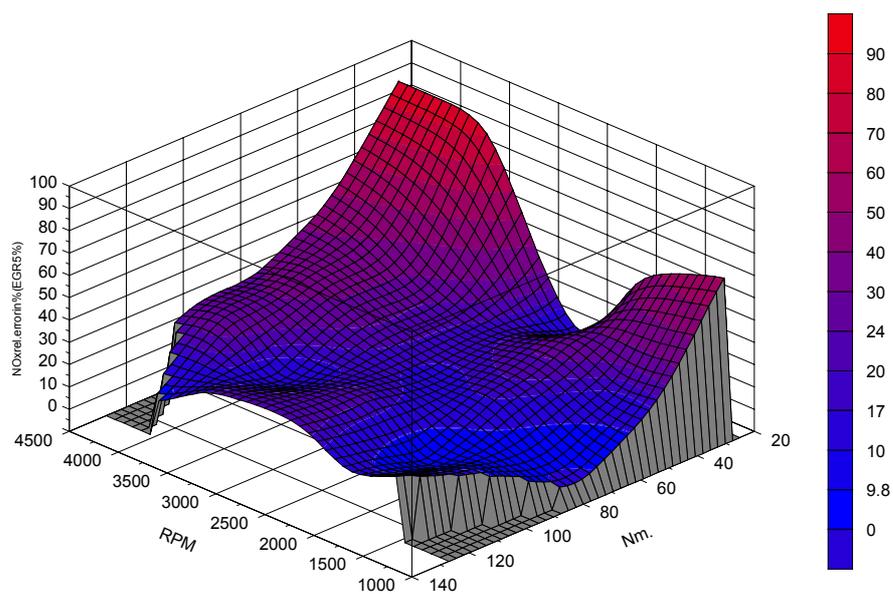


LTT  
Rostock

Two-Zone Combustion Model



# NOx rel.error %, 5% EGR



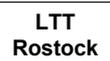
LTT  
Rostock

Two-Zone Combustion Model



## Conclusion

- The model predicts NO<sub>x</sub> with good results in medium to high loads.
- For normal conditions, the model predicts NO<sub>x</sub> in order of 20% relative error between medium to high loads.
- When engine operates without EGR, the model can predict NO<sub>x</sub> less than 10% relative error between medium to high loads in every engine speeds.



Two-Zone Combustion Model





---

# Rechnergestützte Katalysatorauslegung

ASIM – Workshop 2004  
Wismar

A. Schulze  
Lehrstuhl für Technische Thermodynamik  
Universität Rostock

Prof. Dr.-Ing. habil E. Hassel  
Dr.-Ing. J. Nocke  
Dipl.-Ing. C. Janssen

Dr. O. Magnor/IAV GmbH  
Dr. G. Hübner/IAV GmbH



---

## Überblick

- Einleitung/Motivation
- Modellierung
  - Speziesbilanz
  - Energiebilanz
- Chemie
  - Modellannahmen
  - Chemischen Reaktionen
- Ergebnisse
- Zusammenfassung

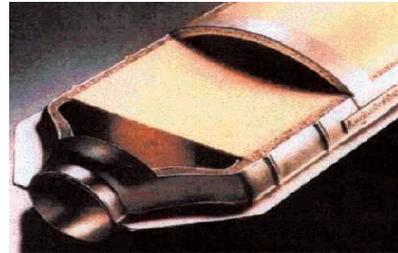


---

## Motivation

Kraftfahrzeuge mit fossilen Brennstoff (Benzin, Diesel) erzeugen große Mengen an Schadstoffen

- Unvollständige Verbrennung im Zylinder
  - Reaktionen haben zu wenig Zeit
  - Flammenverlöschen an Wänden
  - inhomogenes Gemisch  
(mager-> NO<sub>x</sub>, fett-> C<sub>x</sub>H<sub>y</sub>, CO)



**Katalysatoren spielen Abgasnachbehandlung eine große Rolle**

heute: 90% der Schadstoffe werden umgewandelt  
problematisch: Kaltstartverhalten

---

## Motivation

Simulation von Abgasmessungen am PC führt zu:

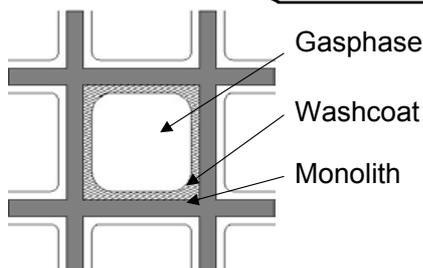
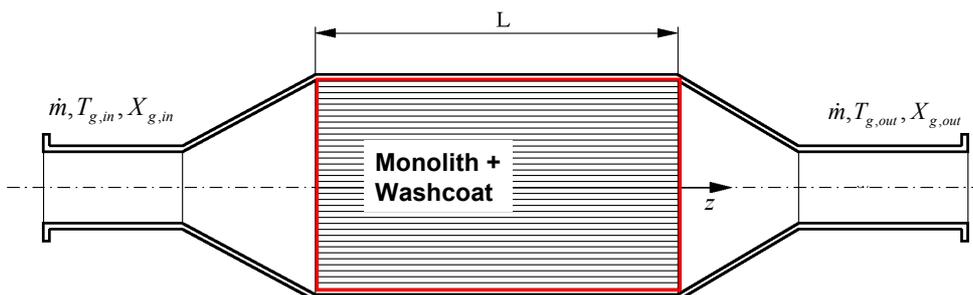
- Kostengünstigen Abgasuntersuchungen
  - Parametervariationen (Durchmesser, Länge, Zellzahl)
- Anpassung der Motorparameter
- Abschätzung der Emissionspotentiale bereits in Entwicklungsphase
  - Optimierung des Emissionsverhaltens
  - preiswertere Modellentwicklung

 **Verbesserung des Emissionsverhaltens  
Weniger Schadstoffausstoß**

## Anforderungen an das Modell

- Eingangsdaten:
  - Abgasmassenstrom, Temperatur-, Abgaszusammensetzung aus Messungen
  - Katalysatorparameter
- Ausgangsdaten:
  - Konvertierungsverhalten des Katalysators
  - Aufheizen, ab wann ist der Kat heiß genug?
- Liefert Aussagen über
  - Katalysatorauswahl (Geometrie)
  - Platzierung des Kats
  - Mögliche  $\lambda$ -Fenster

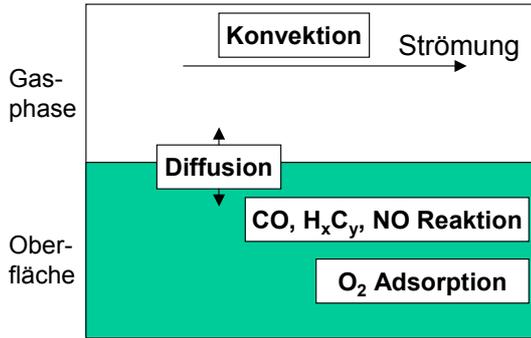
## Modellierung



### Annahmen:

- Homogene Anströmung
- Temperatur konstant über Querschnitt
- 1D-Strömung in Zellen

# Massenbilanz



keine chem. Reaktionen  
Konvektiver Stofftransport  
Stoffaustausch über Oberfläche

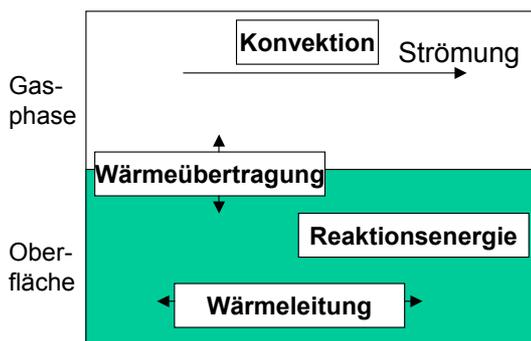
chem. Reaktionen  
Diffusiver Stofftransport  
Stoffaustausch über Oberfläche  
Sauerstoffeinspeicherung in Washcoat

**Speziesbilanz:**

Gas-phase	$\varepsilon w \frac{\partial X_{g,k}}{\partial z} + GSA \beta_k (X_{g,k} - X_{s,k}) = 0$
Oberfläche	$\beta_k (X_{g,k} - X_{s,k}) = \dot{r}_k$



# Energiebilanz



keine chem. Reaktionen  
Keine Wärmeleitung  
Wärmeübertragung an Oberfläche

chem. Reaktionen  
Axiale Wärmeleitung  
Wärmeübertragung an Gasphase  
Aufheizung

**Energieerhaltung:**

Gas-phase	$\rho_g c_{p,g} w \varepsilon \frac{\partial T_g}{\partial z} + GSA \alpha (T_g - T_s) = 0$
Oberfläche	$(1 - \varepsilon) \rho_s c_{p,s} \frac{\partial T_s}{\partial t} - GSA \alpha (T_g - T_s) - (1 - \varepsilon) \lambda_s \frac{\partial^2 T_s}{\partial z^2} = \dot{r}_T$



## Modellierung der Reaktionen

- Verwendung von Bruttoreaktionen und mit reduzierter Chemie
  - Keine Gleichgewichtschemie
  - Kurze Rechenzeiten: z.B. 1 MVEG-Test 2min
- Verwendung empirischer Geschwindigkeitsgesetze
  - Langmuir-Hinshelwood Ansätzen
  - beinhalten die Diffusion innerhalb des Washcoats („lumped parameter“)
- Sauerstoffeinspeicherung ist zeitabhängig
  - Untermodell notwendig, welches die Fülle des Speichers modelliert
  - Nach Koltsakis/Stamatelos, wobei ein Füllungsgrad definiert wird
    - ↳  $0 \leq \Psi \leq 1$
  - Oxygen Storage Capacity – Washcoateigenschaft
- Modell muss auf berechneten Fall angepasst werden
  - Anpassung der chemischen Parameter (z.B. präexponentiellen Faktoren)



## Chemische Reaktionen

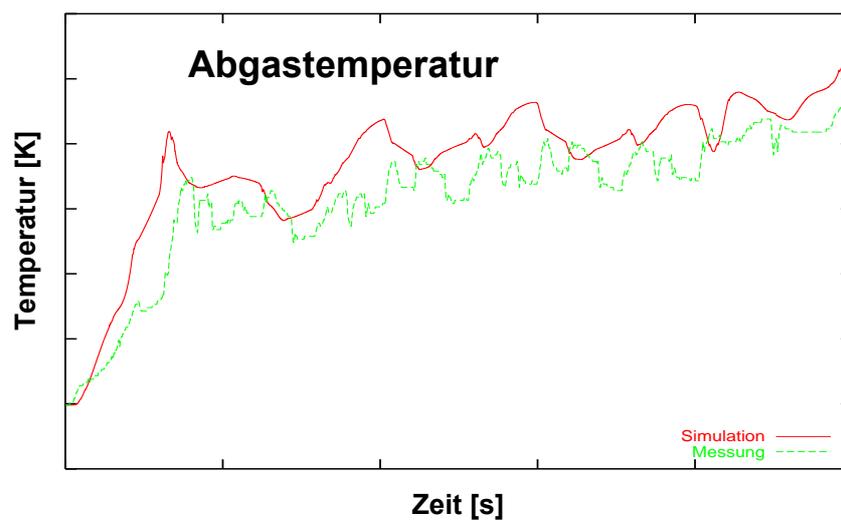
- CO – Abbau, NO-Abbau  $CO + \frac{1}{2}O_2 \Rightarrow CO_2$
- HC- Abbau  $CO + NO \Rightarrow CO_2 + \frac{1}{2}N_2$ 
  - Verwendung von 2 repräsentativen Kohlenwasserstoffen
  - Unterteilung in langsam (15%) und schnell (85%) reagierend
$$C_3H_6 + 4,5O_2 \Rightarrow 3CO_2 + 3H_2O$$
$$C_3H_8 + 5O_2 \Rightarrow 3CO_2 + 4H_2O$$
- Wasserstoff-Oxidation:  $H_2 + \frac{1}{2}O_2 \Rightarrow H_2O$
- Sauerstoffeinspeicherungsreaktionen:  $Ce_2O_3 + \frac{1}{2}O_2 \Leftrightarrow 2CeO_2$



## Berechnungsergebnisse

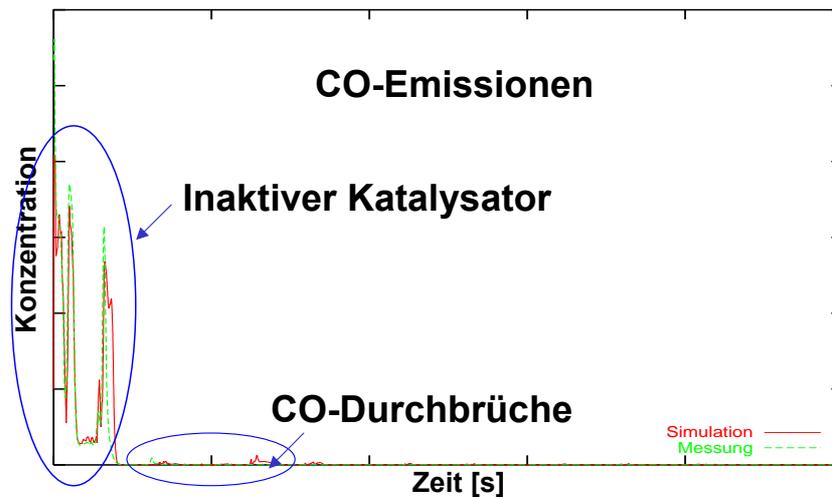
- Simulation eines MVEG – Fahrzyklus
- Hauptkatalysator von einem mit Flüssiggas betriebenen Fahrzeug
- Flüssiggasbetriebener Motor
  - Relativ geringe Abgastemperaturen ( $< 650^{\circ}\text{C}$ )

## Ergebnisse



**Gute Übereinstimmung**

## Ergebnisse



Nicht angepasste Parameter führen zu verkehrten Ergebnissen

## Zusammenfassung

- Abgastemperatur wird gut nachgebildet
- Aufgrund der vereinfachten Chemie ist eine Anpassung der Reaktionsparameter (z.B: präexponentielle Faktoren) notwendig
- Wenn Modellierung ok, dann können Parametervariationen durchgeführt werden
  - Kat-Typ, Kat-Geometrie, Kat-Position usw.

---

## Zusammenfassung

Katalysatormodellierung hat u.a. folgende Vorteile:

- Einsparung teurer Prüfstandsmessungen
- Emissionsabschätzungen bereits in Entwicklung
- Katalysatorauslegung bereits in der Vorentwicklung
- Verbesserung des Emissionsverhaltens

# **Model Predictive Control – Erfahrungen der Anwendung in der Prozessindustrie**

R. Dittmar, FH Westküste Heide (Holst.)\*

## **Zusammenfassung**

Modellgestützte prädiktive Regelungsverfahren sind seit einer Reihe von Jahren das Mittel der Wahl für die Lösung anspruchsvoller Mehrgrößenregelungsprobleme in der Prozessindustrie. Aufsatz und Vortrag gehen Ursachen für Erfolg der Anwendung dieser Advanced-Control-Strategie nach, geben eine Übersicht über kommerziell verfügbare MPC-Tools und charakterisieren gegenwärtige Entwicklungstendenzen.

## **1 Einleitung**

Mit dem Begriff „Modellbasierte prädiktive Regelungen“ (Model Predictive Control, MPC) wird eine ganze Klasse von Regelungsalgorithmen bezeichnet, die sich dadurch auszeichnen, dass ein Modell für das dynamische Verhalten des Prozesses nicht nur in der Entwurfsphase, sondern explizit im laufenden Betrieb des Reglers benutzt wird. Es wird dort für die (Langzeit-)Vorhersage des Verhaltens der Regelgrößen eingesetzt. Auf der Grundlage dieser Prädiktion werden die erforderlichen Stellgrößenänderungen durch Lösung eines Optimierungsproblems in Echtzeit bestimmt.

Modellbasierte prädiktive Regelungen wurden bereits in den 70er Jahren des vorigen Jahrhunderts durch industrielle Regelungstechniker entwickelt und im Raffineriesektor und der Petrochemie eingesetzt, bevor sie größere Aufmerksamkeit auch im akademischen Bereich erregten. In den letzten zehn bis fünfzehn Jahren hat sich das Bild jedoch grundlegend gewandelt: für MPC mit linearen Modellen existieren inzwischen ausgereifte theoretische Grundlagen, die Zahl der Veröffentlichungen ist explosionsartig angestiegen. Das Potenzial der MPC-Technologie wird inzwischen nicht nur in den Bereichen genutzt, von denen die Entwicklung ausging. Es werden zunehmend neue Anwendungsfelder erschlossen. Die Zahl der industriellen Einsatzfälle hat sich allein in den letzten fünf Jahren verdoppelt. Kein anderes der gehobenen Regelungsverfahren weist eine solche Erfolgsgeschichte der industriellen Anwendung auf. MPC ist heute das „Arbeitspferd“ für die Lösung anspruchsvoller Regelungsaufgaben in der

---

\* Studiengang E/I, D-25746 Heide, Germany, dittmar@fh-westkueste.de

Verfahrensindustrie. Eine gute Übersicht über Geschichte und Entwicklungsstand von MPC-Technologien vermittelt [QB03].

Im vorliegenden Beitrag wird versucht, die Ursachen für diesen Erfolg aus den steigenden Anforderungen an die Prozessregelung und den MPC-Anwendungseigenschaften zu begründen. Es wird eine kurze Übersicht über kommerziell verfügbare Entwicklungsumgebungen für MPC-Reglungen mit linearen Prozessmodellen gegeben und der typische Projektablauf beschrieben. Abschließend wird auf gegenwärtige Tendenzen der Weiterentwicklung von MPC-Technologien eingegangen.

## 2 Erfolgsfaktoren der industriellen Anwendung von MPC

Prädiktive Regelungsalgorithmen weisen in ihrer „Minimalversion“ folgende gemeinsame Grundstruktur auf (Abb.1).

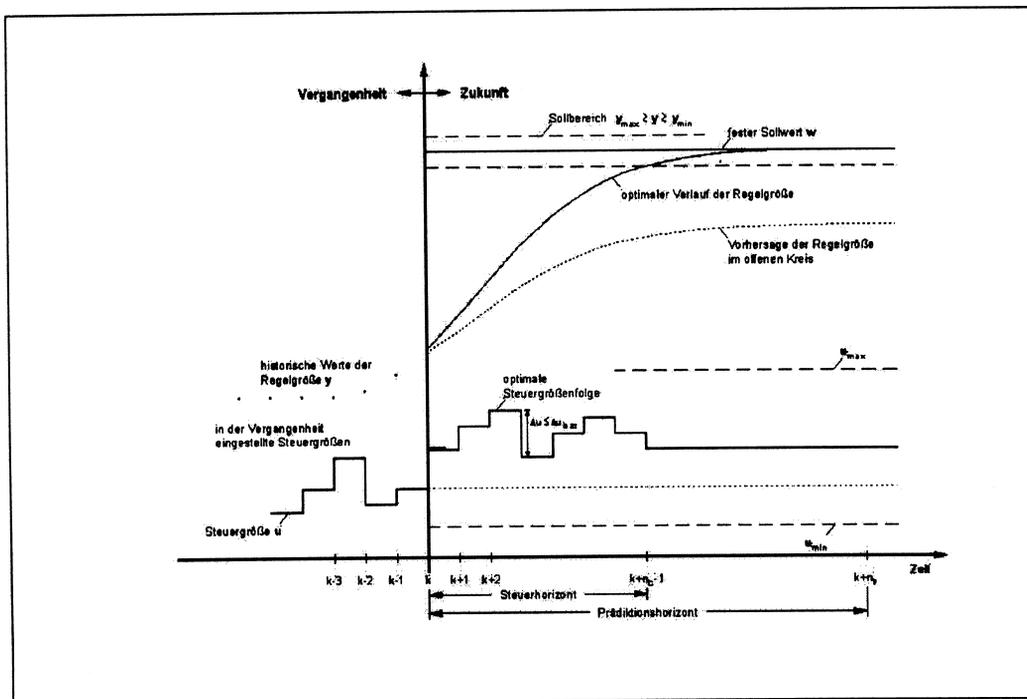


Abbildung 1: Grundprinzip der modellbasierten prädiktiven Regelung

1. Vorhersage des zukünftigen Verlaufs der Regelgröße  $y$  und der Regeldifferenz  $e$  über einen Prädiktionshorizont auf der Grundlage gemessener historischer Werte der Stellgrößen  $u$ , der Reglgrößen  $y$  und evtl. messbarer Störgrößen  $z$ , eines gegebenen zukünftigen Sollwerts  $w$  und eines gegebenen Modells für das dynamische Verhalten des Prozesses. Obwohl dieses Modell im Prinzip jede beliebige Form annehmen kann, ist für die meisten praktisch eingesetzten MPC

Regler die Verwendung eines linearen Prozessmodells charakteristisch, das durch experimentelle Prozessidentifikation ermittelt wird.

2. Bestimmung einer Folge zukünftiger Stellgrößenänderungen  $\Delta u$  über einen gegebenen Steuerhorizont durch Lösung eines Optimierungsproblems im Echtzeitbetrieb. Ziel ist es dabei, die vorhergesagten Regeldifferenzen mit möglichst kleinem Stellaufwand zu minimieren. Diese Vorgehensweise ermöglicht es, Nebenbedingungen für die Stellgrößen (Stellgrößenbereich und Verstellgeschwindigkeit) explizit im Regelalgorithmus zu berücksichtigen. Für die Regelgrößen können alternativ zu Sollwerten auch Gutbereiche oder obere/untere Grenzwerte vorgegeben werden.

3. Anwendung des Prinzips des gleitenden Horizonts. Das bedeutet, dass trotz Berechnung einer ganzen Folge zukünftiger Stellgrößenänderungen im vorhergehenden Schritt nur das erste Element dieser Folge implementiert wird. Nach Verschiebung des betrachteten Zeithorizonts um ein Abtastintervall nach vorn erfolgt im nächsten Abtastintervall eine Wiederholung der gesamten Prozedur.

4. Mit Hilfe des im aktuellen Abtastintervall jeweils neu eintreffenden Messwerts für die Regelgröße  $y$  wird die modellgestützte Vorhersage der Regelgröße fortlaufend korrigiert und auf diese Weise der Regelkreis geschlossen. Auf diese Weise werden nicht gemessene Störgrößen und die i.a. vorhandene Nichtübereinstimmung von Prozessmodell und realem Prozessverhalten im Regelalgorithmus berücksichtigt.

Moderne MPC-Programmsysteme gehen i.A. über diese vier Grundelemente hinaus. Das hängt damit zusammen, dass MPC-Regelungen in der Praxis fast ausschließlich für Mehrgrößenregelungen mit einer frei konfigurierbaren Zahl von Steuer-, Regel- und messbaren Störgrößen eingesetzt werden. Die wichtigsten Erweiterungen des Grundalgorithmus sind:

5. Bestimmung der aktuellen Struktur des Mehrgrößensystems. Bevor die oben beschriebenen Schritte abgearbeitet werden, wird die aktuelle Struktur des Mehrgrößensystems bestimmt. Oder anders ausgedrückt: es wird selbständig die verfügbare Teilmenge aus der Gesamtmenge der beim Regelungsentwurf zunächst konzipierten Steuer-, Regel- und messbaren Störgrößen ermittelt. Dieser Schritt trägt dem Umstand Rechnung, dass sich in Folge verschiedener Ursachen die Zahl der verfügbaren Steuergrößen und die Zahl der zu berücksichtigenden Regelgrößen im Laufe der Zeit ändern kann. Typische Beispiele hierfür sind vorübergehende Ausfälle von Sensoren oder Aktoren. Statt die Mehrgrößenregelung in diesem Fall vollständig außer Betrieb zu setzen wird dann versucht, die mit den noch verfügbaren Variablen bestmögliche Lösung zu erreichen. Man kann diesen Schritt auch als automatische Rekonfiguration des MPC-Reglers auffassen.

6. Statische Arbeitspunktoptimierung. Die Berechnung der optimalen Steuergrößenfolge im Schritt 2 sichert noch nicht, dass die MPC-Regelung im

stationären Zustand einen betriebswirtschaftlich optimalen Arbeitspunkt der Anlage ansteuert. Daher wird parallel mit der Lösung der dynamischen Optimierungsaufgabe ein funktional übergeordnetes, aber programmtechnisch integriertes statisches Optimierungsproblem gelöst, wenn dafür genügend Freiheitsgrade bestehen. Das Ziel besteht darin, die für den stationären Zustand der Anlage optimalen Werte der Steuer- und Regelgrößen durch Minimierung oder Maximierung einer *ökonomischen* Zielfunktion zu ermitteln. Die dynamische Optimierung bestimmt dagegen den günstigsten Weg zu diesem Ziel. Es muss betont werden, dass die in die MPC-Algorithmen integrierte Form der Arbeitspunktoptimierung lokalen Charakter hat, sie bezieht sich nur auf den Teil der Anlage, der durch die in den MPC-Regler eingehenden Steuer- und Regelgrößen umfasst wird.

Die Ursachen für den anhaltenden Erfolg von MPC-Technologien in der Prozessindustrie liegen vor allem darin, dass sie Anwendungseigenschaften aufweisen, die einer Reihe von praktischen Anforderungen und Gegebenheiten der Regelung komplexer verfahrenstechnischer Anlagen in besonderer Weise gerecht werden. Die wichtigsten sollen im Folgenden angerissen werden.

1. Viele verfahrenstechnische Prozesse besitzen einen ausgeprägten Mehrgrößencharakter. In einer Reihe von Fällen sind die Wechselwirkungen zwischen den Prozessgrößen so groß, dass der Einsatz eines zentralen Mehrgrößenreglers zu einer deutlichen Verbesserung der Anlagenfahrweise führen kann, und sich angestrebte Durchsatz- und Qualitätsziele besser erreichen lassen. MPC-Regelalgorithmen lassen sich einfach vom Eingrößen- auf den Mehrgrößenfall erweitern und sind daher für die Regelung von verfahrenstechnischen Mehrgrößensystemen und die Einbeziehung messbarer Störgrößen besonders geeignet.
2. In verfahrenstechnischen Prozessen treten Beschränkungen (Ungleichungs-Nebenbedingungen) sowohl für die Stell- als auch für die Regelgrößen auf. Stoff- und Energieströme lassen sich nur in bestimmten Bereichen und mit einem begrenzten „Tempo“ manipulieren. Für Regelgrößen können neben Sollwerten auch obere und untere Grenzwerte als Vorgaben auftreten, z.B. Mindestanforderungen an die Reinheit eines Produktstroms. Häufig liegt der optimale Betriebspunkt einer Anlage an einem oder sogar an mehreren dieser Nebenbedingungen. MPC-Regelungen sind die einzigen bekannten Regelungsalgorithmen, in denen Begrenzungen (constraints) für die Steuer- und Regelgrößen vorgegeben werden können, die im Regelalgorithmus selbst explizit und systematisch berücksichtigt werden.
3. Der optimale Arbeitspunkt einer Anlage ist nicht unveränderlich, sondern er variiert mit der Zeit und den Bedingungen, unter denen die Anlage betrieben wird. Zu solchen Veränderungen gehören schwankende Rohstoffzusammensetzungen und nicht konstante Heizwerte von Brennstoffen ebenso wie sich ändernde Umgebungsbedingungen, schwankende Preise für Rohstoffe und Energien sowie

der sich ändernde Bedarf für die erzeugten Produkte. Es ist daher betriebswirtschaftlich sinnvoll, den optimalen Arbeitspunkt einer Prozessanlage fortlaufend zu ermitteln und anzufahren. Damit lassen sich erhebliche Kosteneinsparungen erzielen bzw. Gewinnerhöhungen realisieren, die zu Wettbewerbsvorteilen führen. In vielen Zweigen der Prozessindustrie ist das angesichts der in den letzten Jahren sinkenden Kapitalrendite von großer Bedeutung. MPC-Regelungen verfügen über eine integrierte Funktion der lokalen statischen Arbeitspunktoptimierung, die es ermöglicht, die in der aktuellen Situation günstigsten stationären Werte der Steuer- und Regelgrößen zu ermitteln und den Prozess in die Richtung dieses Optimums zu lenken.

4. Unter Produktionsbedingungen lassen sich Ausfälle von Mess- und Stelleinrichtungen nicht völlig vermeiden. Darüber hinaus sind nicht nur bei Anlagenabstellungen, sondern mitunter auch im laufenden Betrieb der Anlage Wartungsarbeiten erforderlich. Größere Störungen können vorübergehend intensive Bedienereingriffe und Hand-Fahrweisen erforderlich machen. Dadurch kann die Situation entstehen, das ursprünglich für eine MPC-Regelung vorgesehene Steuer- und/oder Regelgrößen zeitweilig nicht zur Verfügung stehen. Es wäre dann kontraproduktiv, wenn man in diesen Situationen jeweils die gesamte Mehrgrößenregelung außer Betrieb nehmen oder neu konfigurieren müsste. MPC-Regelungen verfügen über die erforderliche Strukturflexibilität, um auf die sich ändernde Zahl von verfügbaren Steuer- und zu berücksichtigenden Regelgrößen selbständig zu reagieren.

5. Advanced-Control-Konzepte haben praktisch nur eine Chance zur Verwirklichung, wenn die für ihren Entwurf, ihre Inbetriebnahme und ihre Pflege aufzuwendenden Personal- und Sachmittel in einem vernünftigen Verhältnis zu den zu erwartenden Ergebnissen stehen. Nun darf nicht verschwiegen werden, dass die Kosten für Advanced-Control-Projekte unter Nutzung der MPC-Technologie nicht zu unterschätzen sind. Auf der anderen Seite gibt es eine Reihe von Tendenzen, die zu einer Senkung dieser Kosten führen oder in Zukunft führen werden. Dazu gehören u.a.

- die Verkürzung von Anlagentests durch die Einführung fortgeschrittener Methoden der Systemidentifikation
- die Entwicklung ausgereifter Werkzeuge für die Modellbildung, den Reglerentwurf und die Simulation des geschlossenen Regelungssystems
- die Entwicklung von Standards und wiederverwendbaren Plattformen für die Projektabwicklung
- die Entwicklung standardisierter Datenschnittstellen wie z.B. OPC (OLE for Process Control), durch die die zeitaufwändige und fehleranfällige Sonderentwicklung von speziellen Schnittstellen überflüssig wird
- die Entwicklung standardisierter Bedienbilder für den Online-Betrieb von MPC-Reglern, so dass es heute meist möglich ist, auf die projektspezifische Entwicklung von Bedienoberflächen zu verzichten

- die Bereitstellung von Werkzeugen für die Überwachung und Bewertung der Arbeitsweise von MPC-Regelungen im Dauerbetrieb (Control Performance Monitoring)

Die Weiterentwicklung der MPC-Technologie führt daher zu sinkenden Projektkosten (relativ zu den sonstigen Kosten der Automatisierung) bei steigender Qualität der Anwendungen, was die Erschließung weiterer Einsatzgebiete begünstigt.

6. Natürlich verlangt die *Entwicklung* von MPC-Verfahren – einschließlich der dafür notwendigen Modellbildung – vertiefte Kenntnisse der Regelungstheorie und der Systemidentifikation, wie sie z.B. in [Mac02, CB04, Zhu01, DP04] vermittelt werden. Für den *Anwender* solcher Systeme ist das Grundprinzip jedoch unmittelbar verständlich und transparent. Dies wird besonders deutlich beim Vergleich mit anderen modernen Regelungsmethoden. Man muss dabei in Rechnung stellen, dass selbst die in den Unternehmen mit der Betriebsbetreuung von Advanced-Control-Systemen beauftragten Verfahrens- und Automatisierungsingenieure i.A. keine Regelungstechnik-Spezialisten sind. Die vorhandenen Grundkenntnisse reichen aber aus, um sich das erforderliche Spezialwissen durch Schulung und Training anzueignen.

### **3 Projektablauf und kommerziell verfügbare MPC-Entwicklungsumgebungen**

Bei der Abwicklung von Advanced-Control-(AC)-Projekten unter Nutzung der MPC-Technologie hat sich ein Vorgehen in folgenden Schritten bewährt:

- Herausarbeitung der ökonomischen Ziele des AC-Projektes und Sammlung von Daten zur Beschreibung des Istzustandes der Anlagenfahrweise
- Erarbeitung einer Kosten-Nutzen-Analyse und Konzipierung der AC-Funktionen
- Überprüfung der Basisautomatisierung, insbesondere der in das MPC-Konzept einzubeziehenden PID-Basisregelungen und Lösung anlagentechnischer Probleme
- Entwicklung, Implementierung und Inbetriebnahme von Regel- und Rechenschaltungen unter Nutzung von PLS-Software-Funktionsbausteinen
- Vorbereitung und Durchführung von aktiven Versuchen an der Prozessanlage mit dem Ziel der Modellbildung
- Entwicklung von Prozessmodellen für das dynamische Verhalten der Mehrgrößen-Regelstrecke in allen Steuer- und Störkanälen unter Verwendung von Programmpaketen zur Prozessidentifikation
- Konfiguration des MPC-Reglers, Offline-Simulation des geschlossenen Regelungssystems, Grobeinstellung der Reglerparameter
- Portierung des MPC-Reglers auf die Zielhardware, Inbetriebnahme und Feineinstellung der Reglerparameter
- Gestaltung einer nutzerfreundlichen Bedienoberfläche für die Anlagenfahrer

- Inbetriebnahme und Feineinstellung im geschlossenen Regelkreis
- Training und Dokumentation
- Evaluierung der Ergebnisse des AC-Projekts
- Wartung, Pflege und Anpassung des MPC-Regelungssystems im laufenden Anlagenbetrieb

In [DP04 und Di04a] wird dieser Projektablauf im einzelnen beschrieben und anhand von erfolgreich abgeschlossenen industriellen Einsatzfällen illustriert. Tabelle 1 gibt eine (sicher unvollständige) Übersicht über Produkte und Anbieter kommerziell verfügbarer MPC-Programmsysteme, deren Komponenten eine gute Unterstützung für die Durchführung der wichtigsten o.g. Projektschritte geben. In den Bereichen Mineralölindustrie und Petrochemie ist es üblich, spezialisierte Ingenieurfirmen (meist die Anbieter der Programmsysteme selbst) mit der Projektdurchführung zu beauftragen. Die Erfahrung zeigt jedoch, dass eine intensive Mitarbeit des späteren Anwenders in allen Projektphasen den Erfolg maßgeblich beeinflusst. Dies setzt das Vorhandensein einer gewissen personellen Mindestkapazität auf dem Gebiet Advanced Control beim Anwender voraus.

Tabelle 1: Kommerziell verfügbare MPC-Programmsysteme

<b>Programmsystem</b>	<b>Anbieter</b>
DMCplus, Aspen Apollo	Aspen Technology
Profit Controller	Honeywell
Process Perfector	Pavilion Technologies
SMOC	Shell Global Solutions
Connoisseur	Invensys (Foxboro)
Predict & Control	ABB
MVC	General Electric
INCA	IPCOS Technology
IDCOM/HIECON	Adersa
STAR	DOT Products
GMAX	Intelligent Optimization Group
INDISS-MVAC	RSI
Brainwave MultiMax	Brainwave
NOVA NLC	PAS (Dot Products)
DeltaV Predict	Emerson Process Management (Fisher-Rosemount)

MPC-Programmsysteme bestehen i.A. aus folgenden Komponenten:

#### Offline-Werkzeuge

- Werkzeuge zur Identifikation von Mehrgrößensystemen (Testsignalentwurf, Datenvorverarbeitung, Parameterschätzung in parametrischen und nicht-parametrischen Modellen, Modellvalidierung, ...)

- Werkzeuge zur Optimierung unterlagerter PID-Basisregelungen
- Werkzeuge für die Regler-Konfiguration und den MPC-Reglerentwurf
- Werkzeuge zur Simulation des geschlossenen Regelungssystems

#### Online-Werkzeuge

- Online-Version des MPC-Regelungsalgorithmus
- Werkzeuge zum Kommunikation mit dem unterlagerten Prozessleitsystem
- Bedienoberfläche für die Anlagenfahrer
- Werkzeuge zur fortlaufende Beobachtung und Bewertung der MPC-Regelung
- Werkzeuge zur Koordinierung mehrerer MPC-Regelungen in großen Anlagen

## 4 Gegenwärtige Entwicklungstendenzen

Seit der erstmaligen Anwendung von MPC-Regelungen im Raffineriebereich Ende der 70er Jahre haben bereits mehrere Generationswechsel der MPC-Software stattgefunden. Obwohl man sagen kann, dass MPC-Regelalgorithmen unter Verwendung linearer Prozessmodelle inzwischen einen hohen Reifegrad erreicht haben, finden in deren Umfeld signifikante Weiterentwicklungen statt. Hervorzuheben sind insbesondere die folgenden:

1. Traditionell werden in der Phase der Anlagentests alle als Ausgangsgrößen des MPC-Reglers vorgesehenen Prozessvariablen nacheinander mehrfach manuell sprungförmig verstellt und die Zeitverläufe der Regelgrößen beobachtet und aufgezeichnet. Die so gewonnenen Datensätze bilden die Grundlage für die rechnergestützte Prozessidentifikation. Insbesondere bei Reglern mit einer größeren Zahl von Stellgrößen oder/und bei sehr trägen Prozessen resultiert diese Vorgehensweise in langen Versuchszeiträumen und entsprechend hohen Kosten. Daher gibt es in jüngster Zeit verstärkt Bemühungen, Werkzeuge zu entwickeln, die eine automatische simultane Anregung mehrerer Stellgrößen, die Verwendung multipler PRBS-Signale, die Identifikation im geschlossenen Regelkreis und die Gewinnung besserer Aussagen über die erreichte Modellgüte unterstützen [Zhu2004]. Die praktische Erfahrung zeigt, dass es auf diese Weise gelingt, den Zeitaufwand für diese kostenintensive Projektphase spürbar zu verringern. Beispiele hierfür sind Smart Step (Aspen), Profit Stepper (Honeywell) und Tai-Ji ID (Tai-Ji Control).

2. In der Vergangenheit wurden in einer Reihe von Einsatzfällen MPC-Regler für Teilprozesse/Teilanlagen mit einem übergeordneten Programm zur globalen (anlagenweiten) statischen Arbeitspunktoptimierung auf der Basis eines theoretischen Prozessmodells gekoppelt. Das hat einige Nachteile: hoher Aufwand, Verwendung verschiedener Modelle für die MPC-interne lokale und die globale Arbeitspunktoptimierung, keine dynamische Koordinierung der individuellen

MPC-Regler. Daher ist man in jüngster Zeit dazu übergegangen, bei Projekten in Großanlagen (z.B. Ethylenanlagen) eine weitere Ebene der dynamischen Koordinierung von MPC-Regelungen einzuführen, die eine dynamisch aufeinander abgestimmte Annäherung an das anlagenweite Optimum auch ohne Verwendung theoretischer Prozessmodelle gestattet. Beispiele hierfür sind DMCplus Composite (Aspen) und Profit Optimizer (Honeywell).

3. Verfahrenstechnische Prozessanlagen sind wie ein lebendiger Organismus ständigen Veränderungen unterworfen. Diese resultieren sowohl aus unbeabsichtigten Alterungsprozessen als auch aus aktiven Maßnahmen der Instandhaltung, Modifikation oder Erweiterung der Prozessanlagen im laufenden Betrieb. Demzufolge ist das statische und dynamische Verhalten einer Prozessanlage ständigen Änderungen unterworfen. Das macht die fortlaufende Pflege und Anpassung von MPC-Regelungen im laufenden Betrieb erforderlich. Eine wichtige Grundlage hierfür ist die Entwicklung von Werkzeugen für ein systematisches Control Performance Monitoring, ähnlich wie für das System der PID-Basisregelungen [Di04b]. Obwohl noch am Anfang der Entwicklung, gibt es auch hier erste kommerziell verfügbare Werkzeuge, Beispiele sind Aspen Watch (Aspen) und MD<sup>pro</sup> (Shell Global Solutions).

Die erfolgreiche Anwendung von MPC-Regelungen mit linearen Prozessmodellen ist an die Voraussetzung gebunden, dass die zu regelnden Prozesse in einer mehr oder weniger engen Umgebung eines festen Arbeitspunktes betrieben werden und keine gravierenden Nichtlinearitäten aufweisen. Überdies muss gewährleistet sein, dass sich das dynamische Verhalten im Laufe der Betriebsdauer nicht wesentlich ändert. Das ist bei vielen Raffinerie- und Petrochemieanlagen der Fall. Daraus – und überdies aus Wirtschaftlichkeitserwägungen – erklärt sich der hohe Anteil dieser Branchen an der Gesamtzahl der MPC-Einsatzfälle. Bei einer großen Zahl potentieller MPC-Anwendungen ist die Voraussetzung der Linearität im Arbeitsbereich des Prozesses jedoch nicht erfüllt. Daher gibt es seit einigen Jahren verstärkte Bemühungen, das MPC-Konzept auf nichtlineare und zeitveränderliche Prozesse zu erweitern. Dies stellt eine große theoretische und praktische Herausforderung dar, u.a. aus folgenden Gründen:

- Der Aufwand für die Entwicklung eines dynamischen Prozessmodells ist für nichtlineare Systeme im Allgemeinen wesentlich größer als im linearen Fall, das gilt sowohl für die theoretische Modellbildung als auch für den Weg der Identifikation empirischer Modelle.
- Im Gegensatz zu linearen Systemen gibt es für die Identifikation nichtlinearer dynamischer Systeme auf der Grundlage von Messdaten weder eine ausgereifte Theorie noch kommerziell verfügbare Software. Alle Schritte der Identifikation (Wahl der Modellform, Entwurf geeigneter Testsignale, Parameterschätzverfahren, Modellvalidierung) erweisen sich als recht kompliziert und sind an Expertenkenntnisse gebunden.
- Die Lösung beider Optimierungsaufgaben (dynamische Optimierung zur Auffindung der besten Steuergrößenfolge und statische Prozessoptimierung

zur Bestimmung optimaler Sollwerte) sind wesentlich schwieriger als im Fall linearer MPC-Regelungen, i.A. entsteht ein nichtlineares, nichtkonvexes Optimierungsproblem mit Nebenbedingungen, das im Echtzeitbetrieb gelöst werden muss.

- Die Untersuchung wichtiger Eigenschaften des geschlossenen Regelungssystems wie Stabilität und Robustheit gestalten sich viel komplizierter als im linearen Fall.

Obwohl die Entwicklung bei NMPC-Regelungen noch in den Kinderschuhen steckt, gibt es in jüngster erfolversprechende Entwicklungen sowohl auf theoretischem Gebiet [AZ00 und KC01] als auch erste kommerziell verfügbare NMPC-Tools wie z. Process Perfecter (Pavilion), NOVA NLC (PAS) und Aspen Apollo (Aspen Technology). Insbesondere im Zusammenhang mit der beschleunigten Entwicklung dynamischer Prozess-Simulatoren sind auf diesem Gebiet mittelfristig große Fortschritte zu erwarten.

## Literatur

- [AZ00] Allgöwer, F., Zheng, A.(Eds.): Nonlinear Model Predictive Control. Birkhäuser Verlag Basel 2000
- [CB04] Camacho, E.F., Bordons, C.: Model Predictive Control. Springer-Verlag London 2004
- [DP04] Dittmar, R., Pfeiffer, B.-M.: Modellbasierte prädiktive Regelung. Oldenbourg Wissenschaftsverlag München 2004-09-02
- [Di04] Dittmar, R.: Modellgestützte prädiktive Regelung. In: Früh, K.H., Maier, U. (Hrsg.): Handbuch der Prozessautomatisierung. Oldenbourg Industrie-verlag 2004, S. 138 –158
- [Di04b] Dittmar, R.: Control Performance Monitoring. In: Früh, K.H., Maier, U. (Hrsg.): Handbuch der Prozessautomatisierung. Oldenbourg Industrie-verlag 2004, S. 165 - 180
- [KC01] Kouvaritakis, B., Cannon, M. (Eds.): Nonlinear predictive control – theory and practice. IEE London 2001
- [Mac02] Maciejowski, J.: Predictive control with constraints. Prentice Hall 2002
- [QB03] Qin, J., Badgwell, T. A.: A Survey of Industrial Model Predictive Control Technology. Control Engineering Practice 11(2003) H. 7, S.733-764
- [Zhu01] Zhu, Y.: Multivariable system identification for process control. Pergamon Press 2001
- [Zhu04] Zhu, Y.: New developments in industrial MPC identification. 7<sup>th</sup> Symposium Advanced ADCHEM Symposium, Hongkong 2004, Preprints vol. 1, S. 23 – 30

# Neural Network Modelling and Predictive Control of An IC Engine

S. W. Wang, D. L. Yu, J. B. Gomm, G. F. Page and S. S. Douglas

*Control Systems Research Group,  
School of Engineering, Liverpool John Moores University,  
Byrom Street, Liverpool, L3 3AF, UK  
Email: j.b.gomm@livjm.ac.uk*

## Abstract

Automotive engine control is one of the most complex control problems for control system engineers and researchers. This paper presents an application of neural network modelling and MPC (Model-based Predictive Control) for an engine simulation. Three different training algorithms for RBF networks are used for modelling the intake manifold pressure and the crankshaft speed, and their modelling abilities are compared. Based on the obtained neural network models, a MPC strategy for controlling the crankshaft speed is realised by using a non-linear optimisation algorithm. An acceptable control result has been achieved.

Keywords: Engine control; Neural networks; Non-linear modelling; Model-based predictive control

## 1 Introduction

Automotive engine control is one of the most complex control problems for control system engineers and researchers. Due to the increasing requirements of governments and customers, car manufacturers always try to reduce substantially emissions and fuel consumption while maintaining the best engine performance. To satisfy these requirements, a variety of variables need to be controlled, such as engine speed, engine torque, spark ignition timing, fuel injection timing, air intake, air-fuel ratio and so on. These variables are complicatedly related to each other. Moreover, car engines have several different operating modes including start up, idle, running and braking. Engine dynamics are severely non-linear and multivariable because of these factors.

From an enormous amount of research about the modelling and control of automotive engines, it has been shown that neural networks not only provide a simple model structure, but also capture the inherent nonlinearities and the

dynamics of automotive engines with satisfactory accuracy [2,3,4,6]. Therefore, this method is an appropriate choice for model-based control strategies. From the control point of view, MPC (Model-based Predictive Control), as a recent developed practical approach, is a potential control strategy for future high performance car engines. In this paper, work focuses on predictive control based on neural network engine models.

The paper consists of the following. Section 2 introduces the downloaded engine simulation package. Section 3 discusses three neural network modelling methods of car engines, using the knowledge derived from Section 2. The modelling performance of the different methods are compared and analysed. Section 4 provides a neural network model based predictive control strategy for the engine crankshaft speed. Conclusions and remarks are reported in Section 5.

## 2 The engine simulation

Currently, a real engine is not available for experiments. Hence, an engine simulation has been downloaded from the personal website of Professor Elbert Hendricks [1]. This simulation engine is called a generic mean value engine and consists of three submodels that describe the fuel mass flow dynamics, the intake manifold filling dynamics and the crankshaft speed. The two latter submodels were used in this paper. All the variables in this section are defined in the notation.

### 2.1 The intake manifold filling dynamics

The intake manifold filling dynamics is analysed from the viewpoint of the air mass conservation inside the intake manifold. It includes two nonlinear differential equations about the manifold pressure and the manifold temperature.

The manifold pressure is mainly a function of the air mass flow past throttle plate, the air mass flow into the intake port, the EGR (Exhaust Gas Recirculation) mass flow, the EGR temperature and the manifold temperature. It is described as

$$\dot{p}_i = \frac{\kappa R}{V_i} \left( -\dot{m}_{ap} T_i + \dot{m}_{at} T_a + \dot{m}_{EGR} T_{EGR} \right) \quad (1)$$

The manifold temperature can be computed using the following differential equation

$$\dot{T}_i = \frac{RT_i}{p_i V_i} \left[ -\dot{m}_{ap} (\kappa - 1) T_i + \dot{m}_{at} (\kappa T_a - T_i) + \dot{m}_{EGR} (\kappa T_{EGR} - T_i) \right] \quad (2)$$

Here, the EGR mass flow was not considered and simply set to be zero. The air mass flow past throttle plate is related with the throttle position and the manifold pressure. The air mass flow into the intake port is a function of the crankshaft speed and the manifold pressure. Then the manifold pressure can be described by an input-output mapping function

$$\dot{p}_i = f(u, p_i, n, T_i) \quad (3)$$

## 2.2 The crankshaft speed

The crankshaft speed is derived based on the conservation of the rotational energy on the crankshaft.

$$\dot{n} = -\frac{1}{In} \left( P_f(p_i, n) + P_p(p_i, n) + P_b(n) \right) + \frac{1}{In} H_u \eta_i(p_i, n, \lambda) \dot{m}_f(t - \Delta\tau_d) \quad (4)$$

Both the friction power  $P_f$  and the pumping power  $P_p$  are related with the manifold pressure  $p_i$  and the crankshaft speed  $n$ . The load power  $P_b$  is a function of the crankshaft speed  $n$  only. The indicated efficiency  $\eta_i$  is a function of the manifold pressure  $p_i$ , the crankshaft speed  $n$  and the air/fuel ratio  $\lambda$ . The engine port fuel mass flow  $\dot{m}_f$  is described by the following equation

$$\dot{m}_f = \frac{\dot{m}_{ap}}{\lambda L_{th}} \quad (5)$$

Similarly, the crankshaft speed can be derived by another input-output mapping function.

$$\dot{n} = f(p_i, n) \quad (6)$$

## 3 Neural network modelling

### 3.1 Data collection

A set of Random Amplitude Signals (RAS) shown in Figure 1 was designed for the throttle position to obtain a representative set of input-output data. The range of this excitation signal was bounded between 20 and 40 degrees. The length of each amplitude was set to be 5 seconds which is 10 times the sampling time of the simulation system. As shown in Figure 2, three sets of data were collected for the neural network training and testing.

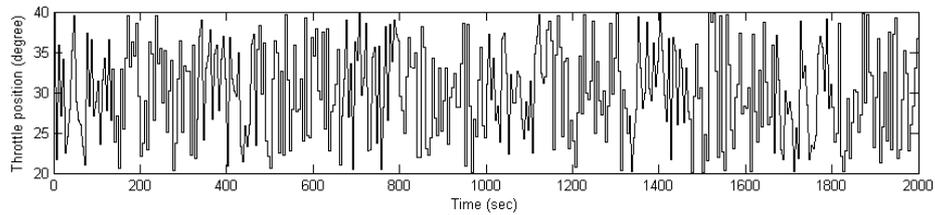


Figure 1: Random amplitude signals of the throttle position

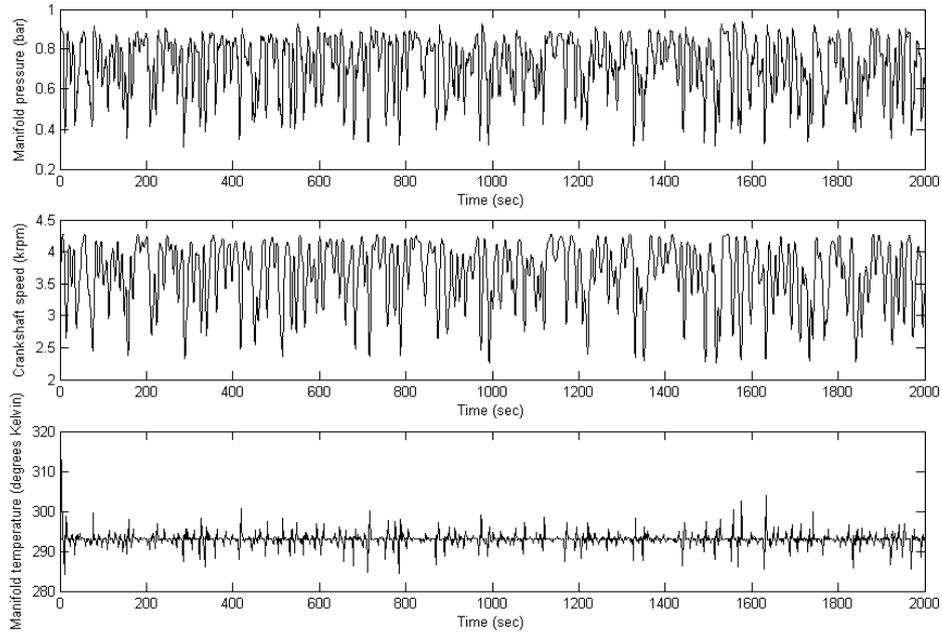


Figure 2: Three sets of collected data

### 3.2 Neural network structure

As shown in Figure 3, the radial basis function (RBF) neural network was chosen to construct a second-order engine model with four inputs and two outputs.

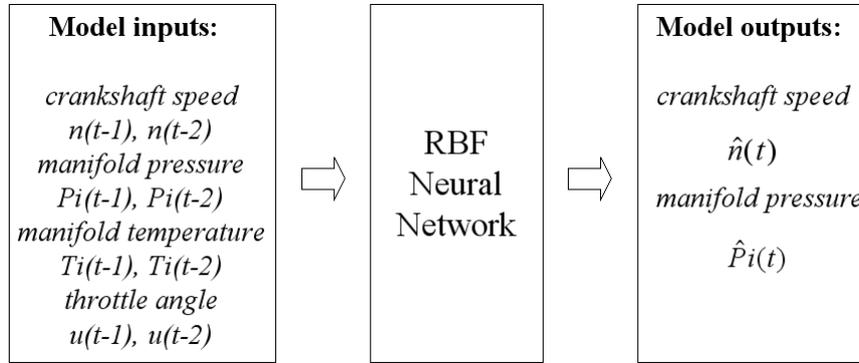


Figure 3: The structure of RBF neural network

The  $i$ th output of the neural network model at time  $t$  is

$$y_i(t) = \sum_{j=1}^{nh} \phi_j(t) w_{ji} \quad (7)$$

where  $\phi_j(t)$  is the  $j$ th activation function and  $w_{ji}$  is the weight. The Gaussian RBF, which has the form of the multi-dimensional bell-shaped curve, was used as the activation functions

$$\phi_j(t) = \exp\left(-\frac{\|x(t) - c_j(t)\|^2}{\sigma_j^2}\right) \quad (8)$$

where  $x(t)$  is RBF neural network inputs,  $c_j(t)$  is the  $j$ th centre and  $\sigma_j$  is the corresponding width.

### 3.3 The three training algorithms

In order to obtain the best modelling performance and analyse the modelling ability, three different training algorithms were used.

**Algorithm 1:** *BLS (Batch Least Squares) + K-means clustering*

This algorithm is widely used for off-line training. The centres are set by the K-means clustering method. The widths are computed by the p-nearest neighbours method. The batch least squares method is responsible for training the weights  $W$  by using equation (9).

$$W = (\Phi^T \Phi)^{-1} \Phi^T Y \quad (9)$$

where  $W$  is the matrix of weights,  $\Phi$  is the matrix of activation function outputs and  $Y$  is the matrix of training targets.

**Algorithm 2:** *RLS (Recursive Least Squares) + K-means clustering*

The recursive least squares method is used for on-line training. It helps to construct on-line system models which should be more accurate and flexible than the off-line models. The equations of this algorithm are summarized as follows [7].

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \quad (10)$$

$$\hat{w}(t) = \hat{w}(t-1) + L(t)[y(t) - \varphi^T(t)\hat{w}(t-1)] \quad (11)$$

$$P(t) = \frac{1}{\lambda(t)} \left[ P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \right] \quad (12)$$

$\lambda(t)$  is set to be one and the initial conditions are

$$P(0) = 10000 \times I_{nh \times nh} \quad \hat{w}(0) = 2.22 \times 10^{-16} \times U_{nh \times 2}$$

where  $I$  is unit matrix and  $U$  represents a matrix whose components are one.

At each sampling time, the parameters  $L(t)$ ,  $\hat{w}(t)$  and  $P(t)$  are updated orderly with the activation function  $\varphi(t)$ . Hence, in this algorithm, only the weights are adapted recursively, the centres are fixed after running the K-means clustering program for a certain set of training data.

**Algorithm 3:** *RLS (Recursive Least Squares) + recursive K-means clustering*

The difference between algorithm 3 and algorithm 2 lies in the adaptation of the centres. In algorithm 3, instead of using the fixed centres, the centres are adapted in each sampling period by the recursive K-means clustering method [5]. It helps to realise real on-line modelling especially when the off-line training data for the K-means clustering is not available. To adapt the centres recursively, two procedures should be followed:

(i) Compute distances  $a_i(t)$  between the sampled input data  $x(t)$  and each centre  $c_i(t-1)$ . Find a minimum distance and use  $k$  to represent its index.

$$c_i(t) = c_i(t-1) \quad (13)$$

$$c_k(t) = c_k(t-1) + \alpha_c(t)(v(t) - c_k(t-1)) \quad (14)$$

$$a_i(t) = \|x(t) - c_i(t-1)\| \quad (15)$$

$$k = \arg[\min\{a_i(t)\}] \quad \begin{matrix} 1 \leq i \leq nh \\ i \neq k \end{matrix} \quad (16)$$

(ii) Update the corresponding centre

The learning rate is

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + \text{int}[t / nh]^{1/2}} \quad (17)$$

where  $\text{int}[\ ]$  means the integer part of the argument.

### 3.4 Comparison of the modelling results

During the neural network modelling process, 1000 data samples (1~1000) were used for training and another 1000 data samples (1001~2000) were used for testing. The sum squared errors of the modelling outputs for different numbers of hidden neurons are shown in Table 1.

Table 1 Modelling errors of the three different training algorithms

Hidden neurons	SSE (n)			SSE (Pi)		
	BLS (fixed centres)	RLS (fixed centres)	RLS (recursive centres)	BLS (fixed centres)	RLS (fixed centres)	RLS (recursive centres)
2	6.64	6.67	100.02	10.34	10.37	13.62
5	4.73	5.13	8.93	4.91	9.88	11.23
10	3.71	4.44	4.25	2.71	4.33	2.83
15	2.26	4.33	3.77	2.67	3.86	2.86
18	2.11	4.06	3.91	2.55	2.97	2.69
20	1.99	4.07	3.84	2.59	3.06	2.66
25	1.98	3.79	3.96	2.30	2.82	2.67
30	2.19	3.81	3.47	2.32	2.80	2.60

As the number of hidden neurons increases, the modelling errors of the three methods have a decreasing intention. Comparing the three methods, algorithm 1 has the best modelling performance and there is not an obvious difference between algorithms 2 and 3. Further comparison is needed by using real engine data and considering much more disturbance. The number of hidden neurons should be in

the range between 15 and 25 because increasing the number further does not improve the modelling performance significantly.

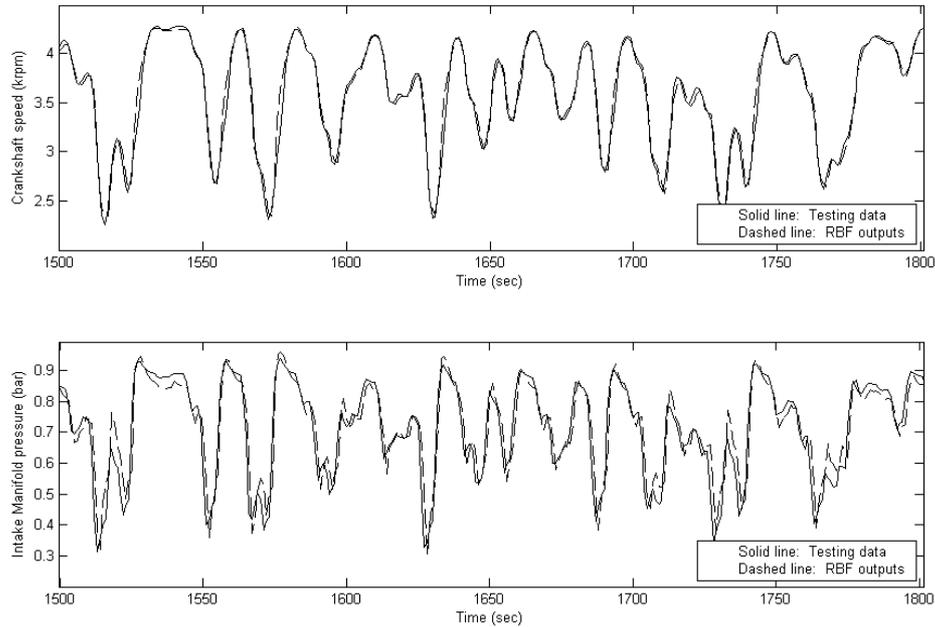


Figure 4: Modelling results for 18 hidden neurons (Data samples: 1500~1800)

Taking the model trained by algorithm 2 with 18 hidden neurons as an example, the modelling results in Figure 4 were obtained. It can be seen in the figure that for the crankshaft speed, the modelling curve can track the testing data very well. On the other hand, the modelling results of the manifold pressure are not as accurate due to its rapid changing dynamics.

#### 4 Model-based predictive control for the crankshaft speed

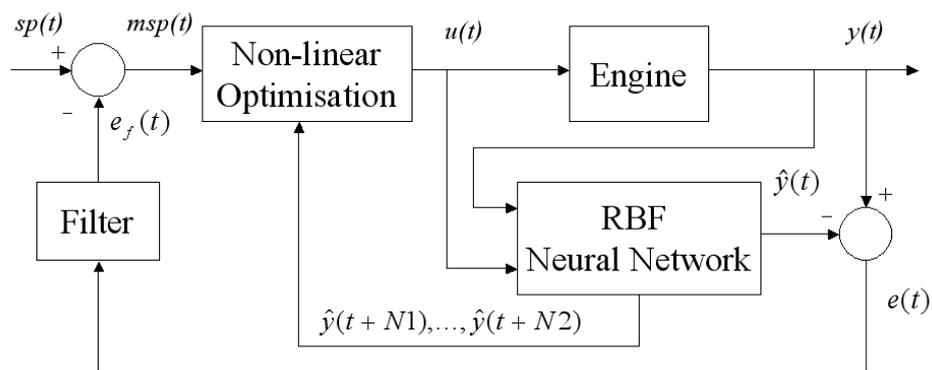


Figure 5: The neural network model-based predictive control

Based on one of the obtained neural network models in Section 3, a predictive control strategy was realised for the crankshaft speed control. As shown in Figure 5, the neural network is used to predict the engine output for  $N_2$  steps ahead. The

nonlinear optimiser minimizes the errors between the set point and the engine output by using the cost function,

$$J = \sum_{i=t+N1}^{t+N2} [r(i) - y_m(i)]^2 + \lambda \sum_{i=t}^{t+N_u} [u(i) - u(i-1)]^2 \quad (18)$$

$N1$ ,  $N2$  define the prediction horizon.  $\lambda$  is a control weighting factor which penalizes excessive movement of the control input, the throttle position  $u(t)$ .  $N_u$  is the control horizon.

In this paper, the training algorithm 2 mentioned in Section 3 was used for the second-order RBF neural network with four inputs and two outputs. The parameters of the nonlinear optimisation was chosen as  $N1=1$ ,  $N2=2$ ,  $\lambda=0.5$ ,  $N_u=0$ . The control results are shown in Figure 6.

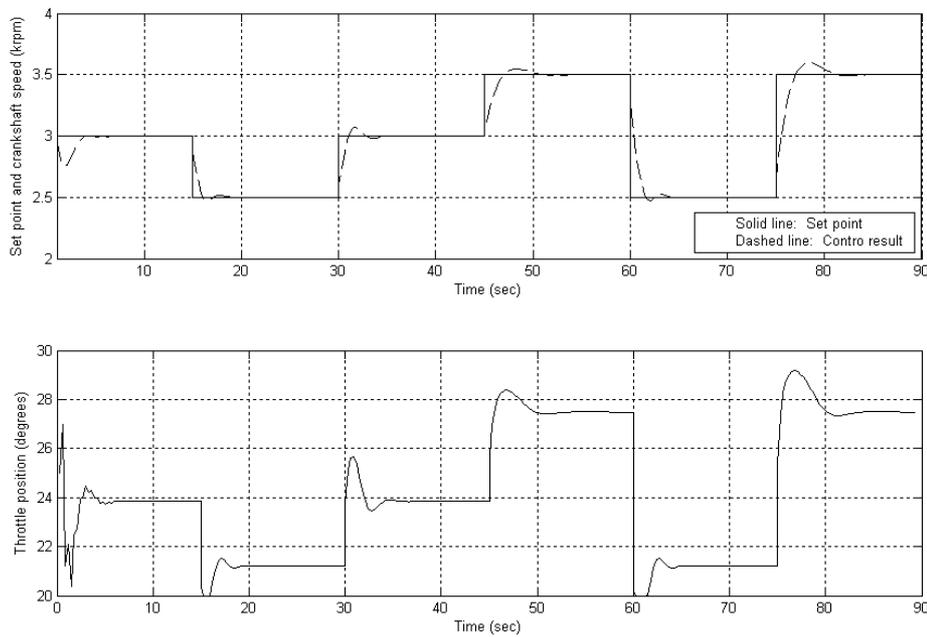


Figure 6: Crankshaft speed control results

## 5 Conclusions

This paper has shown an example of using RBF neural networks trained with three algorithms for modelling the engine dynamics, i.e. the intake manifold filling dynamics and the crankshaft speed, and the model-based predictive control for the crankshaft speed. Some conclusions can be obtained as follows:

- (1) The RBF neural network is suitable for modelling the engine speed, but improvement is needed for modelling the manifold pressure.
- (2) For this simulation engine, all the three training algorithms for the RBF neural networks can achieve good modelling performance. Further research

about their modelling abilities is needed based on real engine systems with more complex dynamics.

- (3) The MPC strategy can achieve good engine speed control results. It is a potential control method for controlling other engine variables such as the ignition angle and the fuel injection angle to improve current engine control strategies.

## Notation of the engine simulation

$p_i$ manifold pressure (bar)	$\kappa$ ratio of the specific heats = 1.4 for air
$R$ gas constant (here $287 \times 10^{-3}$ )	$V_i$ manifold + port passage volume ( $m^3$ )
$P_b$ load power (kW)	$H_u$ fuel lower heating value (kJ/kg)
$P_p$ pumping power (kW)	$\dot{m}_{at}$ air mass flow past throttle plate (kg/sec)
$P_f$ friction power (kW)	$T_a$ ambient temperature (degrees Kelvin)
$\dot{m}_{EGR}$ EGR mass flow (kg/sec)	$\dot{m}_{ap}$ air mass flow into intake port (kg/sec)
$u$ throttle position (degrees)	$T_{EGR}$ EGR temperature (degrees Kelvin)
$n$ crankshaft speed (krpm)	$I$ crank shaft load inertia ( $kg\ m^2$ )
$\eta_i$ indicated efficiency	$\dot{m}_f$ engine port fuel mass flow (kg/sec)
$\lambda$ air/fuel ratio	$T_i$ intake manifold temperature (degrees Kelvin)
$\Delta\tau_d$ injection torque delay time (sec)	$L_{th}$ stoichiometric air/fuel ratio (14.67)

## References

- [1] Hendricks, E., A Generic Mean Value Engine Model for Spark Ignition Engines. *41<sup>st</sup> Simulation Conference, SIMS 2000*, DTU, Lyngby, Denmark, Sept 18-19 (2000), also available at [www.iau.dtu.dk/~eh/](http://www.iau.dtu.dk/~eh/)
- [2] Yonghong Tan and Mehrdad Saif, Neural-networks-based nonlinear dynamic modelling for automotive engines. *Neurocomputing* 30, 129-142 (2000).
- [3] Isermann R. and Muller N., Modelling and Adaptive Control of Combustion Engines with Fast Neural Networks. *European Symposium on Intelligent*

*Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems*, Tenerife, Spain, December 13-15 (2001).

- [4] F Vinsonneau J.A., Shields D.N., King P.J. and Burnham K.J., Polynomial and neural network spark ignition engine intake manifold modelling. *Proc. 16<sup>th</sup> Int. Conf. on Systems Engineering, ICSE*, Vol 2, 718-723 (2003).
- [5] S. Chen, S.A. Billings and P. M. Grant, Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *Int. J. Control* 55(5), 1051-1070 (1992)
- [6] De Nicolao G., Scattolini R. and Siviero C., Modelling the volumetric efficiency of IC engines: parametric, non-parametric and neural techniques. *Control Eng. Practice* 4(10), 1405-1415 (1996).
- [7] Lennart Ljung, *System Identification—Theory for the User*, second edition, Prentice-Hall, 361-369 (1999)

# Anwendung von adaptiven, prädiktiven Reglern in der Anästhesie

O. Simanski \*, R. Kähler† Universität Rostock

## Zusammenfassung

Motiviert durch die Entwicklung immer kürzer wirkender Muskelrelaxantien gehen die Anästhesisten zunehmend von einer Bolusapplikation der Medikamente zu einer kontinuierlichen Applikation über, eine günstige Voraussetzung für die Automatisierung der Medikamentengabe. Nachfolgend werden verschiedene Ansätze für die Regelung der neuromuskulären Blockade diskutiert und die Ergebnisse eines adaptiven prädiktiven Reglers präsentiert.

## 1 Einleitung und Motivation

Während einer Operation nutzt der Anästhesist verschiedene Medikamente, um eine ausreichende Narkose sicherzustellen und die chirurgischen Manipulationen bei Aufrechterhaltung der Vitalfunktionen des Patienten zu kompensieren. Ihm stehen für diese Aufgabe Überwachungsmonitore zur Messung der Patientenparameter und gegebenenfalls Infusionspumpen für die Medikamentenapplikation zur Verfügung. Die Anästhesie stellt sich als ein Multi-Input - Multi-Output (MIMO) Problem dar. Um eine adäquate Narkose für den einzelnen Patienten zu erreichen, hat der Anästhesist eine Vielzahl von Sollwerten einzuhalten und vielfältige Einflussmöglichkeiten.

## 2 Regelungskonzept

Unter Berücksichtigung aller Teilbereiche der Anästhesie stellt sich dem Regelungstechniker die Aufgabe, eine Mehrgrößenregelung zu entwerfen. Voraussetzung für die Regelung ist die Festlegung und Messung der Regelgrößen. Während diese für die neuromuskuläre Blockade mit relativ geringem Aufwand bestimmbar ist, wird die Messung der Hypnosetiefe uneindeutiger. Die Interpretation der Veränderung der Hirnströme als Maß für die Hypnosetiefe ist noch nicht endgültig geklärt. Noch defizitär ist die Quantifizierung des Schmerzempfindens für die Ermittlung einer Regelgröße zur Regelung der Analgesie. Abbildung 1 zeigt das Mehrgrößensystem. Die neuromuskuläre Blockade wird elektromyographisch (EMG) durch die Messung von Summenaktionspotentialen nach einer Neurostimulation erfasst. Die neuromuskuläre Blockade kann zwischen 0%, keine Blockade und 100%, vollständige Blockade der Nerv-Muskel-Reizweiterleitung, variieren. In der Allgemeinchirurgie wird häufig eine neuromuskuläre Blockade von 90% angestrebt. Für die Erfassung der Hypnosetiefe wird in der beschriebenen Konfiguration ein BIS-Monitor eingesetzt. Dieser wertet die Hirnströme des Patienten aus und berechnet nach Bispectralanalyse (BIS) einen Index zwischen 0 und 100, wobei 100 einen wachen Patienten charakterisiert. Angestrebt wird bei allgemein chirurgischen Eingriffen ein BIS-Wert zwischen 40 und 60.

---

\*Institut für Automatisierungstechnik, Fakultät für Informatik und Elektrotechnik, D-18119 Rostock, olaf.simanski@technik.uni-rostock.de

†Abteilung für Mess- und Regelungstechnik, Fakultät für Maschinenbau, D-18059 Rostock, ralf.kähler@mbst.uni-rostock.de

Für die Analgesie suchen wir gegenwärtig aktiv nach einem Parameter, der die Reaktion des Patienten auf mögliche Schmerzen unter Narkose beschreiben kann.

Für Mehrgrößensysteme können unterschiedliche Regelungsstrategien Anwendung finden. Die

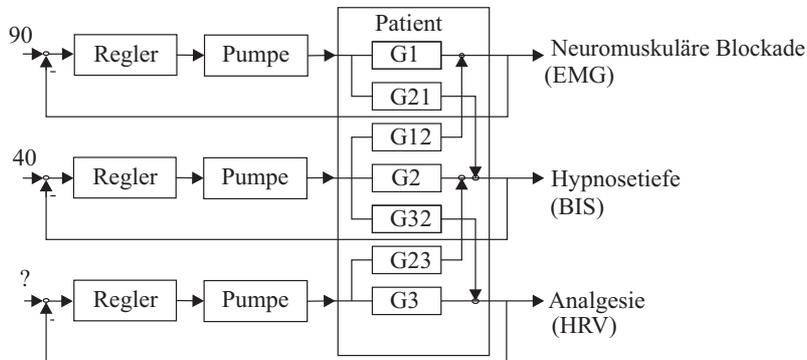


Abbildung 1: Mehrgrößenregelung in der Anästhesie

Querkopplungen, wie in Abbildung 1 dargestellt, sind beim Reglerentwurf zu berücksichtigen. Mehrgrößenregelungssysteme können je nach Art der Verkopplung und Regelungsstrategie unterteilt werden in *schwach gekoppelte Mehrgrößensysteme - separate Eingrößenregler*, *statisch entkoppelte Mehrgrößensysteme - dezentrale Regelung*, *entkoppelte Mehrgrößensysteme - Mehrgrößenregler mit Entkopplungsglied* und in die *direkte Mehrgrößenregelung*.

Bei *schwach gekoppelten Mehrgrößensystemen* sind die Kopplungen zwischen den Regelkreisen so geringfügig, dass sie beim Reglerentwurf vernachlässigt werden können. Die Mehrgrößenregelungsaufgabe wird durch den Entwurf mehrerer einfacher einschleifiger Regelkreise gelöst. Der Entwurf der Regler kann nacheinander ohne besondere Berücksichtigung der Mehrgrößenaufgabe nach den Entwurfsmethoden für Eingrößenregelungen erfolgen. Diese Vorgehensweise wird auch als *sequential loop closing* [Mac93] bezeichnet.

Sind die Querkopplungen nicht mehr vernachlässigbar, müssen sie beim Reglerentwurf berücksichtigt werden. Die Regelung kann dann ebenfalls durch zwei getrennte Teilregler realisiert werden. Der Begriff der *dezentralen Regelung* wird für diese Art einer Mehrgrößenregelung genutzt. Je nach den Anforderungen an die Regelung ist auch bei stärker gekoppelten Systemen eine statische Entkopplung, eine Entkopplung im eingeschwungenen Zustand, ausreichend.

Bei starken Querkopplungen in der Strecke sind oft entsprechende Querkopplungen im Regler notwendig. Ziel ist es, mit geeigneten Kopplungen im Regler, die Querkopplungen, die in der Strecke auftreten, zu kompensieren. Die beschriebene Entkopplung wurde häufig getrennt vom eigentlichen Regler vor die Regelstrecke geschaltet. Der Mehrgrößenregler besteht dann aus den entsprechenden Eingrößenreglern und einem Entkopplungsglied.

Die Entkopplung wird heute als ein nicht mehr notwendiger Zwischenschritt angesehen. Durch die Entkopplung wird dem System, ähnlich wie bei den Kompensationsreglern, ein für das System nicht in jedem Fall günstiges internes Verhalten aufgezwungen. Es führt zum Verlust an Robustheit und kann sogar interne Instabilität bewirken.

Heute werden in der Regel nur noch zwei Wege für die Mehrgrößenregelung beschrritten. Entweder sind die Querkopplungen so schwach, dass sie vernachlässigt bzw. wie bei der dezentralen Regelung im Entwurf der Eingrößenregler berücksichtigt werden können, oder sie werden in einen direkten Mehrgrößenreglerentwurf eingearbeitet.

Im gegenwärtigen Entwicklungsstand wurde die Anästhesieregelung mittels zweier unabhängig voneinander arbeitender Regler realisiert. Die auftretenden Querkopplungen werden als Ausgangs-

störungen der jeweiligen Teilregelkreise aufgefasst. Somit war ein getrennter Reglerentwurf, sowohl für die Regelung der neuromuskulären Blockade als auch für die Hypnosetiefe möglich. Nachfolgend wird die Realisierung für den Blockadetieferegler detaillierter ausgeführt.

### 3 Regelung der neuromuskulären Blockade

Voraussetzung für eine praktische Realisierung einer Regelung ist die Auswahl der entsprechenden Regel- und Stellgrößen. Regelgröße für die neuromuskuläre Blockade ist die mittels Elektromyographie (EMG) ermittelte Relaxation. Als Stellgröße berechnet der Regler die Infusionsrate in *ml/h* an Muskelrelaxans Mivacurium, die dem Patienten über die Stelleinrichtung Spritzenpumpe injiziert wird. Aus Sicht der Regelung stellt der Patient mit seiner neuromuskulären Blockade die Regelstrecke dar. Die gesamte Regelung der neuromuskulären Blockade wird in zwei Phasen unterteilt. Es können folgende Zusatzanforderungen an die Regelung gestellt werden.

- In der Phase eins soll der angestrebte Sollwert von einer 90%-igen neuromuskulären Blockade schnellstmöglich erreicht werden.
- Während der zweiten Phase steht die Ausregelung auftretender Störungen und somit die Gewährleistung eines in den von den Ärzten vorgegebenen Toleranzgrenzen von  $\pm 3\%$  verbleibenden neuromuskulären Blocks im Vordergrund.

Um in der ersten Phase möglichst schnell den gewünschten Blockadegrad von 90% zu erreichen, beginnt der Regler mit einer Bolusapplikation. Da es sich bei der Relaxansapplikation-Messeffekt-Beziehung um einen nichtlinearen Prozess handelt, wurde als einfachster nichtlinearer Regler ein 2-Punkt-Regler eingesetzt. Für die zweite Phase der Regelung bestanden für die Reglerauswahl folgende Optionen

- Auswahl eines nichtlinearen Reglers für den als nichtlinear charakterisierten Prozess "Patient" oder
- Durchführung einer Linearisierung des Prozesses um den Arbeitspunkt und der damit möglichen Wahl eines für diesen Arbeitspunkt gültigen Reglers.

Für den zu regelnden Prozess können folgende für die Auswahl des Reglers wichtigen Annahmen getroffen werden:

- Bei der Reaktion der Patienten auf die Gabe des Muskelrelaxans existieren große individuelle Unterschiede.
- Die messbare Reaktion auf die Gabe des Muskelrelaxans ist vor Beginn der Regelung nicht eindeutig bekannt und kann sich intraoperativ verändern.
- Vielfältige intraoperative Störungen durch im Operationssaal genutzte medizintechnische Geräte wie z.B. den Thermocauter, oder durch operative Besonderheiten wie z. B. einen massiven Blutverlust können die Messungen beeinflussen.

Aus diesen Eigenschaften können zusätzliche Anforderungen an die Regelung abgeleitet werden:

- Die Regelung muss an allen Patienten funktionieren und sich dazu an jeden einzelnen Patienten anpassen. Die Patientenanpassung kann zu Beginn der Regelungsphase geschehen, sie muss sich aber während der Regelung wiederholen.

- Die unbekannt oder schwer zu quantifizierenden Störeinflüsse wie Blutverlust, temperaturabhängige Veränderung der Patientenreaktion, Störungen in der Messwerterfassung dürfen nur sehr geringen Einfluss auf die Regelung haben. Die Totzeit des Prozesses ist nur näherungsweise bekannt und kann sich intraoperativ verändern. Die eingesetzte Regelung muss robust gegenüber Unsicherheiten in den Parametern und der Totzeit sein.

Für den Entwurf von Regelungen mit initialen Modellunsicherheiten und zeitvarianter Dynamik lassen sich zwei Herangehensweisen charakterisieren. Einerseits kann Ziel des Entwurfes ein fester aber robuster Regler sein, der die Stabilität der Regelung für alle möglichen Unsicherheiten und Dynamikveränderungen sicherstellt. Folge der Robustheit ist oft ein nicht optimal angepasstes Regelungsverhalten. Andererseits können adaptive Algorithmen, die den Regler mit Blick auf ein verbessertes Regelungsverhalten selber einstellen oder nachstellen, implementiert werden [Cla96]. Grundlage der Adaption bildet oftmals ein Modell, dessen Parameter identifiziert werden. Beim Entwurf von adaptiven Reglern lassen sich nach *Clarke* zwei Hauptrichtungen unterscheiden [Cla96]:

- Für den Fall, dass Systeme geregelt werden sollen, die eine unbekannt aber konstante Dynamik haben, werden einmalige Einstellungen vorgenommen. Für den Fall, dass der einzustellende Regler der als „Industriestandard“ bezeichnete PID-Regler ist, wird in diesem Zusammenhang der Begriff *autotuning* verwendet. Ist der einzustellende Regler hingegen komplexer, z.B. wenn die Strecke eine nicht zu vernachlässigende Totzeit hat und ein vollständiges Prozessmodell identifiziert werden muss, um mittels analytischem Entwurf den Regler einzustellen, spricht man vom *self-tuning* der Koeffizienten eines festen Regelungsgesetzes.
- Bei *zeitvarianten Systemen* hingegen wird das Modell über den gesamten Zeitraum aktualisiert und die Koeffizienten des Regelungsalgorithmus werden dann automatisch eingestellt. Mit dieser *adaptiven* Methode werden die Veränderungen im Systemverhalten erfasst und der Regler ist in Abhängigkeit von der Modellgüte stets optimal angepasst, so dass ein günstiges Regelungsverhalten zu erwarten ist. Limitiert wird dieses Verfahren oft durch die Modellgüte.

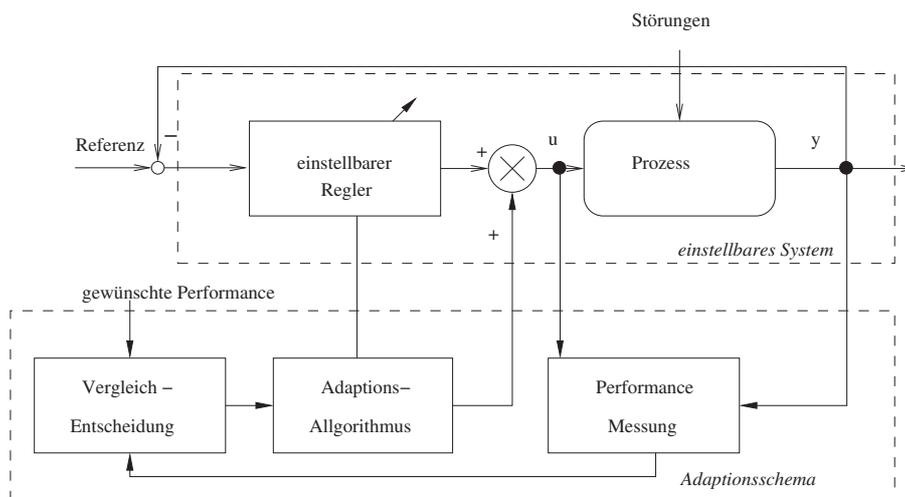


Abbildung 2: Basiskonfiguration für ein adaptives Regelungssystem [LLM98]

Abbildung 2 zeigt einen generellen Aufbau eines adaptiven Regelungssystems. Die verschiedenen Adaptionstechniken unterscheiden sich durch die Form, in der sie Informationen über den Prozess

in Echtzeit sammeln und den Regler so einstellen, dass dieser eine gewünschte Leistungsfähigkeit des Regelkreises sichert. Da die Parameter für den Regler von den Messungen der Systemvariablen durch den Adaptionskreis abhängig sind, ist ein adaptives Regelungssystem nichtlinear. Aus diesem Grund kann die Regelung selbst bei linearer Strecke und linearem Regler als nichtlineare Regelung mit unvollständigen Informationen aufgefasst werden [LLM98]. Unbekannte und nicht messbare Variationen in den Prozessparametern verringern im Allgemeinen die Leistungsfähigkeit des Regelungssystems. Unabhängig von der Wirkung der Störungen auf die zu regelnde Größe kann angenommen werden, dass Variationen in den Prozessparametern von Parameterstörungen herrühren. Die Störungen, die auf ein Regelungssystem Einfluss haben, können klassifiziert werden

- als Störung, die auf die zu regelnde Variable wirkt, oder
- als Parameterstörung, die sich auf die Leistungsfähigkeit des Regelungssystems auswirkt.

Aus der detaillierten Aufgabenstellung geht für den Entwurf der beiden Teilregler hervor, dass die Narkosetiefe relativ unabhängig von der neuromuskulären Blockade geregelt werden kann. Für die Regelung der neuromuskulären Blockade bietet sich zunächst ein *Regler mit 2-Freiheitsgraden* an, da somit die Führungs- und Störgrößen separat gewichtet werden können. Abbildung 3 zeigt den Grundgedanken dieses Reglers mit RST-Struktur. Mit Hilfe von  $T_1$  und  $S_1$  können relativ un-

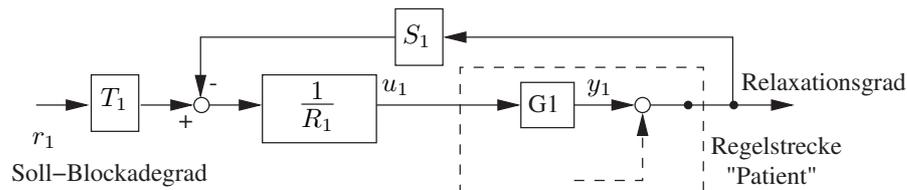


Abbildung 3: Regelung mit 2 Freiheitsgraden

abhängige Übertragungsfunktionen für Führung  $G_r(q^{-1})$  und Störung  $G_z(q^{-1})$  aufgestellt werden. Für den Regler ergibt sich dann die Form

$$R_1(q^{-1})u(k) = T_1(q^{-1})r(k) - S_1(q^{-1})y(k), \quad (1)$$

wobei der Index 1 nachfolgend nicht mehr mitgeführt wird.

Mit der Annahme, dass der Blockadeverlauf in der Form

$$A(q^{-1})y(k) = B(q^{-1})u(k) \quad (2)$$

beschrieben werden kann, wird die Führungsübertragungsfunktion  $G_r(q^{-1})$  mit

$$G_r(q^{-1}) = \frac{Q(q^{-1})}{P(q^{-1})} = \frac{B(q^{-1})T(q^{-1})}{A(q^{-1})R(q^{-1}) + B(q^{-1})S(q^{-1})} \quad (3)$$

eingeführt. Ausgehend von Gleichung (3) ergibt sich die diophantische Gleichung

$$[A(q^{-1})R(q^{-1}) + B(q^{-1})S(q^{-1})]Q(q^{-1}) = P(q^{-1})B(q^{-1})T(q^{-1}). \quad (4)$$

Die Eigenschaften der Führungsübertragung können durch die Wahl der Polynome  $P(q^{-1})$  und  $Q(q^{-1})$  vorgegeben werden. Für den Entwurf können die Prozessnullstellen nur durch kürzen verändert werden. Da die Kürzung instabiler und schwach stabiler Nullstellen, an der Stabilitätsgrenze liegender Nullstellen, zu Stabilitätsproblemen führt, werden diese ins  $Q$ -Polynom übernommen. Das  $B$ -Polynom kann somit in einen Teil mit stabilen kürzbaren Nullstellen  $B(q^{-1})^+$

und in einen Teil  $B(q^{-1})^{-}$  mit den zu übernehmenden Nullstellen zerlegt werden. Mit der Annahme von

$$Q(q^{-1}) = B(q^{-1})^{-}Q'(q^{-1})$$

ergibt sich dann

$$\begin{aligned} B(q^{-1})^{+}B(q^{-1})^{-}T(q^{-1})P(q^{-1}) = \\ B(q^{-1})^{-}Q'(q^{-1})[A(q^{-1})R(q^{-1}) + B(q^{-1})^{+}B(q^{-1})^{-}S(q^{-1})] \end{aligned} \quad (5)$$

mit

$$R(q^{-1}) = B(q^{-1})^{+}R'(q^{-1})$$

zu

$$P(q^{-1})T(q^{-1}) = Q'(q^{-1})[A(q^{-1})R'(q^{-1}) + B^{-}(q^{-1})S(q^{-1})]. \quad (6)$$

$T$  berechnet sich somit aus  $Q'(q^{-1})$  und dem Beobachterpolynom  $L(q^{-1})$

$$T(q^{-1}) = Q'(q^{-1})L(q^{-1}).$$

Für die Störübertragungsfunktion  $G_z(q^{-1})$  ergibt sich bei Einsatz des Reglers

$$G_z(q^{-1}) = \frac{B'(q^{-1})R(q^{-1})}{P(q^{-1})L(q^{-1})}. \quad (7)$$

Über das Beobachterpolynom  $L(q^{-1})$  kann unabhängig von einer festen Führungsübertragungsfunktion  $G_r(q^{-1})$  Einfluss auf die Störgrößenausregelung genommen werden.

Der *Regler mit 2-Freiheitsgraden*, der nach dem Prinzip der Polvorgabe arbeitet, erfüllt einen Großteil der oben genannten Eigenschaften. Er ist in der Lage, mit beliebigen Nullstellen zu arbeiten, kann adaptiv ausgelegt werden und in Erweiterungen auch bei Totzeitsystemen eingesetzt werden. Der größte Vorteil liegt in der einfachen Bestimmung der getrennten Stör- und Führungsübertragungsfunktion.

Bei der Regelung am Patienten sind nicht immer minimalphasige Modelle zu ermitteln. Sie werden bei einem Regler mit Polvorgabe auch nicht vorausgesetzt. Außerdem kann die Totzeit und die Patientenreaktion, -dynamik sich intraoperativ verändern. *Prädiktive Regler* haben den Vorteil, dass man bei ihnen leicht Begrenzungen implementieren kann. Sie bewerten die Führung und Störung ebenfalls unterschiedlich, auch wenn dies im Entwurf nicht so einfach zu beeinflussen ist [Sim02].

Der für die Regelung der neuromuskulären Blockade ausgewählte *Generalized Predictive Controller* erfüllt die genannten Anforderungen hinsichtlich der Robustheit gegenüber überparametrisierten, oder nichtminimalphasigen Modellen mit variabler Totzeit und kann mit einem, den Prozess hinreichend genau beschreibenden Modell arbeiten. Er wurde von *Clarke et al.* 1987 vorgestellt und gehört zu den *long-range* prädiktiven Reglern [CMT87]. Der GPC kann in einen Polvorgabe-Regler mit RST-Struktur umgerechnet werden, wie nachfolgend gezeigt wird.

Für den Einsatz des GPC wird angenommen, dass der zu regelnde Prozess nichtlinear, aber um den Arbeitspunkt durch ein lineares Modell angenähert werden kann. Somit wird das folgende lokal linearisierte Modell zugrunde gelegt:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + x(t), \quad (8)$$

mit

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \\ B(q^{-1}) &= q^{-d}(b_0 + b_1q^{-1} + \dots + b_{n_b}q^{-n_b}), \end{aligned}$$

wobei  $A$  und  $B$  Polynome im Verschiebeoperator  $q^{-1}$  sind und  $t$  die diskrete Zeit charakterisiert. Die Stellgröße wird durch  $u(t)$  und die Regelgröße durch  $y(t)$  beschrieben. Die geschätzte Totzeit symbolisiert  $d$ . Der Term  $x(t)$  charakterisiert die Störungen. Es wird ein Verhalten mit gleitender Mittelwertbildung angenommen, z.B.:

$$x(t) = C(q^{-1}) \frac{\xi_u(t)}{\Delta} \quad (9)$$

wobei gilt:

$$\begin{aligned} C(q^{-1}) &= c_0 + c_1 q^{-1} + \dots + c_{n_c} (q^{-n_c}) \\ \xi_u(t) &= \text{unkorrelierte Zufallsfolge} \\ \Delta &= 1 - q^{-1}. \end{aligned}$$

Wird nun (9) in (8) eingesetzt, so ergibt sich das Controlled Auto-Regressive and Integrated Moving-Average-Modell, CARIMA-Modell [CMT87] zu

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1}) \frac{\xi_u(t)}{\Delta}. \quad (10)$$

Das mit der Gleichung (10) beschriebene CARIMA-Modell bildet die Grundlage für die Ableitung des GPC - Regelalgorithmus.

Bevor der von *Clarke* [CMT87] entwickelte Algorithmus näher untersucht werden soll, zeigt Abbildung 4 die Idee, die den long-range prädiktiven Reglern zugrunde liegt [ML98].

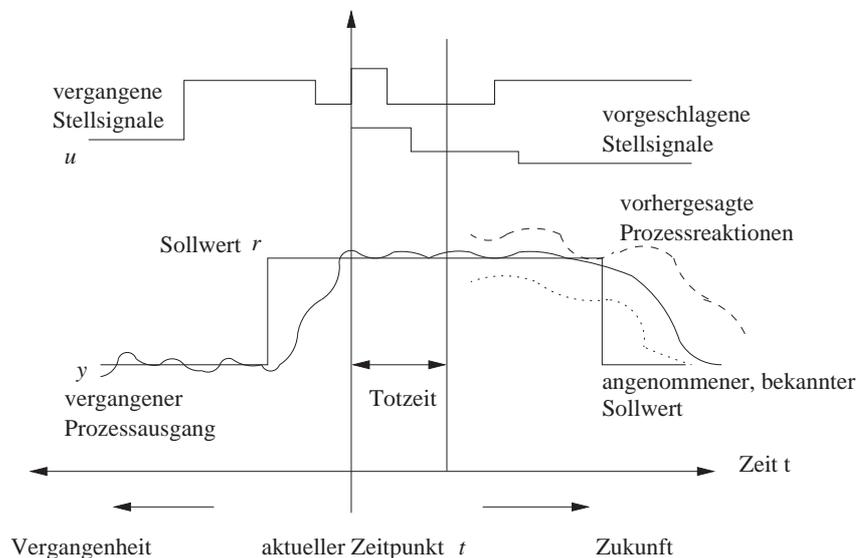


Abbildung 4: Prinzip der prädiktiven Regelung

- Zu jedem Zeitpunkt  $t$  wird mit Hilfe eines mathematischen Modells, das die Dynamik des Prozesses beschreibt, eine Vorhersage über einen langen Zeitbereich (long-range horizon) gemacht.
- Als Ergebnis dieser Vorhersagen werden verschiedene Stellgrößenverläufe generiert. Es wird aber nur die in Bezug auf den angestrebten Sollwert beste Strategie ausgewählt.

- Es wird dasjenige Stellsignal zur aktuellen Zeit  $t$  ausgegeben, welches der ausgewählten Strategie am besten entspricht. Der gesamte Vorgang wird in jedem Abtastschritt wiederholt.

Für die Berechnung der Stellgröße nutzt der Regler die Optimierung des folgenden Kriteriums in der Notation von *Clarke* [CMT87]:

$$J(N_1, N_2, NU, \lambda) = E[(Q_1 + Q_2)] \quad (11)$$

mit

$$Q_1 = \sum_{j=N_1}^{N_2} [\hat{y}(t+j) - r(t+j)]^2$$

$$Q_2 = \sum_{j=1}^{NU} [\lambda(j)(\Delta u(t+j-1))]^2$$

Wobei  $N_1$  den minimalen Vorhersagehorizont,  $N_2$  den maximalen Vorhersagehorizont,  $NU$  den Stellhorizont und  $\lambda(j)$  die Wichtung des Stellsignals darstellen.  $r(t+j)$  gibt die zukünftigen Sollwerte an.

Für die Entwicklung des Verfahrens wurde zunächst angenommen, dass  $C(q^{-1}) = 1$  gilt. Alternativ dazu kann  $C(q^{-1})$  in  $A(q^{-1})$  und  $B(q^{-1})$  näherungsweise eingebracht sein [CMT87]. *Clarke et al.* beschreiben in [CMT87] die Eigenschaften des  $C(q^{-1})$ -Polynoms.

Im GPC-Algorithmus ist die rekursive Lösung der Diophantischen Gleichung

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-1}F_j(q^{-1}), \quad (12)$$

in der  $E_j(q^{-1})$  und  $F_j(q^{-1})$  intern verwendete Polynome sind, für die Vorhersage über den gesamten Bereich  $j$  implementiert.

Die Minimierung der Funktion (11) ergibt für das Inkrement des Stellsignals:

$$\Delta u(t) = \mathbf{M}(r - f) \quad (13)$$

bzw.

$$\Delta u(t) = \sum_{i=N_1}^{N_2} m_i [\omega(t+i) - f(t+i)]. \quad (14)$$

Hier berechnet sich  $f$  aus

$$f(t+j) = \mathbf{G}' \Delta u^f(t-1) - F_j y^f(t). \quad (15)$$

$u^f$  und  $y^f$  ergeben sich aus der Filterung von  $u$  und  $y$  mit  $\frac{1}{T(q^{-1})}$ .

Dabei möge für  $T(q^{-1})$

$$T(q^{-1}) = 1 + t_1 q^{-1} + \dots + t_{n_t} q^{-n_t}$$

gelten.  $T(q^{-1})$  wird als Beobachterpolynom bezeichnet, in der aktuellen Implementation gilt  $T(q^{-1}) = 1$ . Der Vektor  $\mathbf{G}'(q^{-1})$  setzt sich unter der Annahme, dass  $d$  die Totzeit charakterisiert nach [CB00] folgendermaßen zusammen

$$\mathbf{G}'(q^{-1}) = \begin{pmatrix} (G_{d+1}(q^{-1}) - g_0)q \\ (G_{d+2}(q^{-1}) - g_0 - g_1 q^{-1})q^2 \\ \vdots \\ (G_{N_2}(q^{-1}) - g_0 - g_1 q^{-1} - \dots - g_{NU-1} q^{-(NU-1)})q^{NU} \end{pmatrix}. \quad (16)$$

Die ersten  $j$  Terme in jedem  $G'_i(q^{-1})$  entsprechen den Parametern der Sprungantworten, deshalb gilt  $g_{ij} = g_j$  für  $j = 0, 1, 2 \dots < i$ , wenn  $i = 0 \dots n_b$ . Allgemeiner gilt für die einzelnen Zeilen von  $G'(q^{-1})$

$$G'_i(q^{-1}) = g_{j0} + g_{j1}q^{-1} + \dots + g_{j(i-1)}q^{-i+1}.$$

$\mathbf{M}$  berechnet sich aus der ersten Reihe der Matrix  $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T$ , wobei  $\mathbf{G}$  die Dynamik-Matrix (Sprungantworten) nach *Clarke* ist [CMT87]

$$\mathbf{G} = \begin{pmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \dots & g_0 \\ \vdots & \vdots & \dots & \vdots \\ g_{N_2-1} & g_{N_2-2} & \dots & g_{N_2-NU} \end{pmatrix}. \quad (17)$$

Die Dimension der Matrix  $\mathbf{G}$  ist  $N_2 \times NU$ . Allgemein gilt

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f} \quad (18)$$

für die vorhergesagte Systemantwort.

Der GPC kann in einen Regler mit 2-Freiheitsgraden umgerechnet werden. *Camacho* und *Brodons* [CB00] berechnen die Polynome  $R(q^{-1})$  und  $S(q^{-1})$  entsprechend mit

$$R(q^{-1}) = \frac{T(q^{-1}) + q^{-1} \sum_{i=N_1}^{N_2} m_i I_i}{\sum_{i=N_1}^{N_2} m_i} \quad (19)$$

und

$$S(q^{-1}) = \frac{\sum_{i=N_1}^{N_2} m_i F_i}{\sum_{i=N_1}^{N_2} m_i}, \quad (20)$$

mit  $m_i$  als Elemente von  $\mathbf{M}$  und  $I_i$  als erster Zeile von  $G'(q^{-1})$ . Abbildung 5 zeigt die Struktur

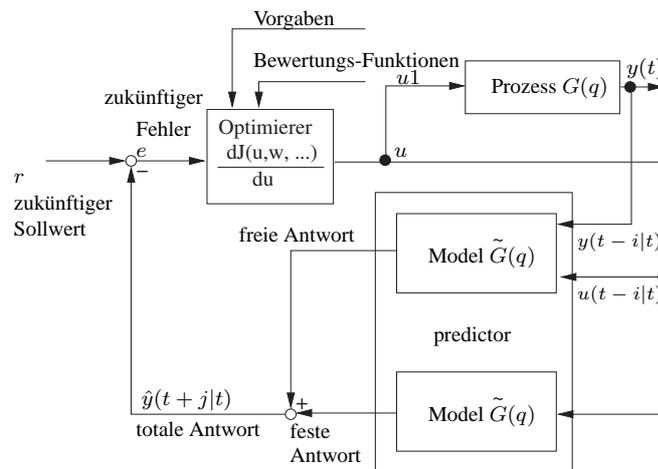


Abbildung 5: Struktur des GPC's

des Generalized Predictive Controllers. Der GPC generiert in jedem Abtastschritt ein Set von

möglichen zukünftigen Stellsignalen, aber nur das erste Element der besten Stellsignalsequenz wird als Eingangssignal für das System verwendet. Die Vorhersage des Systemausganges  $\hat{y}$  setzt sich aus zwei verschiedenen Komponenten zusammen.

Die sogenannte “freie Antwort” repräsentiert das vorhergesagte Verhalten des Systemausganges  $\hat{y}(t + j | t)$  im Bereich von  $t + 1$  bis  $t + N_2$  basierend auf den bereits vorhandenen “alten” Ausgängen  $y(t - i | t)$  und Eingängen  $u(t - i | t)$ . Die “feste Antwort” spiegelt den mittels  $G$  erzeugten Systemausgang  $\hat{y}$  in (18) wider. Für lineare Systeme können beide Komponenten additiv zur “totalen” Antwort zusammengefasst werden [KLL01].

### Der adaptive Generalized Predictive Controller (aGPC)

Als adaptiver Generalized Predictive Controller (aGPC) wird eine Erweiterung des GPC um eine rekursive online-Modellbildung angesehen.

Da der Gesamtprozess der neuromuskulären Blockade von der Medikamentenapplikation bis zur Messung des Blockadegrades nichtlinear ist, wurde das Modell um den gewünschten Blockadegrad von 90% linearisiert. Als Adaptionalgorithmus für den GPC wurde ein lineares ARX-Modell 3. Ordnung eingesetzt. Das Modell wird nach der rekursiven “Methode der kleinsten Quadrate”<sup>1</sup> in jedem Abtastschritt angepasst, so dass ein adaptiver Generalized Predictive Controller im Regelkreis arbeitet. Die Parameter des aGPC wurden wie folgt eingestellt, dabei blieb  $\lambda$  als Wichtungsfaktor für die Stellgrößenänderung konstant:

$$N_1 = 1 \quad N_2 = 36 \quad NU = 6 \quad \lambda(j) = 26$$

Abbildung 6 zeigt das Gesamtkonzept für die Regelung der neuromuskulären Blockade. Alle 12 Sekunden wird der Grad der neuromuskulären Blockade gemessen und die Infusionsmenge an die Infusionspumpe übertragen. In der Startphase appliziert ein modifizierter 2-Punkt-Regler einen

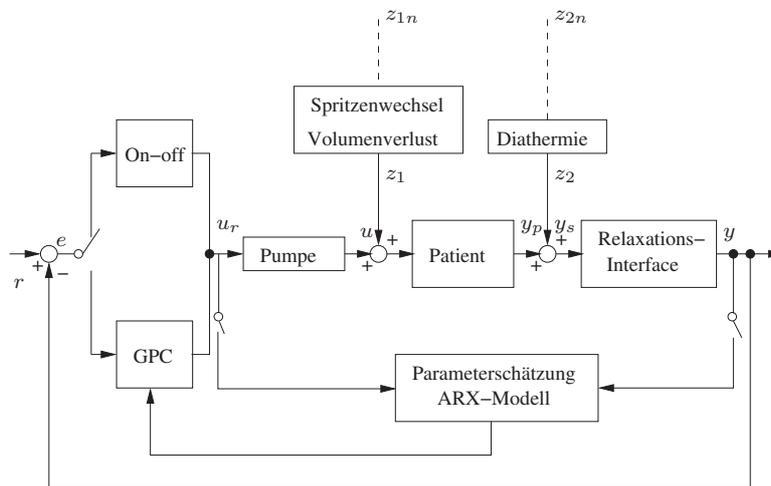


Abbildung 6: Struktur der Regelung der neuromuskulären Blockade

gewichtsabhängigen Initialbolus. Sobald bei eingeschaltetem 2-Punkt-Regler der Arbeitsbereich von 88% – 100% NMB erreicht ist, nimmt die mit Nullwerten vorinitialisierte rekursive ARX-Modellbildung ihre Arbeit auf. Nachdem der 2-Punkt-Regler das System mit zwei positiven Stellsignalen von mindestens 2 Minuten Dauer angeregt hat, wird die Regelung dem adaptiven Generalized Predictive Controller übergeben. Mit zunehmender Regelungszeit wird der Vergessensfaktor

<sup>1</sup> dadurch recursive ARX (RARX)

$\lambda$  der online-Modellbildung angepasst.

Der 2-Punkt-Regler wurde für die Startphase der Medikamentenapplikation gewählt, da er durch seine typische Arbeitsbewegung einen Kompromiss zwischen einfacher Regelung und Systemanregung darstellt. Für die Modellbildung begünstigt der 2-Punkt-Regler durch seine ausgeprägten Stellsignaländerungen eine schnelle Parameterkonvergenz. Während in der Modellbildungsphase ausgeprägte Stellsignaländerungen mit entsprechenden Systemreaktionen vorteilhaft sind, ist in der anschließenden Regelungsphase mit dem aGPC ein Kompromiss zwischen einem ruhigen Regelungsverlauf und hinreichender Systemanregung für die fortlaufende Adaption zu finden. Mit Hilfe der Parameter  $NU$  und  $\lambda$  des aGPC erfolgte die für die Stabilität notwendige Stellsignalbedämpfung.

Dem Regelungsalgorithmus ist eine Plausibilitätskontrolle überlagert, die fehlerhafte Messwerte und Artefakte herausfiltert. Arbeitet der aGPC nicht zuverlässig, kann jederzeit auf den 2-Punkt-Regler umgestellt werden.

## 4 Zusammenfassung

Der adaptive Generalized Predictive Controller aGPC hat sich in der Simulation und klinischen Anwendung als günstig erwiesen. Er ist in der Lage nichtminimalphasige Strecken mit variabler Totzeit und möglichen Parametervariationen zu regeln. Somit lässt er sich auch mit „einfachen“ Online-Identifikationsverfahren kombinieren.

Abbildung 7 zeigt eine intraoperative Regelung der neuromuskulären Blockade und der Hypnosetiefe. Die Regelung der Hypnosetiefe erfolgt aufgrund der unsicheren Modellierung des hoch-

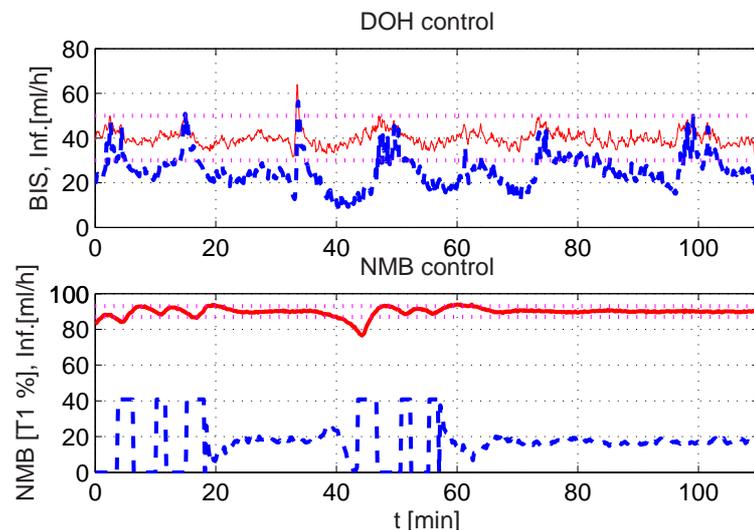


Abbildung 7: Regelung der neuromuskulären Blockade und der Hypnosetiefe während eines chirurgischen Eingriffs

gradig nichtlinearen Prozesses unter Verwendung eines Fuzzy-Reglers. Die Regelung wurde an mehreren hundert Patienten erfolgreich eingesetzt und steht vor der nächsten Ausbaustufe, der Integration einer Bewertung der Analgesierung des Patienten.

## Literatur

- [CB00] E. F. Camacho and C. Borndos. *Model Predictive Control*. Advanced Textbooks in Control and Dignal Processing. Springer, London Berlin Heidelberg New York, 2. edition, 2000.
- [Cla96] D. W. Clarke. Self-tuning control. In W.S. Levine, editor, *The Contol Handbook*, chapter 53, pages 827–846. CRC Press, Inc., Boca Raton, Florida, 1996.
- [CMT87] D.W. Clarke, C. Mothadi, and P.S. Tuffs. Generalised Predictive Control - Part I and II. *Automatica*, 23:137–160, 1987.
- [KLL01] R. Kennel, A. Linder, and M. Linke. Generalized Predictive Control (GPC)-Ready for Use in Drive Applications ? In *IEEE Power Electronics Specialists Conference PELS*, Vancouver, Canada, 17.-22. June 2001.
- [LLM98] I.D. Landau, R. Lozano, and M. M'Saad. Introduction to adaptive control. In B.W. Dickinson, A. Fettweis, J.I. Massey, J.W. Modestino, E.D. Sontag, and M. Thoma, editors, *Adaptive Control*, Communication and Control Engineering, pages 1–30. Springer Verlag, Berlin Heidelberg New York, 1998.
- [Mac93] J. M. Maciejowski. *Multivariable Feedback Design*. Addison-Wesley, Cornwall, Great Britain, 1993.
- [ML98] M. Mahfouf and D.A. Linkens. *Generalised Predictive Control and Bioengineering*. Taylor & Francis, 1 Gunpowder Square, London EC4A 3DE, UK, 1998. ISBN 0-7484-0597-6.
- [Sim02] O. Simanski. *Entwicklung eines Systems zur Messung und Regelung der neuromuskulären Blockade und der Narkosetiefe*. Dissertation, Universität Rostock, Fachbereich Elektrotechnik und Informationstechnik, Institut für Automatisierungstechnik, 2002.

# Active Set Method Approach for Real Time Model Predictive Engine Control

K. N. Lekhadia<sup>1</sup>, H.-P. Dünow<sup>2</sup>, T. Jeinsch<sup>1</sup>

<sup>1</sup> IAV-GmbH, Büro Wismar  
Torsten.Jeinsch@iav.de

<sup>2</sup> Hochschule Wismar  
University of Technology, Business and Design  
Forschungsgruppe Computational Engineering and Automation  
p.duenow@et.hs-wismar.de

## Abstract

In this paper we discuss the implementation of a quadratic programming (QP) optimization algorithm for use as a basic routine in realtime model predictive control (MPC) systems. The goal of the development is to harness MP-Control principles in an electronic control unit (ECU) for combustion engine control. In addition to the QP-routine solved by active set methods, we introduce briefly the implemented MPC-algorithm. The functional efficiency and the performance of the realized controller we show with some realtime and simulation examples.

## 1 Introduction

Customer requirements and environmental concerns have driven the car engine development to be more efficient and less polluting. Several new engine design ideas have been introduced in the last years. This is a challenge for engine control systems because of the increased number of control and sensor signals, the nonlinear characteristics and time-dependents of the engine processes, unavoidable time delays, variable constraints etc. Currently the most controllers in engine control units are formulated by using PI- or PID-controllers with gain scheduling along with map-based feed forward control functions. Around 9000 parameters and more than 600 control and diagnostic functions [Bosch03] are implemented in modern electronic control units. This makes the system highly complex. Adjustments of the gain scheduling controllers are mostly done manually by experts after long research and experimentations.

With the increasing computational powers of ECU's, the implementation of more advanced methods for control and supervision like Neural Networks, Fuzzy control, Adaptive control, Sliding mode, Predictive control and others are feasible. The use of such principles could hopefully handle more complexities and could improve the quality of the engine control systems towards the optimum.

MPC is widely adopted in industries, primarily in chemical engineering processes. This control principle is an established method to deal with large constrained multivariable

control problems. The main idea of MPC is to choose the control action by repeatedly solving an optimization problem. The optimization aims to minimize a performance criterion over a time-horizon subject to selected control and plant signals.

An appropriate criterium is e.g. the quadratic cost function

$$\begin{aligned}
J(k) &= \sum_{i=0}^{H_p} (\hat{y}(k+i|k) - r(k+i))^T Q_i (\hat{y}(k+i|k) - r(k+i)) \\
&+ \sum_{i=0}^{H_u-1} \Delta \hat{u}^T(k+i|k) R_i \Delta \hat{u}(k+i|k).
\end{aligned} \tag{1}$$

$\hat{y}(k+i|k)$  denotes the predicted output for the time  $k+i$  calculated at the time instant  $k$ .  $\Delta \hat{u}$  denotes the difference control signal (4).  $Q$  and  $R$  are the weighting matrices for the input moves and the output deviations respectively. While  $H_p$  and  $H_u$  are prediction and control horizons. In case of multivariable plant,  $\hat{y}$  and  $\Delta \hat{u}$  are vectors of adequate dimensions. To predict a plant output, an appropriate process model is required in predictive control algorithm. On the other side, engine control processes are mostly nonlinear and time dependent. Moreover the state variables and control-signals are usually bounded. Unlike huge computing potentialities in chemical engineering, the hardware capacity of engine control is limited in memory and computing power. Thus the MPC-optimization problems for this application can be very complex and difficult to solve. So to bring improvement to this problem, we have implemented and investigated a suitable quadratic program (QP) in MPC for the application in engine control systems from the available well-known optimization algorithms. The implemented method allows the application to multivariable systems involving variable constraints with reasonable computing efforts. One assumption for the approach is, that the plant can be described piecewise by linear models. The view is mainly directed to the implementation problems of a quadratic programming algorithm for use in a realtime engine control system. Below the basics of the algorithm (also described in [Mac02] or [God01]) are briefly outlined.

## 2 Basic Approach

For the MP-Controller we use a set of linear constant time-discrete state-space process models of the form

$$\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k)
\end{aligned} \tag{2}$$

( $u$ -input signal,  $y$ -output signal,  $x$ -state of the system,  $A$ -system matrix,  $B$ -input- and  $C$ -output matrix).

With this model we can predict the states for a horizon of  $H_p$  steps at time  $k$ :

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+H_u|k) \\ \hat{x}(k+H_{u+1}|k) \\ \vdots \\ \hat{x}(k+H_p|k) \end{bmatrix} = \begin{bmatrix} A \\ \vdots \\ A^{H_u} \\ A^{H_{u+1}} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} B \\ \vdots \\ \sum_{i=0}^{H_u-1} A^i B \\ \sum_{i=0}^{H_u} A^i B \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i B \end{bmatrix} u(k-1) +$$

$$\begin{bmatrix} B & \dots & 0 \\ AB + B & \dots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} A^i B & \dots & B \\ \sum_{i=0}^{H_u} A^i B & \dots & AB + B \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-1} A^i B & \dots & \sum_{i=0}^{H_p-H_u} A^i B \end{bmatrix} \begin{bmatrix} \Delta\hat{u}(k|k) \\ \Delta\hat{u}(k+1|k) \\ \vdots \\ \Delta\hat{u}(k+H_u-1|k) \end{bmatrix} \quad (3)$$

with

$$\begin{aligned} \Delta\hat{u}(k|k) &= \hat{u}(k|k) - u(k-1) \\ \Delta\hat{u}(k+1|k) &= \hat{u}(k+1|k) - \hat{u}(k|k) \\ &\dots \end{aligned} \quad (4)$$

and the outputs of the system:

$$\begin{aligned} \begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+H_p|k) \end{bmatrix} &= \begin{bmatrix} CA \\ \vdots \\ CA^{H_u} \\ CA^{H_u+1} \\ \vdots \\ CA^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} CB \\ \vdots \\ C \sum_{i=0}^{H_u-1} A^i B \\ C \sum_{i=0}^{H_u} A^i B \\ \vdots \\ C \sum_{i=0}^{H_p-1} A^i B \end{bmatrix} u(k-1) + \\ &\begin{bmatrix} CB & \dots & 0 \\ C(AB+B) & \dots & 0 \\ \vdots & \ddots & \vdots \\ C \sum_{i=0}^{H_u-1} A^i B & \dots & CB \\ C \sum_{i=0}^{H_u} A^i B & \dots & C(AB+B) \\ \vdots & \vdots & \vdots \\ C \sum_{i=0}^{H_p-1} A^i B & \dots & C \sum_{i=0}^{H_p-H_u} A^i B \end{bmatrix} \underbrace{\begin{bmatrix} \Delta\hat{u}(k|k) \\ \Delta\hat{u}(k+1|k) \\ \vdots \\ \Delta\hat{u}(k+H_u-1|k) \end{bmatrix}}_{\Delta\mathcal{U}(k)} \\ &= \mathcal{Y}(k) = \Psi x(k) + \Upsilon u(k-1) + \Theta \Delta\mathcal{U}(k) \end{aligned} \quad (5)$$

With the setpoint signal  $r(k+i|k)$ ,  $i = 0 \dots H_p$ , the equation

$$\varepsilon(k) = \begin{bmatrix} r(k|k) \\ r(k+1|k) \\ \vdots \\ r(k+H_p|k) \end{bmatrix} - \Psi x(k) - \Upsilon u(k-1) \quad (6)$$

and the weight matrices

$$\mathcal{Q} = \begin{bmatrix} Q(0) & 0 & \dots & 0 \\ 0 & Q(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q(H_p) \end{bmatrix} \quad (7)$$

respectively

$$\mathcal{R} = \begin{bmatrix} R(0) & 0 & \dots & 0 \\ 0 & R(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R(H_{u-1}) \end{bmatrix} \quad (8)$$

the cost functional (1) becomes to

$$\begin{aligned} J(k) &= [\Delta\mathcal{U}(k)^T \Theta^T - \varepsilon(k)^T] \mathcal{Q} [\Delta\mathcal{U}(k) \Theta - \varepsilon(k)] + \\ &\quad \Delta\mathcal{U}(k)^T \mathcal{R} \Delta\mathcal{U}(k) \\ &= \varepsilon(k)^T \mathcal{Q} \varepsilon(k) - 2\Delta\mathcal{U}(k)^T \Theta^T \mathcal{Q} \varepsilon(k) + \Delta\mathcal{U}(k)^T [\Theta^T \mathcal{Q} \Theta + \mathcal{R}] \Delta\mathcal{U}(k) \\ &= \text{constant} - \Delta\mathcal{U}(k)^T \mathcal{G} + \Delta\mathcal{U}^T \mathcal{H} \Delta\mathcal{U}(k) \end{aligned} \quad (9)$$

$$(10)$$

where

$$\mathcal{G} = 2\Theta^T \mathcal{Q} \varepsilon(k) \quad (11)$$

and

$$\mathcal{H} = \Theta^T \mathcal{Q} \Theta + \mathcal{R}. \quad (12)$$

In case of simple range constraints on control signal we have to take account to the conditions

$$\begin{aligned} u_{min}(k) &\leq u(k|k) \leq u_{max}(k) \\ u_{min}(k+1) &\leq u(k+1|k) \leq u_{max}(k+1) \\ &\vdots \\ u_{min}(k+H_u-1) &\leq u(k+H_u-1|k) \leq u_{max}(k+H_u-1) \end{aligned} \quad (13)$$

which can be represented in a matrix-inequality form as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 & \dots & 1 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & -1 & 0 & 0 & \dots & 0 \\ -1 & -1 & -1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & -1 & \dots & -1 \end{bmatrix} \begin{bmatrix} \Delta\hat{u}(k|k) \\ \Delta\hat{u}(k+1|k) \\ \vdots \\ \Delta\hat{u}(k+H_u-1|k) \end{bmatrix} \leq \begin{bmatrix} u_{max}(k) - u(k-1) \\ u_{max}(k+1) - u(k-1) \\ \vdots \\ u_{max}(k+H_u-1) - u(k-1) \\ -u_{min}(k) + u(k-1) \\ -u_{min}(k+1) + u(k-1) \\ \vdots \\ -u_{min}(k+H_u-1) + u(k-1) \end{bmatrix}$$

with

$$\begin{aligned}
\hat{u}(k|k) &= \Delta\hat{u}(k|k) + u(k-1) \\
\hat{u}(k+1|k) &= \Delta\hat{u}(k+1|k) + \hat{u}(k|k) \\
&= \Delta\hat{u}(k+1|k) + (\Delta\hat{u}(k|k) + u(k-1)) \\
&\vdots
\end{aligned}$$

Thus (with  $u = \Delta\mathcal{U}$ ) the main problem for the MP-Controller is to minimize the cost function

$$J(k) = \frac{1}{2}u(k)^T(2\mathcal{H})u(k) + (-\mathcal{G}^T)u(k) \quad (14)$$

subject to linear inequality constraints

$$\mathcal{A}u(k) \leq \mathcal{U}_{constr}. \quad (15)$$

This is a standard convex optimization problem and efficient numerical procedures (called Quadratic Programming (QP)) are available to solve it. The main implementation steps and some aspects for realtime application are described as follows.

### 3 Active Set methods

The developed QP program here uses Active-Set methods. Active set is a set of constraints which are active at a given point of operation in control process. It solves the quadratic cost function of the form:

$$\min J(u) = \frac{1}{2}u^T H u + \mathcal{G}^T u \quad (16)$$

subject to linear inequality constraints

$$\mathcal{A}u \leq b \quad (17)$$

where  $H = H^T \geq 0$ , i.e. H must be a symmetric positive definite matrix.

Equation (16) is a cost function  $J(u)$ , which has to be minimized.

While here a major emphasis has been laid out in developing a better QP algorithm for a real-time systems from tailor-made algorithms already available, the development of a new theory of QP algorithm is beyond the scope of this paper. And thus the theory presented here relates to those who have already contributed in this field so as just to understand the basic idea. The theoretical explanation of Active Set methods described here is based on [Fle87, GMW81, GMW91, Mac02]. A Flow chart view of this algorithm is also shown in figure 1 for quick understanding of Active-Set methods.

The necessary and sufficient conditions for  $u$  to be the global optimum for convex QP problems, are given by the *Karush-Kuhn-Tucker (or KKT)* conditions [Fle87], which says that: there must exist vectors (Lagrange multipliers)  $\lambda \geq 0$  and  $\zeta$ , and a vector  $t \geq 0$ , such that

$$Hu + A_{eq}^T \zeta + A^T \lambda = -\mathcal{G} \quad (18)$$

$$-Au - t = -b \quad (19)$$

$$t^T \lambda = 0 \quad (20)$$

The Active Set method involves two phases. The first phase involves the calculation of a feasible point (if one exists). The second phase involves the generation of an iterative

sequence of feasible points that converge to the solution. In this method an active set,  $A_a$  (where dimension  $A_a : (m \times n)$ ), is maintained that is an estimate of the active constraints (i.e., those that are on the constraint boundaries) at the solution point.  $A_a$  is updated at each iteration  $k$ , and this is used to form a basis for a search direction SD or  $\Delta u$ . Search direction  $\Delta u$  is calculated and minimizes the objective function while remaining on any active constraint boundaries. The feasible subspace for  $\Delta u$  is formed from a basis  $Z_k$  whose columns are orthogonal to the estimate of the active set  $A_a$  (i.e.,  $A_a Z_k = 0$ ). Thus a search direction, which is formed from a linear summation of any combination of the columns of  $Z_k$ , is guaranteed to remain on the boundaries of the active constraints. The search direction is given by the equation:

$$\Delta u(k) = Z_k \left\{ \frac{-Z_k^T g_k}{Z_k^T H Z_k} \right\} \quad (21)$$

where  $g_k$  is the gradient of the objective function at the current iterate  $u_k$  (i.e.,  $g_k = H u_k + \mathcal{G}$ ) (equal to the coefficients of the linear objective function). A step is then taken of the form

$$u_{k+1} = u_k + \alpha \Delta u \quad (22)$$

At each iteration, because of the quadratic nature of the objective function, there are only two choices of step length  $\alpha$ . A step of unity along  $\Delta u$  is the exact step to the minimum of the function restricted to the null space of  $A_a$ . If such a step can be taken, without violation of the constraints, then this is the solution to QP. Otherwise, the step along  $\Delta u$  to the nearest constraint is less than unity and a new constraint is included in the active set at the next iteration. The distance to the constraint boundaries in any direction  $\Delta u$  is given by

$$\alpha = \min \left\{ \frac{-(A_i u_k - b_i)}{A_i \Delta u} \right\} \quad (23)$$

where ( $i = 1, \dots, m$ ) which is defined for constraints not in the active set, and where the direction  $\Delta u$  is towards the constraint boundary, i.e., Gradient of a search direction  $GSD = A_i \Delta u > 0$ , ( $i = 1, \dots, m$ )

When  $n$  independent constraints are included in the active set, without location of the minimum, Lagrange multipliers,  $\lambda_k$ , are calculated that satisfy the nonsingular set of linear equations

$$A_a^T \lambda_k = g_k \quad (24)$$

The global optimal solution can be checked if KKT condition (18–20) are satisfied for all  $\lambda$  at  $u_{k+1}$ . As  $t$  is already non-negative in equation 20, it simply means that if all  $\lambda$  are also non-negative then the global optimal solution is found. But if any of the  $\lambda < 0$  then KKT condition gets violated and hence such a solution can't be global one and further iterations would be needed. So those inequality constraints which contributes to the negative  $\lambda$  has to be eliminated from the assumed active set. And if more than one element of  $\lambda$  is negative, then the constraint corresponding to the most negative one is removed from the active set. Thus at new iteration the whole process is repeated and a new active set is selected till KKT conditions are fulfilled.

**Initialization:** Still the first phase of active set method to find the initial feasible solution is remaining. This is needed in order to get the iterations of the active set method started. You can modify the preceding QP algorithm for LP problems by setting the

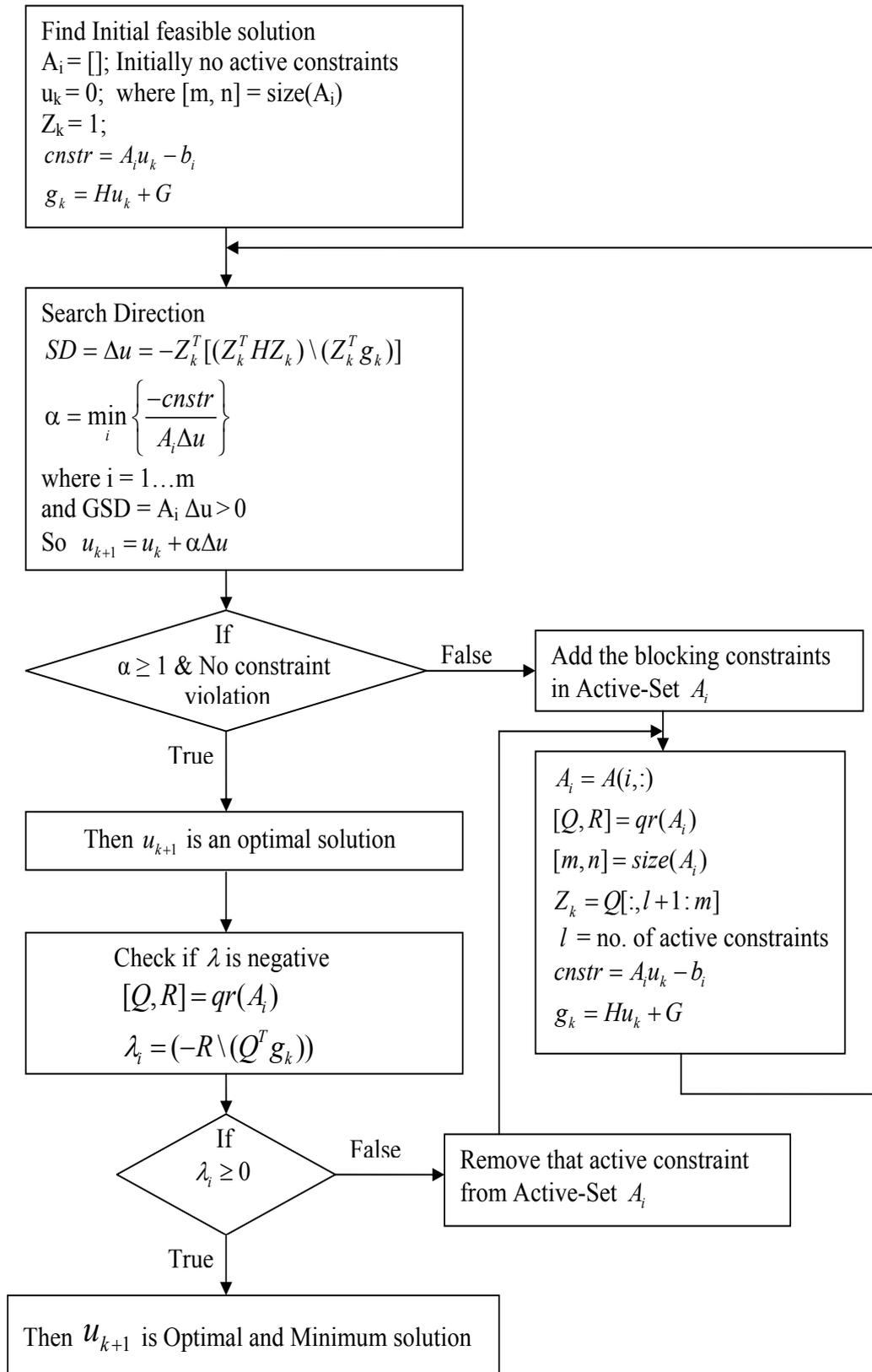


Fig. 1: Active-Set Algorithm in Flow Chart

search direction to the steepest descent direction at each iteration by using  $Z_k = 1$ .

$$\Delta u = Z_k \left\{ \frac{-Z_k^T g_k}{Z_k^T H Z_k} \right\} \quad (25)$$

If a feasible point is found using the preceding LP method, the main QP phase is entered. The search direction  $\Delta u$  is initialized with a search direction  $\Delta u_1$  found from solving the set of linear equations

$$H \Delta u_1 = -g_k \quad (26)$$

For implementation of subroutines in the QP-algorithm results from freely available numerical packages LAPACK and BLAS were used ([Lek04]).

## 4 Experiments and Results

**Realtime Experiments** The implemented MPC-algorithm was investigated in a C-S-function for Simulink to use it on a xpcTarget-system and with the matlab realtime-toolbox of Humusoft ([Hum02]). It should be also possible to compile it for any embedded controller system with appropriate memory and computing resources.

For laboratory experiments we used a 2-input/2-output airflow-system for realtime and performance-tests (Fig. 2). Control inputs are the blower and a throttle. The target is

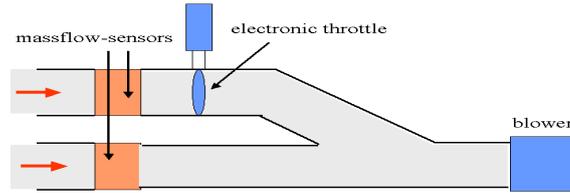


Fig. 2: Test Process

to control the massflow in the two tubes on the left side. The process variables of the system are highly coupled. The simulink-model with the realtime block and the process interfacing is shown in Fig. 3. The C-S-Function was observed to be almost 7 times faster than the original quadprog program available in matlab. Figure 4 shows a plot of an input-

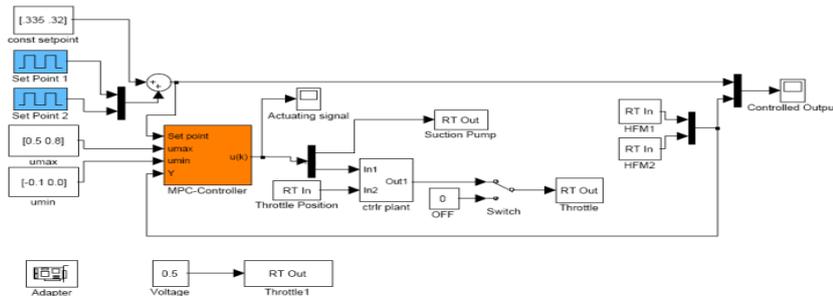


Fig. 3: Simulink model with realtime interface

output behavior for a MPC controlled airflow system in presence of constraints. Where various tuning parameters were set to the following values: prediction horizon  $H_p = 20$ , control horizon  $H_u = 8$ . The upper limit of the input constraints  $umax = [0.5, 0.8]$ , while the lower limit of the input constraints  $umin = [-0.1, 0.0]$ . Simulation time was 60 sec and sampling time was 50 millisecond. In this figure the input constraint on the signal of blower is found to be active over the entire period of simulation. And thus the controller output for the pump strictly followed the constraint without violating it.

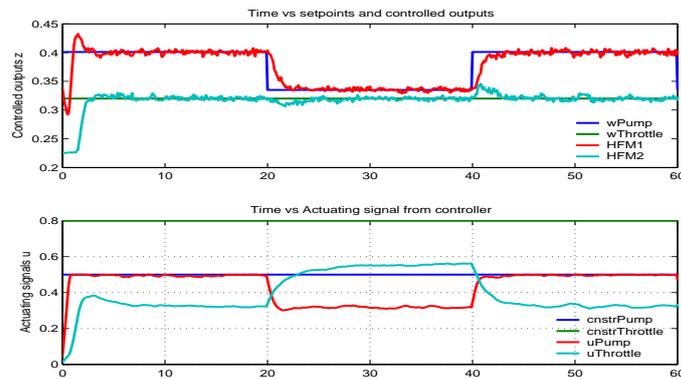


Fig. 4: MPC-Control of the system

**Engine Control Example: Idle Speed** A typical multivariable engine control problem with constraints is the control of the idle speed. The main actuator variables are the advance angle and the throttle (airflow). The main control variable is the idle speed (Figure 5). In the MPC also the advance angle is used as process output variable with

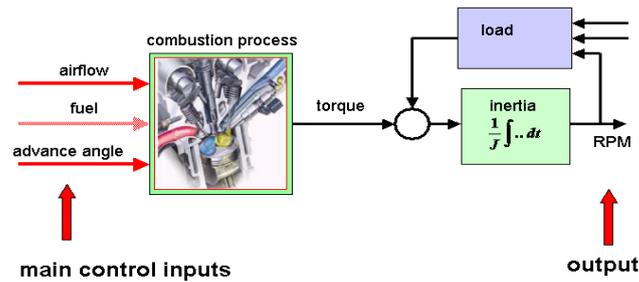


Fig. 5: idle speed process

a set point of zero. In this way static values in the advance angle controller output are disappear. The linear transfer matrix for this process is of the form

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{bmatrix} G_1 & G_2 \\ 0 & \frac{1}{sT+1} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (27)$$

with the idle speed  $y_1$  and the advance angle  $y_2$  as outputs.  $u_1$  is the airflow -input and  $u_2$  the advance angle input. Figure 6 shows a result of a simulation experiment. The advance

angle input is limited. Because of the chosen model (27) this input becomes to zero after a while. With the time constant  $T$  as a design parameter in the transfer matrix this time can be influenced. For more practical engine experiments the xpcTarget-System will be

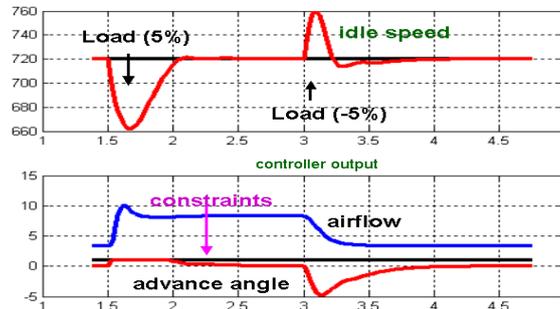


Fig. 6: Simulation of idle speed - MPC

used to work as a bypass control-unit for a SI- engine (Fig. 7).

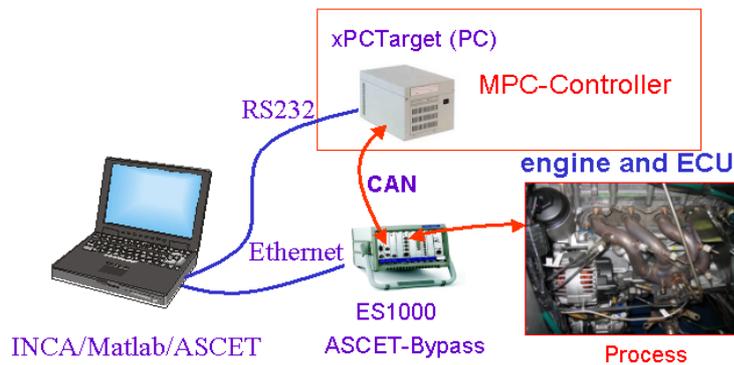


Fig. 7: ECU- Bypass System

## 5 Conclusion

Although the described MP-Control concept uses linear process models it is a promising method for advanced engine control especially in the existence of constraints. With the formulation as QP-problem and the solution by the described active set method there should be a chance to implement it in future ECU's for realtime engine control. Based on the developed algorithm a number of enhancements are imaginable. Some ideas are

- the use of multimodels for consideration of operating points,
- the estimation and consideration of future constraints and set points,

- the use of time- and operating point depending weight matrices or
- the implementation of (slow) optimization shell for the tuning of the design parameters (weight matrices, horizons,etc.)

For all this points the QP-procedure solved by active set method remains the kernel of the control algorithm.

## References

- [Bosch03] Robert Bosch GmbH(Hrsg.). *Ottomotor-Management: System und Komponenten*. Friedr. Vieweg & Sohn verlagsgesellschaft mbH, 2 edition, 2003.
- [Fle87] R.R. Fletcher. *Practical Methods of Optimization*. Wiley, 2nd edition, 1987.
- [GMW81] P.E. Gill W. Murray and M.H. Wright. *Practical Optimization*. Academic Press, London, 1981.
- [GMW91] P.E. Gill W. Murray and M.H. Wright. *Numerical Linear Algebra and Optimization*, volume 1. Addison Wesley, 1991.
- [God01] M. E. Salgado G. C. Goodwin, S. F. Graebe. *Control System Design*. Prentice Hall, London, 2001.
- [Hum02] Humusoft. Matlab-realtime toolbox. Prague, 2002.
- [Lek04] K. Lekhadia. Development of qp-algorithms for real-time predictive control systems, master thesis. Technical report, Fachhochschule Darmstadt, September 2004.
- [Mac02] J.M.Maciejowski. *Predictive Control with constraints*. Prentice Hall, 2002.





