

R. Mikut, M. Reischl (Hrsg.)

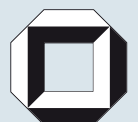
Institut für Angewandte Informatik /
Automatisierungstechnik



Proceedings

16. Workshop Computational Intelligence

Dortmund,
29. November –
1. Dezember 2006



R. Mikut, M. Reischl (Hrsg.)

Proceedings 16. Workshop Computational Intelligence

Dortmund, 29. November - 1. Dezember 2006

Schriftenreihe des
Instituts für Angewandte Informatik / Automatisierungstechnik
an der Universität Karlsruhe (TH)
Band 15

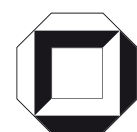
Eine Übersicht über alle bisher in dieser Schriftenreihe erschienenen
Bände finden Sie am Ende des Buchs.

Proceedings

16. Workshop Computational Intelligence

Dortmund, 29. November - 1. Dezember 2006

R. Mikut
M. Reischl
(Hrsg.)



universitätsverlag karlsruhe

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2006
Print on Demand

ISSN: 1614-5267
ISBN-13: 978-3-86644-057-9
ISBN-10: 3-86644-057-X

VORWORT

Dieser Tagungsband enthält die Beiträge des 16. Workshops „Computational Intelligence“ des Fachausschusses 5.14 der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und der Fachgruppe „Fuzzy-Systeme und Soft-Computing“ der Gesellschaft für Informatik (GI), der vom 29. November – 1. Dezember 2006 im Haus Bommerholz bei Dortmund stattfindet.

Der GMA-Fachausschuss 5.14 „Computational Intelligence“ entstand 2005 aus den bisherigen Fachausschüssen „Neuronale Netze und Evolutionäre Algorithmen“ (FA 5.21) sowie „Fuzzy Control“ (FA 5.22). Der Workshop steht in der Tradition der bisherigen Fuzzy-Workshops, hat aber seinen Fokus in den letzten Jahren schrittweise erweitert.

Die Schwerpunkte sind Methoden, Anwendungen und Tools für

- Fuzzy-Systeme,
- Künstliche Neuronale Netze,
- Evolutionäre Algorithmen und
- Data-Mining-Verfahren

sowie der Methodenvergleich anhand von industriellen und Benchmark-Problemen.

Die Mitglieder des Programmkomitees sind Dr. Mikut (Forschungszentrum Karlsruhe GmbH), Prof. Dr. H.-G. Beyer (FH Vorarlberg), Prof. Dr. A. Kistner (Universität Stuttgart), Prof. Dr. F. Klawonn (FH Braunschweig/Wolfenbüttel), Dr. A. Kroll (ABB Forschungszentrum) und PD Dr. Th. Runkler (Siemens AG).

Die Ergebnisse werden von Teilnehmern aus Hochschulen, Forschungseinrichtungen und der Industrie in einer offenen Atmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

Nähere Informationen zum GMA-Fachausschuss erhalten Sie unter

<http://www.iai.fzk.de/medtech/biosignal/gma/index.html>.

Die Herausgeber bedanken sich an dieser Stelle bei allen Autoren und Rednern sowie bei den Mitgliedern des Programmkomitees, die maßgeblich an der Vorbereitung des Workshops beteiligt waren.

Ralf Mikut und Markus Reischl

INHALTSVERZEICHNIS

N. Beume, B. Naujoks, G. Rudolph <i>Universität Dortmund:</i> Mehrkriterielle Optimierung durch evolutionäre Algorithmen mit S-Metrik-Selektion	1
C. Blume <i>FH Köln, Campus Gummersbach:</i> GLEAM - Ein EA für Prozessabläufe am Beispiel von Steuerungen für Industrieroboter	11
T. Bartz-Beielstein, M. Preuß <i>Universität Dortmund:</i> Moderne Methoden zur experimentellen Analyse evolutionärer Verfahren	25
T. Bäck, F. Duddeck, M. Schütz <i>NuTech Solutions GmbH, London University:</i> Efficient Product Development of Car Bodies Using Multi-disciplinary Optimization	33
H. Mielenz, R. Dölling, W. Rosenstiel <i>Robert Bosch GmbH Reutlingen, Universität Tübingen:</i> Modellierungsmethodik für Mixed-Signal Automotive Schaltkreise mittels Data-Mining Methoden und Fuzzy-Systemen	47
T. A. Runkler <i>Siemens AG, München:</i> The Effect of Kernelization in Relational Fuzzy Clustering	48
H. Schulte <i>Bosch-Rexroth AG, Elchingen:</i> Modellierung und Simulation hydrostatischer Getriebe in mobilen Arbeitsmaschinen mittels Takagi-Sugeno-Fuzzy-Systemen	62
P. Bretschneider <i>FhG AST Ilmenau:</i> Methoden und Lösungen für das Energiemanagement im liberalisierten Energemarkt	71
E. Garcia, P. Bretschneider, Th. Rauschenbach <i>FhG AST Ilmenau:</i> Vergleich nichtlinearer Vorhersagemodelle anhand von Referenzbeispielen	77
A. Gommlich, Th. Alber <i>Endress+Hauser Conducta GmbH+Co. KG, Gerlingen:</i> Methoden der Sensordiagnose zur Bewertung von Verfügbarkeit, Qualität und Sicherheit	90
A. Grau, J. Brenig, J. Krone, U. Lehmann <i>Fachhochschule Südwestfalen, Abteilung Iserlohn:</i> Optimierungsplattform für verteilte Computational Intelligence Controller	96

Ch. Katz, T. A. Runkler, J. Hollatz <i>Siemens AG, München, Erlangen:</i> European Monetary Union Revisited: How Good was the Fuzzy Cognitive Map Prediction?	105
R. Mikut, O. Burmeister, M. Reischl, T. Loose <i>Forschungszentrum Karlsruhe:</i> Die MATLAB-Toolbox Gait-CAD	114
T. Schmidt, D. Henrich <i>Universität Bayreuth:</i> Vergleich von klassischem und temporalem Fuzzy-Regler beschrieben in Fuzzy Control Language mit PID-Reglern	125
J. Beringer, E. Hüllermeier <i>Universität Magdeburg:</i> Adaptive Optimization of the Number of Clusters in Fuzzy Clustering	140
R. Warias, M. Gerke <i>Fernuniversität Hagen:</i> Zielorientierte Kollisionsvermeidung für autonome mobile Systeme	150
M. Sterneke, P. Krause <i>Universität Dortmund, NuTech Solutions GmbH:</i> Formulierung problemspezifischer Einstellregeln für das Fuzzy-ROSA-Verfahren	165
F. Höppner <i>FH Braunschweig/Wolfenbüttel:</i> Klassifikation von Zeitreihen und Sequenzen	179
E. Hüllermeier, K. Brinker <i>Universität Magdeburg:</i> Classification via Fuzzy Preference Learning	190
R. Rettberg, H. Sommer <i>Universität Kassel:</i> Granulierungseffekte in unscharf definierten Medien bei Strukturbildungsprozessen mit Bottom-Up-Verfahren	200
A. Lodwich, B. Zaaimi, R. Grebe, J. Krone, U. Lehmann, F. Wallois <i>Fachhochschule Südwestfalen Iserlohn, Université Jules Verne de Picardie Amiens:</i> Optimierung der Parameter für die Vagusnervstimulation mit KNN zur Behandlung refraktärer Epilepsie bei Kindern (OptiVaNeS)	210
N. Cebron, M. Berthold <i>Universität Konstanz:</i> Adaptive Klassifikation von Zellbildern	223
O. Burmeister <i>Forschungszentrum Karlsruhe:</i> Analyse von Zeitreihen in der Medizin: Informationsgehalt, Klassifikation und Unsicherheit	234

J. Limberg, R. Seising <i>Medizinische Universität Wien:</i> Fuzzy-Konzepte für die Begriffe Gesundheit, Kranksein und Krankheit. Darstellung und Implementierung eines Ansatzes von Kazem Sadegh-Zadeh	248
R. Seising <i>Medizinische Universität Wien:</i> Fuzzy Sets als Werkzeug der Wissenschaftstheorie	262
W. Brockmann, D. Meyer, A. Horst <i>Universität Osnabrück:</i> Ein immunsystem-inspirierter Ansatz zum Überwachen des Lernens in Neuro-Fuzzy-Systemen	277
C. Endisch, D. Schröder <i>TU München:</i> Quasi-Online Newtonverfahren zur Identifikation nichtlinearer dynamischer Systeme mit strukturierten rekurrenten Netzen	291

Mehrkriterielle Optimierung durch evolutionäre Algorithmen mit S-Metrik-Selektion

Nicola Beume, Boris Naujoks und Günter Rudolph

Lehrstuhl für Algorithm Engineering, Universität Dortmund
44221 Dortmund, Deutschland

Tel.: (0231) 755-7701

Fax: (0231) 755-7740

E-Mail: {nicola.beume, boris.naujoks, guenter.rudolph}@udo.edu,

<http://ls11-www.cs.uni-dortmund.de/>

Zusammenfassung

Bei der mehrkriteriellen Pareto-Optimierung wird zu konfliktären Anforderungen eine Menge von Kompromisslösungen gesucht, die die bestmöglichen Lösungen approximieren. Evolutionäre Algorithmen haben sich hierbei als effektive und robuste Verfahren bewährt. Die Güte einer Approximation lässt sich durch das von ihr dominierte Hypervolumen im Zielraum, der sogenannten S-Metrik, quantifizieren. Die Maximierung der S-Metrik ist also ein erstrebenswertes Ziel und gleichzeitig eine hinreichende einkriterielle Ersatzzielfunktion. Ein evolutionärer Algorithmus setzt diese innerhalb der Selektion ein und erreicht dadurch hervorragende Ergebnismengen. Wir zeigen anhand von Benchmarkproblemen und realen Anwendungen aus der Flugzeugtechnik, dass dieser Algorithmus außerordentlich effektiv ist. Dies zeigt sich insbesondere für den Fall, wenn mehr als drei Ziele zu optimieren sind, weil dann andere populäre mehrkriterielle evolutionäre Algorithmen versagen.

1 Einführung in Mehrkriterielle Evolutionäre Algorithmen

Bei der Mehrzieloptimierung werden an eine Problemlösung mehrere Anforderungen gestellt, die es bestmöglich zu erfüllen gilt. Diese Anforderungen (*Zielfunktionen*) sind typischerweise konfliktär, sodass ein Optimum bezüglich aller Funktionen nicht mit einer Lösung zu erreichen ist. Wie aus dem Alltag vertraut, gibt es keine Lösung, die alle Wünsche optimal erfüllt.

Mehrzieloptimierverfahren suchen bei der Pareto-Optimierung nicht nach *einer* möglichst guten Lösung, sondern nach einer Menge von Kompromisslösungen, aus denen der Anwender nach seinen Vorlieben eine zur Realisierung auswählt. Eine Lösung eines d -kriteriellen Problems wird über ihren zugehörigen d -dimensionalen Zielfunktionsvektor bewertet. Zielfunktionsvektoren sind partiell geordnet, d. h. zwei Vektoren sind unvergleichbar, falls jeder eine bessere Komponente enthält als der andere. Ein Vektor *dominiert* einen anderen, falls er in mindestens einer Komponente besser ist und in keiner schlechter. Die Lösungen, deren Zielfunktionsvektoren von keinem in der Lösungsmenge dominiert werden, heißen *nicht-dominiert*. Die Zielfunktionsvektoren, die bezogen auf den gesamten Suchraum nicht-dominiert, also *Pareto-optimal* sind, heißen *Pareto-Front* und ihre erzeugenden Lösungen *Pareto-Menge*.

Die Zielsetzung von Optimierverfahren ist meist die Approximation der Pareto-Front. Es sollen möglichst viele nicht-dominierte Lösungen gefunden werden, damit dem Anwender eine große Auswahl zur Verfügung steht. Die gefundenen Zielfunktionsvektoren

sollen nah an der Pareto-Front liegen und sich dabei entlang der gesamten Pareto-Front verteilen.

Evolutionäre Mehrziel-Optimier-Algorithmen (EMOA) haben sich als leistungsfähige Verfahren bewährt. Es sind randomisierte Suchheuristiken, die eine gute Lösung nicht durch Analyse des Problems konstruieren, sondern durch Ausprobieren von Lösungsmöglichkeiten schrittweise zu Verbesserungen gelangen. Sie sind inspiriert von der darwinistischen Vorstellung der natürlichen Evolution als eines Verbesserungsprozesses durch Reproduktion, Variation und Auslese. Eine grundlegende Übersicht zu EMOA bieten Deb [1] oder Coello Coello et al. [2].

Hat man mehrere Mehrzieloptimierverfahren zur Verfügung, möchte man ihre Ergebnismengen vergleichen, um herauszufinden, welches Verfahren für das Problem am geeignetsten erscheint. Gibt es in einer Menge einen Vektor, der bzgl. der Pareto-Dominanz mit keinem in der anderen Menge vergleichbar ist, dann sind auch die Mengen unvergleichbar. Dies ist der typische Fall bei Vergleichsstudien, daher verwendet man Qualitätsmaße (*Metriken*), die die Güte einer Menge bezüglich eines Qualitätsmerkmals quantifizieren. Entsprechend der skalaren Metrikerwerte können Mengen vollständig geordnet werden, wobei die Sinnhaftigkeit vieler Metriken umstritten ist. Eine einfache Metrik ist die *Konvergenz-Metrik*, die den durchschnittlichen Abstand der Menge zur Pareto-Front misst.

Unter den zahlreichen Metriken gilt die *S-Metrik* von Zitzler und Thiele [3] als eines der sinnvollsten und gerechtesten Maße (vgl. Zitzler et al. [4]). Sie misst das dominierte Hypervolumen, d.h. den Bereich des Zielraums, in dem sich Vektoren befinden, die schlechter sind als mindestens ein Mitglied der Menge. Der S-Metrikerwert ist zu maximieren, was bildlich einer „Eroberung“ des Raums entspricht. Das größte Hypervolumen wird von der Pareto-Front dominiert.

2 Dominiertes Hypervolumen als Selektionskriterium

2.1 Motivation und Ablauf des SMS-EMOA

Da die S-Metrik ein anerkannter Qualitätsindikator ist, ist ein hoher S-Metrikerwert der Ergebnismenge erstrebenswert. Somit ist es naheliegend, die Maximierung des S-Metrikerwertes schon während des Optimierprozesses explizit anzusteuern. Dies wurde realisiert in dem S-Metrik-Selektions-EMOA (SMS-EMOA) von Emmerich et al. [5, 6]. Das Hypervolumen-basierte Selektionskriterium bildet die partiell geordneten Zielfunktionsvektoren auf skalare Fitnesswerte ab. Dies ermöglicht eine vollständige Ordnung der Lösungen und eine entsprechende Selektion der Besten. Die Selektion wählt die Teilmenge von Lösungen als Folgepopulation aus, deren S-Metrikerwert unter den möglichen Teilmengen maximal ist.

Der Ablauf des SMS-EMOA ist in Algorithmus 1 dargestellt. Für die Startpopulation werden μ Lösungen (*Individuen*) zufällig generiert oder mit bekannten Problemlösungen initialisiert. Jede Generation des evolutionären Prozesses beginnt mit der Variation, die mit Hilfe randomisierter Operatoren aus der Population der vorhandenen Lösungen genau ein neues Individuum erzeugt. Es wurden keine speziellen Variationsoperatoren für den SMS-EMOA entworfen, diese können aus der Literatur (siehe z.B. Bäck et al. [7], Deb [1]) oder problemspezifisch gewählt werden.

$P_0 \leftarrow \text{init}$	<i>Initialisiere Start-Population mit μ Individuen</i>
$t \leftarrow 0$	<i>Initialisiere Generationenzähler</i>
repeat	
$\mathbf{o} \leftarrow \text{variation}(P_t)$	<i>Erzeuge 1 Ind. durch Variationsoperatoren</i>
$D \leftarrow \text{dominierte_Individuen}(P_t \cup \mathbf{o})$	
if $D \neq \emptyset$ then	
$\mathbf{a}^* \leftarrow \text{argmax}_{\mathbf{a} \in D} [dz(\mathbf{a}, P_t \cup \mathbf{o})]$	<i>Ind. mit max. Dominanzzahl</i>
else	
$\mathbf{a}^* \leftarrow \text{argmin}_{\mathbf{a} \in \{P_t \cup \mathbf{o}\}} [\Delta_S(\mathbf{a}, P_t \cup \mathbf{o})]$	<i>Ind. mit min. S-Metrik-Beitrag</i>
$P_{t+1} \leftarrow \{P_t \cup \mathbf{o}\} \setminus \{\mathbf{a}^*\}$	<i>entferne gewähltes Ind.</i>
$t \leftarrow t + 1$	
until <i>Abbruchkriterium erfüllt</i>	

Algorithmus 1: SMS-EMOA

Die Selektionskomponente reduziert die $\mu + 1$ Lösungen auf μ . Moderne EMOA bestehen aus einem primären Selektionskriterium, das auf der Pareto-Dominanz basiert und einem sekundären, welches herangezogen wird, falls nicht alle bzgl. des primären Kriteriums gleichwertigen Individuen in die Folgepopulation aufgenommen werden können. Beim SMS-EMOA ist das primäre Kriterium die Einteilung der Population in nicht-dominierte und dominierte Individuen. Falls die Population dominierte Individuen enthält, wird unter diesen das mit der größten *Dominanzzahl* aussortiert. Die Dominanzzahl eines Individuums ist die Anzahl Individuen, die dieses dominieren. Eine Lösung gilt also als schlecht, wenn es viele bessere gibt. Besteht die Population nur aus nicht-dominierten Individuen, wird das mit kleinstem S-Metrik-Beitrag entfernt. Das ist das Hypervolumen, welches nur von einem einzigen Individuum dominiert wird und somit verfällt, wenn es aus der Population entfernt wird. Der kleinste Beitrag minimiert diesen Verlust.

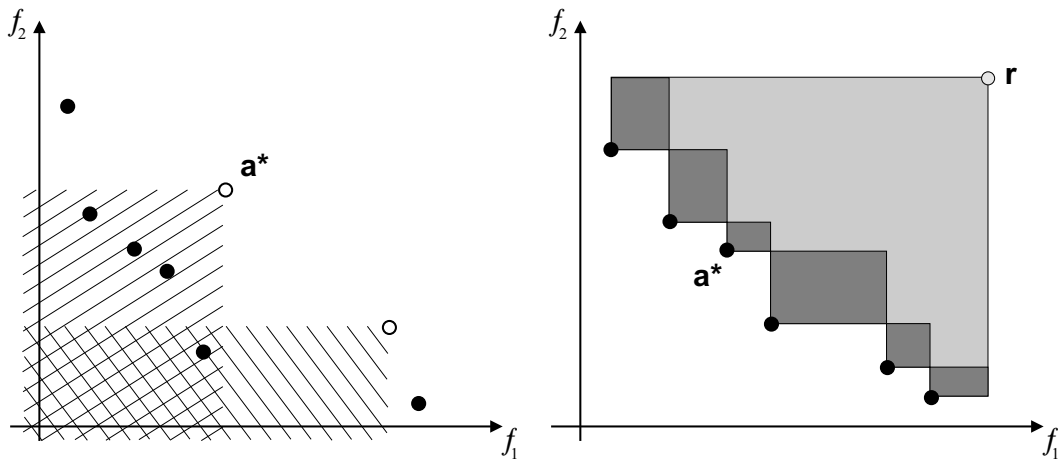


Bild 1: In einem 2-dimensionalen Zielraum von Minimierungsproblemen sind die Werte der Individuen eingezeichnet. Links: Für die beiden dominierten Lösungen sind die Bereiche, in denen dominierende Lösungen liegen, schraffiert. Rechts: Die dunklen Rechtecke zeigen die S-Metrik-Beiträge der Punkte. Der Referenzpunkt \mathbf{r} begrenzt den dominierten Raum.

2.2 Der SMS-EMOA aus Anwendersicht

Der SMS-EMOA zeichnet sich dadurch aus, dass er sehr leistungsfähig ist und außer den Parametern, die für jeden evolutionären Algorithmus spezifiziert werden müssen, keine zusätzlichen benötigt. Seine Anwendung erfordert daher kein tieferes Expertenwissen. Wie bei jedem EMOA müssen zur Bewertung der Lösungen Zielfunktionen definiert werden, die die gewünschten Anforderungen in einen mathematischen Zusammenhang setzen. Die Variationsoperatoren werden problemspezifisch gewählt, in Abhängigkeit der Repräsentation der Lösungen. Für typische Szenarien wie die reellwertige (restringierte) Optimierung oder kombinatorische Probleme sind bewährte Operatoren bekannt. Die Populationsgröße sollte in Abhängigkeit der zur Verfügung stehenden Optimierzeit gewählt werden. Sind Funktionsauswertungen zeitaufwändig, können unter Zeitbeschränkung nur wenige Generationen durchgeführt werden. Die Populationsgröße sollte dann klein sein, damit die Individuen der Population hinreichend oft ausgetauscht werden.

Die Rechenzeit der Operatoren eines EMOA ist normalerweise gegenüber der Zeit für die Funktionsauswertungen vernachlässigbar. Die S-Metrik-Berechnung ist in hoch-dimensionalen Räumen allerdings rechenintensiv, sodass der SMS-EMOA zeitaufwändig ist. Bei zwei oder drei Zielfunktionen ist die Rechenzeit ähnlich wie bei anderen EMOA.

Die Besonderheit des SMS-EMOA besteht darin, dass er bei hoch-dimensionalen Zielfunktionen überhaupt funktioniert. Populäre Verfahren wie NSGA-II oder SPEA2 erreichen selbst auf gängigen Testfunktionen keine Konvergenz gegen die Pareto-Front (vgl. Beume et al. [8]).

2.3 Der SMS-EMOA aus wissenschaftlicher Sicht

Aus der Sicht eines Algorithmikers hat der SMS-EMOA interessante Eigenschaften. Da immer die Folgepopulation mit dem höchsten S-Metrik-Wert ausgewählt wird, ist der S-Metrik-Wert der Population über die Generationen monoton steigend. Da der optimale S-Metrikwert von der Pareto-Front angenommen wird, strebt die Population darauf zu. Das eigentliche Optimierziel stimmt demnach mit der Maximierung der S-Metrik überein und sie erweist sich als adäquate einkriterielle Ersatzzielfunktion.

Die Berechnung der S-Metrik ist eine große algorithmische Herausforderung. Im Rahmen der Entwicklung des SMS-EMOA konnte die Laufzeit der S-Metrik-Berechnung von $O(n^{d-1})$ auf $O(n \log n + n^{d/2})$ für n Punkte im d -dimensionalen Zielraum (vgl. Beume und Rudolph [9]) gesenkt werden. Eine Generation des SMS-EMOA hat ohne die Rechenzeit für die Funktionsauswertung eine Laufzeit von $O(n \log n + n^{(d/2)+1})$.

Das sekundäre Selektionskriterium ist notwendig, weil in mehr-dimensionalen Zielräumen viele Individuen bzgl. der Dominanzrelation unvergleichbar sein können. Ein Punkt in einem d -dimensionalen Raum ist nur mit einem Anteil von $1/2^{d-1}$ des Zielraums vergleichbar. Das sekundäre Selektionskriterium ist bei Standard-EMOA eine Art Diversitätsmaß, das die Verteilung der Punkte optimiert, aber den Fortschritt in Richtung der Pareto-Front kaum berücksichtigt. Das rechenintensive, aber informative Selektionskriterium des SMS-EMOA ist in der Lage, den Optimierprozess kompetent zu steuern.

Der S-Metrikwert wird bzgl. eines Referenzpunktes berechnet. Dieser muss allerdings nicht vom Anwender spezifiziert werden, sondern wird durch eine interne Regelung in jeder Generation neu bestimmt.

3 Studien auf Testfunktionen und Anwendungen in der Flugzeugtechnik

3.1 Akademische Testfunktionen

Für Vergleichsstudien gelten die Familien der ZDT- und DTLZ-Funktionen [10, 11] als anerkannte mehrkriterielle Testfunktionen. Etablierte EMOA wurden auf diesen Funktionen von Deb et al. [12] studiert und die Analyse des SMS-EMOA diesen Studien mit gleicher Parametrisierung nachempfunden. Die Ergebnismengen wurden mit Hilfe der S-Metrik und der Konvergenz-Metrik [12] bewertet.

Auf den zwei-kriteriellen ZDT-Funktionen können alle EMOA als erfolgreich angesehen werden und erzielen ähnliche Ergebnisse. Der SMS-EMOA erreicht auf der Testfamilie insgesamt die besten Ergebnisse (vgl. Emmerich et al. [5]). Die Anzahl der Zielfunktionen ist bei der DTLZ-Familie skalierbar. Mit zunehmender Dimension wird die Überlegenheit des SMS-EMOA deutlicher. Bereits bei vier-kriteriellen Funktionen sind die EMOA NSGA-II [13], SPEA2 [14] und ϵ -MOEA [12] nicht mehr konkurrenzfähig (vgl. Beume et al. [8]). Der ϵ -MOEA ist unter diesen der beste bei steigender Dimension, vorausgesetzt der sensible Parameter ϵ ist mit hoher Genauigkeit passend eingestellt. Der Leistungsabfall des NSGA-II ist besonders deutlich. Er erreicht auf der vier-kriteriellen DTLZ1 nicht einmal die Hälfte des S-Metrikwertes des SMS-EMOA und sein Wert bzgl. des Konvergenzmaßes ist um etwa das 1000-fache schlechter. Die Qualität der Ergebnismenge des SMS-EMOA bleibt bei steigender Dimension relativ gleich. Es werden immer ca. 98% des S-Metrikwertes der Pareto-Front erreicht und die gefundenen Punkte liegen nahezu auf der Pareto-Front.

Das einzige evolutionäre Verfahren, das auch bei hoch-dimensionalen Zielfunktionen konkurrenzfähig zum SMS-EMOA ist, wird derzeit von der Arbeitsgruppe analysiert (vgl. Beume et al. [8]). Es ist ein neuartiger Algorithmus namens MSOPS von Hughes [15], der statt der Pareto-Relation intern verschiedene Aggregationen der Zielfunktionen verwendet. Der Erfolg des Algorithmus hängt stark von den Gewichtungen der Zielfunktionen ab, die vom Anwender spezifiziert werden müssen. Demgegenüber hat der SMS-EMOA den Vorteil, dass er nahezu parameterlos arbeitet.

3.2 Anwendung in der Flugzeugtechnik

In der Flugzeugindustrie kommen Optimierverfahren in vielen Bereichen zur Anwendung. Die Tragflächenoptimierung bietet einfache Anwendungen, wie das Design von Tragflächenprofilen, insbesondere die Rekonstruktion optimaler Profile zu verschiedenen Strömungsbedingungen, die Optimierung von Widerstand und Auftrieb einer Tragfläche oder die Widerstandsminimierung unter verschiedenen Strömungsbedingungen, wozu das vorgestellte Anwendungsproblem zählt. Die drei Zielfunktionen sind die Widerstandsminimierung der Referenztragfläche RAE 2822 unter drei verschiedenen Flugphasen. Die entsprechenden Strömungsbedingungen sind Tabelle 1 zu entnehmen.

Für weitere relevante Kennzahlen (neben dem Widerstandsbeiwert) sind Restriktionen zu beachten, damit sie nicht zu stark von den Werten der Referenztragfläche abweichen.

Tabelle 1: Strömungsbedingungen für die Widerstandsminimierung der Referenztragfläche RAE 2822 im Anwendungsfall Tragflächenoptimierung

	Reiseflug	Abweichung 1	Abweichung 2
Machzahl M	0.734	0.754	0.680
Reynoldszahl Re	$6.5 \cdot 10^6$	$6.2 \cdot 10^6$	$5.7 \cdot 10^6$
Anstellwinkel α	2.8	2.8	1.8
Transition	3%	3%	11%

Der Auftriebsbeiwert muss mit dem der Referenztragfläche übereinstimmen und die maximale Dicke der Referenztragfläche darf nicht überschritten werden. Erlaubt sind Abweichungen des Kippmoments um 2%, der Dicke der Tragfläche um 5%, des Radius der Flügelspitze um 10% und des Winkels der Profilhinterkante um 20%.

Der SMS-EMOA wurde auf der beschriebenen Anwendung mit dem NSGA-II von Deb et al. [13], einem Standardverfahren der evolutionären multikriteriellen Optimierung, verglichen (vgl. Naujoks et al. [16]). Beide Verfahren verwendeten die selben Variationsoperatoren mit identischer Paramterisierung und eine Populationsgröße von 20 Individuen. Fünf Läufe wurden jeweils mit der Referenztragfläche in der initialen Population gestartet und nach 1000 Zielfunktionsauswertungen gestoppt. Die Ergebnisse des NSGA-II sind in Abbildung 2 in der linken Spalte aufgeführt und die des SMS-EMOA in der rechten. Zur Verdeutlichung sind die Ergebnismengen als zwei-dimensionale Projektionen auf die Ebenen $f_3 = 0$, $f_2 = 0$ und $f_1 = 0$ dargestellt.

In den oberen Abbildungen weisen beide Verfahren die geringsten Abweichungen von einander auf. Es lässt sich lediglich eine leichte Streckung der Resultate entlang einer gedachten Geraden feststellen, um die die Lösungen angeordnet sind. Dies führt zu einer besseren Verteilung der Lösungen über den dargestellten Bereich des Suchraums. Allerdings lassen sich daraus noch keine Rückschlüsse für die Leistungsfähigkeit der Verfahren ziehen.

Die Unterschiede zwischen den Verfahren werden deutlich, vergleicht man die mittlere und die untere Zeile der Abbildung 2. Beide Projektionen erscheinen für die jeweiligen Algorithmen recht ähnlich, was mit der beschriebenen Korrelation der Werte bzgl. f_1 und f_2 zusammenhängt. Aufgrund der Ähnlichkeit der Graphiken konzentriert sich die folgende Analyse auf die Beschreibung eines Falls.

Es lässt sich deutlich eine Streckung der Ergebnisse des SMS-EMOA über den dargestellten Bereich des Suchraums erkennen. Ferner sind einzelne Ergebnisse wesentlich näher an der unteren linken Ecke der Graphik positioniert, was einer Verschiebung in Richtung besserer Lösungen entspricht. Insbesondere fallen Lösungen im rechten, unteren Bereich der Graphik für den SMS-EMOA auf, die die Werte der Referenztragfläche dominieren. Dies bedeutet eine Verbesserung der Referenztragfläche bezüglich aller Kriterien, also aller Druckbeiwerte, unter den gegebenen Strömungsbedingungen und unter Erfüllung aller Restriktionen. Eine derartige Verbesserung hinsichtlich aller Kriterien konnte bisher lediglich unter Verwendung von approximierten Zielfunktionsauswertungen erzielt werden (vgl. Emmerich und Naujoks [17]). Die Streckung und die Verschiebung der Front hin zu besseren Ergebnisse bedeuten eine Zunahme des durch die Front überdeckten Hypervolumens. Es kann daher geschlossen werden, dass die gezeigten sehr guten Ergebnisse auf die Verwendung des Hypervolumens als Selektionskriterium zurückzuführen sind.

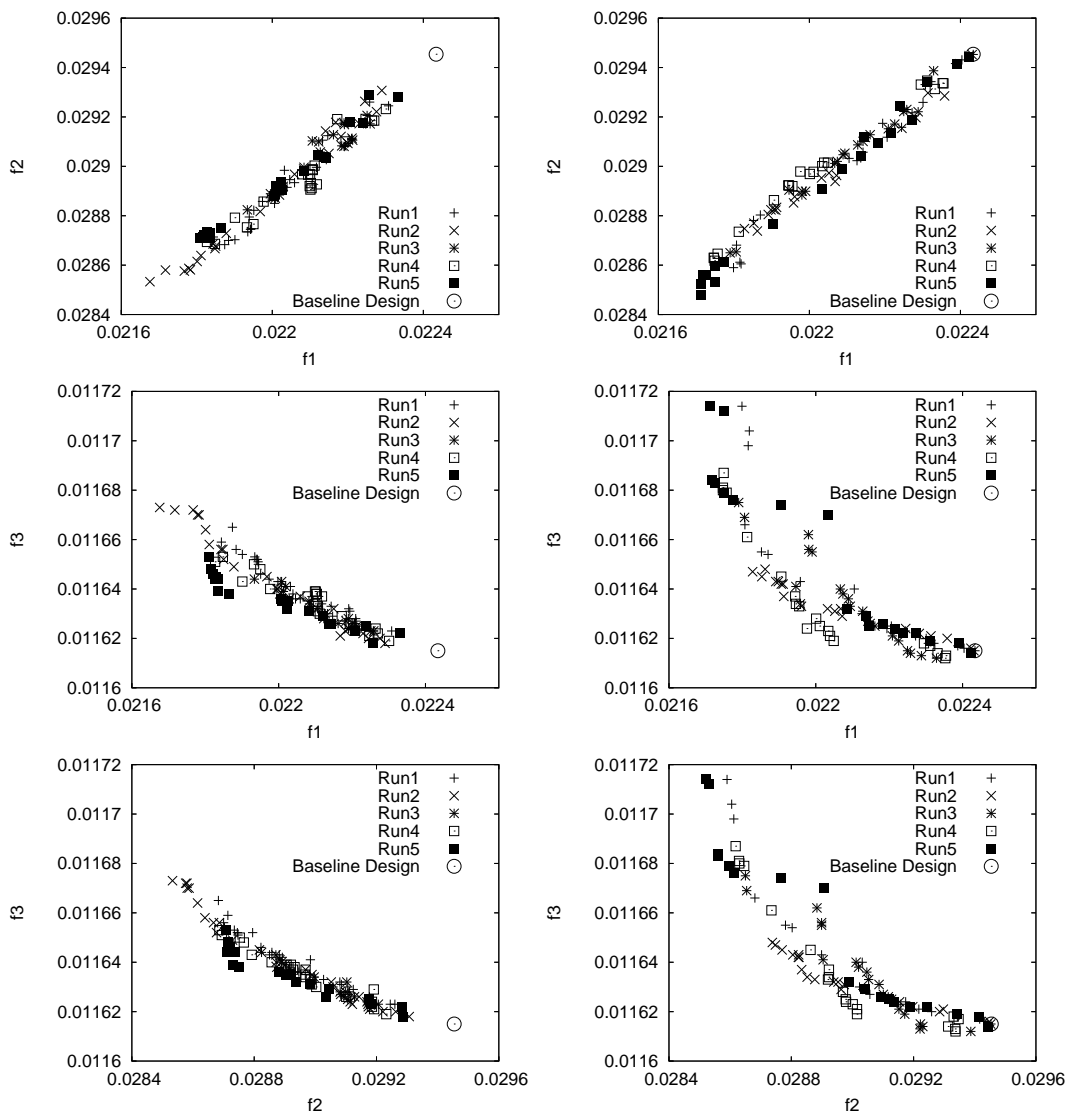


Bild 2: Projektionen der resultierenden Ergebnismengen für die Widerstandsminimierung der Referenztragfläche RAE 2822. Die aufgetragenen Funktionen sind die Widerstandsbeiwerte zu den jeweiligen Strömungsbedingungen aus Tabelle 1. Die Graphiken zeigen fünf Ergebnisse nach jeweils 1000 Zielfunktionsauswertungen mit dem NSGA-II (links) und dem SMS-EMOA (rechts).

4 Zusammenfassung und weitere Forschungsvorhaben

Vor den Studien des SMS-EMOA gab es keinen Pareto-basierten EMOA, der in hoch-dimensionalen Zielräumen annähernd so gut funktioniert wie bei zwei oder drei Zielen. Der SMS-EMOA ist derzeit der einzige Pareto-basierte EMOA, der auf hoch-dimensionalen Testfunktionen die Pareto-Front erreicht und mit einer relativ gleichmäßigen Verteilung repräsentativ abdeckt. Zudem stellte er seine Praxis-Tauglichkeit in prototypischen Optimierungen von z. B. Tragflügeln unter Beweis. Er ist anwenderfreundlich, da er außer den bei der evolutionären Optimierung üblichen Parametern keine weiteren benötigt.

Die Leistungsfähigkeit des SMS-EMOA basiert auf der Steuerung des Optimierprozesses durch die S-Metrik. Die Berechnung der S-Metrik ist in hoch-dimensionalen Zielfunktionsräumen rechenintensiv. Für zwei oder drei-kriterielle Funktionen ist die Laufzeit

mit der üblicher anderer EMOA vergleichbar. Für praktische Anwendungen mit hochdimensionalen Zielfunktionen ist der SMS-EMOA geeignet, wenn die Auswertung der Zielfunktion zeitaufwändig ist, was häufig aufgrund komplexer Simulationen der Fall ist.

Der SMS-EMOA soll zusätzlich beschleunigt werden, indem Informationen zur S-Metrik-Berechnung gespeichert und in den folgenden Generationen verwendet werden. Die S-Metrik-Beiträge sollen dadurch effizient aktualisiert werden. Zudem werden die theoretischen Eigenschaften des SMS-EMOA analysiert, insbesondere die Konvergenzgeschwindigkeit, sowie die Verteilung der Punkte im Zielraum und das Erreichen eines optimalen S-Metrik-Wertes. Durch ein tieferes Verständnis dieser Vorgänge können Empfehlungen ausgesprochen werden, wie die Parameter der evolutionären Optimierung beim SMS-EMOA in Abhängigkeit vom Anwendungsproblems zu wählen sind. Parameterstudien helfen, diese Vermutungen auch auf andere Verfahren zu übertragen. Weitere Anwendungsprobleme sollen Einblick in typische Herausforderungen geben, die SMS-EMOA zu bewältigen hat und die Identifikation typischer Optimierszenarien ermöglichen. Insbesondere werden kombinatorische Optimierprobleme betrachtet.

Demnächst wird der SMS-EMOA als Modul der Optimierungsumgebung PISA¹ [18] zur Verfügung stehen. Der Source Code des von Beume und Rudolph [9] entwickelten Algorithmus zur S-Metrik-Berechnung wird ebenfalls bald öffentlich zugänglich sein.

Danksagungen

Diese Arbeit wurde unterstützt von der *Deutsche Forschungsgemeinschaft (DFG)* als Teil des *Sonderforschungsbereichs „Computational Intelligence“ (SFB 531)* und im Rahmen des Projekts *„Ein Verfahren zur Optimierung von aus Mehrkomponenten bestehenden Schiffsantrieben“* (Schw 361/15-1).

Literatur

- [1] Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, UK. 2001.
- [2] Coello Coello, C. A.; Van Veldhuizen, D. A.; Lamont, G. B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic Publishers. 2002.
- [3] Zitzler, E.; Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Case Study. In: *Parallel Problem Solving from Nature (PPSN V)* (Eiben, A. E.; et al., Hg.), LNCS 1498, S. 292–301. Springer, Berlin. 1998.
- [4] Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C. M.; Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation* 7 (2003) 2, S. 117–132.

¹PISA - Platform and Programming Language Independent Interface for Search Algorithms, ETH Zürich (www.tik.ee.ethz.ch/pisa/)

- [5] Emmerich, M.; Beume, N.; Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: *Evolutionary Multi-Criterion Optimization: 3rd Int'l Conf. (EMO 2005)* (Coello Coello, C. A.; et al., Hg.), S. 62–76. Springer, Berlin. 2005.
- [6] Beume, N.; Naujoks, B.; Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* (2006). (Im Druck).
- [7] Bäck, T.; Fogel, D. B.; Michalewicz, Z. (Hg.): *Handbook of Evolutionary Computation*. Bristol, United Kingdom: IOP Publishing and Oxford University Press. 1997.
- [8] Beume, N.; Naujoks, B.; Wagner, T.: Pareto-based, Indicator-based, and Aggregation Methods in Many-objective Optimisation. Interner Bericht des Sonderforschungsbereichs 531 *Computational Intelligence*, Universität Dortmund. 2006.
- [9] Beume, N.; Rudolph, G.: Faster S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem. Interner Bericht des Sonderforschungsbereichs 531 *Computational Intelligence* CI–216/06, Universität Dortmund. 2006.
- [10] Zitzler, E.; Deb, K.; Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8 (2000) 2, S. 173–195.
- [11] Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E.: Scalable Multi-Objective Optimization Test Problems. In: *Proc. of the 2002 Congress on Evolutionary Computation (CEC 2002)*, Bd. 1, S. 825–830. IEEE Press, Piscataway NJ. 2002.
- [12] Deb, K.; Mohan, M.; Mishra, S.: Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. In: *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003* (Fonseca, C. M.; Fleming, P. J.; Zitzler, E.; Deb, K.; Thiele, L., Hg.), S. 222–236. Faro, Portugal: Springer. Lecture Notes in Computer Science. Volume 2632. 2003.
- [13] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation* 6 (2002) 2, S. 182–197.
- [14] Zitzler, E.; Laumanns, M.; Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: *Evolutionary Methods for Design, Optimisation, and Control*, S. 95–100. Barcelona, Spain: CIMNE. 2002.
- [15] Hughes, E. J.: Evolutionary Many-Objective Optimisation: Many Once or One Many? In: *Evolutionary Computation 'Congress (CEC'05), Edinburgh, UK*, Bd. 1, S. 222–227. IEEE Press, Piscataway NJ. 2005.
- [16] Naujoks, B.; Beume, N.; Emmerich, M.: Multi-objective optimisation using S-metric selection: Application to three-dimensional solution spaces. In: *Evolutionary Computation Congress (CEC'05), Edinburgh, UK*, Bd. 2, S. 1282–1289. Piscataway NJ: IEEE Press. 2005.
- [17] Emmerich, M.; Naujoks, B.: Metamodel-assisted multi-objective optimisation with implicit constraints and their application in airfoil design. In: *Int'l Conf. ERCOFTAC 2004*. CIMNE, Barcelona, Spain (CD-ROM). 2004.

- [18] Bleuler, S.; Laumanns, M.; Thiele, L.; Zitzler, E.: PISA — A Platform and Programming Language Independent Interface for Search Algorithms. In: *Evolutionary Multi-Criterion Optimization, 2nd Int'l Conf. (EMO 2003)* (Fonseca, C. M.; et al., Hg.), LNCS 2632, S. 494 – 508. Springer, Berlin. 2003.

GLEAM - Ein EA für Prozessabläufe am Beispiel von Steuerungen für Industrieroboter

Christian Blume

FH Köln, Campus Gummersbach
51643 Gummersbach, Am Sandberg 1
Tel. (02261) 8196-297 oder -330 oder -332
Fax (02261) 819615
E-Mail: blume@gm.fh-koeln.de

1 Einführung

Evolutionäre Algorithmen (EAs) werden zur Lösung von Planungs- und Optimierungsproblemen in einem weiten Rahmen eingesetzt [1], [2], der Einsatz bei Prozessabläufen im industriellen Bereich erfordert für die Anwendung in der Praxis jedoch die Entwicklung spezieller Konzepte. Daher wurde ein Modell zur Behandlung der zeitlichen Abhängigkeit von Prozessparametern bei Steuerungen entwickelt, dass die Anwendung eines Evolutionären Algorithmus' auf die Planung und Optimierung von industriellen Prozessabläufen erlaubt.

In diesem Zusammenhang wurde eine Codierung definiert, die einen Genetischen Code beschreibt, der zur zeitabhängigen Steuerung von Prozessabläufen dient. In der Praxis ist der genaue zeitliche Ablauf von Einstellungsänderungen verschiedener Prozessparameter die Grundlage für den Erfolg oder Misserfolg einer Prozessoptimierung. Außerdem erlaubt die Codedefinition zusammen mit den vordefinierten genetischen Operatoren eine flexible Abbildung unterschiedlichster Aufgabenstellungen auf eine festimplementierte evolutionäre Maschine ohne die Option zusätzlicher aufgabenbezogener Operatoren aufzugeben.

Basierend auf dem Zeitmodell wurde ein EA entworfen und implementiert sowie eine entsprechende Genetische Codierung definiert, die zur Lösung von Problemen in der industriellen Praxis eingesetzt werden. Die Implementierung wurde erfolgreich in eine Steuerung eines namhaften deutschen Roboterherstellers eingebunden. Sie dient der automatischen Erzeugung von Roboterbefehlen zur automatischen Programmierung einer kollisionsfreien, optimierten Bewegungsbahn eines Industrieroboters. Die so erzeugten Programme werden direkt von der Robotersteuerung ausgeführt.

1.1 Bewegungsprogrammierung und –steuerung von Industrierobotern

Die Steuerung eines Industrieroboters kann auf zwei Arten programmiert werden. Beim Teach-in-Verfahren wird der Roboter vom Programmierer per Handprogrammiergerät zu einzelnen Bewegungszielen gesteuert und die Zwischen- und Zielpositionen abgespeichert. (Daneben gibt es noch die Folgeprogrammierung, dabei wird der Industrieroboter gewissermaßen an die Hand genommen und die Bewegung „vorgemacht“. Dies spielt hier keine Rolle). Im anschließenden Automatikbetrieb, wenn das Programm von der Steuerung ausgeführt wird, werden in einem bestimmten Zeittakt (z.B. alle 15 ms) Sollwerte für Stützpunkte der Bewegungsbahn zur Zielposition berechnet. Dabei werden auch vom Programmierer vorgegebene Geschwindigkeits- und Geometrieparameter verarbeitet, siehe auch [3].

Bei der im vorliegenden Fall relevanten Offline-Programmierung gibt der Roboterprogrammierer textuell oder mit Hilfe eines Robotersimulationssystems die Zwischen- und Zielpositionen sowie andere Parameter der Bewegungsbahn des Roboters an [4]. In der Praxis wird meist eine Kombination aus Teach-in- und Offline-Programmierung eingesetzt [5]. Denn das Teach-in-Verfahren hat zwar den Vorteil, dass der Programmierer alle Positionen präzise mit dem Roboter anfahren kann, die Bewegungsbahn zwischen den Positionen ist damit aber nicht unbedingt für den Programmierer vorhersehbar. Vor allem aber ist der zeitliche Aufwand teilweise erheblich (Tage bis Wochen für ein Bewegungsprogramm), während dieser Zeit fällt der Industrieroboter für die Produktion aus.

Ein besonderes Problem stellt bei der Roboterprogrammierung die Erzeugung kollisionsfreier Bewegungsbahnen dar, um z.B. einen Tank passgenau in ein Auto einzusetzen. Hier kann man die Offline-Programmierung kaum einsetzen (der Programmierer kann die genaue Roboter-Konfiguration nicht ermitteln, d.h., wie bewegen sich die einzelnen Robotergelenke im zeitlichen Ablauf) und das Teach-in-Verfahren verschlingt kostbare Programmierzeit.

1.2 Der Evolutionäre Algorithmus GLEAM

Der Evolutionäre Algorithmus GLEAM (Global Learning Evolutionary Algorithm and Method) [6] wurde nicht nur, aber vor allem zur Lösung der Steuerungsoptimierung von Prozessabläufen entwickelt. Dabei diente die automatische Erzeugung von Roboterbewegungsbahnen und den zugehörigen Bewegungsprogrammen als eine Art Benchmark, da eine von GLEAM generierte und optimierte Bewegungsbahn vom Anwender leicht zu beurteilen ist. Er braucht nur die Roboterbewegung in der Simulation oder real anzuschauen bzw. die Bewegungszeit messen.

Als eine Art Seiteneffekt stellte sich dann heraus, dass sich auf diese Weise automatisch Programme für kollisionsfreie, optimierte Bewegungsbahnen bzw. zugehörige –programme von GLEAM erzeugen ließen. Es zeigte sich in diesem Zusammenhang, dass die Genetische Codierung des Problems und ein zeitbasiertes Steuerungsmodell eine besondere Rolle bei der Lösung des Problems spielen.

2 GLEAM-Konzept

Im Folgenden werden vor allem die Unterschiede bzw. neuen Konzepte von GLEAM im Vergleich zu anderen EAs behandelt, die Grundlagen eines EA werden nicht weiter behandelt. Wie bereits erwähnt, wurde GLEAM für die Steuerungsoptimierung bei Prozessabläufen entwickelt. Außerdem sollte GLEAM ein breites Anwendungsspektrum erlauben und neue, von der Biologie inspirierte Metastrukturen, enthalten. Im folgenden werden vor allem neue konzeptionelle Eigenschaften von GLEAM beschrieben, wobei auch ein „Modell“ oder eine neue Methode eingeführt wird, die einen neuartigen Umgang mit der Interpretation des Genetischen Codes zugrunde legt. Dies erfordert auch eine „freiere“ Gestaltung des Genetischen Codes, der in seiner Struktur und in seinem dynamischen Aufbau über die bisherige Festlegung auf reell-wertige Parameter hinausführt.

2.1 Genetischer Code von GLEAM

In der Biologie wird der genetische Code nicht nur als “Bauplan” im statischen Sinne verwendet, etwa um die Farbe der Haare zu definieren, sondern er steuert auch den

Start von Prozessen, z.B. die Festlegung des Zeitpunktes, wann andere Gene „eingeschaltet“ werden, oder der einer Hormonausschüttung. Dies geht über die rein statische Beschreibung durch einen „Bauplan“ hinaus, es wird ein Zeitmodell oder zumindest ein Zeitbezug zugrunde gelegt. Denn die durch den Genetischen Code definierten Lebewesen entwickeln sich im Ablauf der Zeit, indem sie sich in Größe und Struktur verändern. Da dieser Prozess ebenso wie der momentane Aufbau eines Lebewesens u.a. von seinen Genen gesteuert wird, ist es naheliegend, auch die Planung und Optimierung des zeitlichen Ablaufs technischer Prozesse wie die Bewegung eines Roboters mit Hilfe eines Genetischen Codes zu beschreiben.

Daher basiert der Evolutionäre Algorithmus GLEAM auf einem sog. Aktionsmodell. Es enthält nicht nur die Parameter zur Steuerung von technischen Prozessen und Abläufen, sondern enthält auch die Festlegung, zu welchem Zeitpunkt welche Prozesse begonnen oder beendet werden. Ein Prozess besteht in diesem Zusammenhang aus einer oder mehreren Aktionen, deren Parameter durch die Evolution verändert werden und deren Bedeutung sich durch die Implementierung einer Aktionssimulation mit einem entsprechendem Interpreter der Aktionen ergibt.

Neben dem Aktionsmodell enthält GLEAM ein weiteres neues Konzept, nämlich das der "Segmente". In der Biologie werden die Gene zusammengefasst zu Chromosomensträngen, außerdem sind bestimmte Genabschnitte in ihrer Gesamtheit für einen Vorgang zuständig, etwa das Wachstum. GLEAM enthält die Zusammenfassung von Aktionsfolgen des Gencodes zu sog. Segmenten. Diese Segmente spielen eine besondere Rolle bei Mutation und Rekombination, sie werden als eine Einheit, die eine Teillösung des Problems enthalten kann (und nach einer gewissen Evolutionszeit auch enthält), betrachtet, und diese Information als "Block" weitergegeben. Dies dient einer Beschleunigung der Evolution.

Die Umsetzung dieses neuen oder erweiterten Ansatzes (GLEAM basiert ansonsten auf den bekannten Konzepten, die Evolutionären Algorithmen zugrunde liegen, wie Ranking, Paarselektion, u.a.) erfolgte mit Hilfe von Konzepten der Informatik, wie strukturierten Datentypen oder dynamischen Listen. Diese sind erforderlich, da der Aufbau des Genetischen Codes in GLEAM nicht mehr einfache reellwertige Vektoren umfasst, sondern variabel aufgebaute und dynamisch veränderbare Strukturen. Denn es ist in der Regel nicht vorher bekannt, wie viele Aktionen zur Bewältigung eines Prozesses benötigt werden, etwa wie viele Bewegungsanweisungen für eine kollisionsfreie Bewegungsbahn eines Roboters notwendig sind.

Natürlich kann GLEAM auch für Probleme angewandt werden, die im obigen Sinne als statisch betrachtet werden können. So wurde mit GLEAM die Anordnung von Maschinenbelegungen für einen optimalen Produktionsplan erstellt, [7]. In diesem Fall ist auch die Anzahl der Aktionen (eine Aktion entspricht einem Bearbeitungsschritt) fest vorgegeben. Weitere Anwendungen betreffen die Ressourcenoptimierung in der Verfahrenstechnik [8], die Verfahrenwegoptimierung bei einer Palettenbestückung [9] sowie die Stapeloptimierung von Betonplatten [9], am Forschungszentrum Karlsruhe wurden Arbeiten zur Optimierung einer Aktuatorpumpe [10] und der Designoptimierung [11] auf GLEAM-Basis durchgeführt.

2.2 Struktur des Genetischen Codes von GLEAM

Der Genetische Code für ein Individuum besteht in GLEAM aus einer Anzahl von Segmenten (mindestens eins). Jedes Segment besteht aus einer Anzahl von Aktionen (mindestens eine). Ein Aktionscode setzt sich zusammen aus der Aktionskennung (eine

Zahl), der je nach Aktionstyp eine gewisse Anzahl von Parametern folgen. Es gibt auch Meta-Aktionen, welche den Prozessablauf steuern, die keine Parameter enthalten. Ein Parameter kann vom Typ Integer oder Real sein (ein Zeichen, also in C Typ char, ist so betrachtet natürlich auch ein Integer).

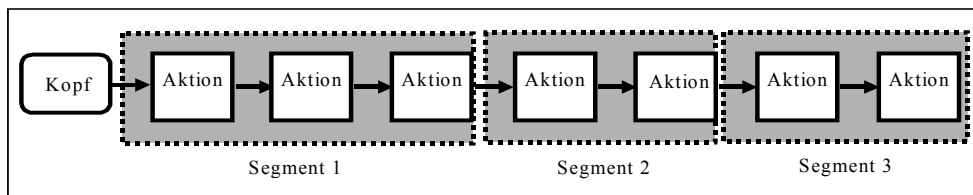
Eine Population M von μ Individuen (d.h. zu einer Aktionskette aneinandergereihte Aktionen) wird formal folgendermaßen definiert:

$$\begin{aligned}
 M &= \{ c_1, c_2, \dots, c_\mu \} && \text{mit } \mu \in \mathbb{N}, \text{ i.d.R. ist } \mu > 2 \\
 c_i &= (V, S_1, S_2, \dots, S_r) && \text{mit } V = (l_1, l_2, \dots, l_r), l_i \in \mathbb{N} \\
 l_i &= k \text{ oder } p && \text{mit } k, p \in \mathbb{N}, p \text{ ist Adresse bzw. Zeiger} \\
 S_i &= (a_{i1}, a_{i2}, \dots, a_{im}) && \text{mit } i, m \in \mathbb{N} \\
 a_{ik} &= (ak) \text{ oder } (ak, p_1, p_2, \dots, p_z) && \text{mit } ak, z \in \mathbb{N} \text{ und } p_i \in \mathbb{N} \vee \mathbb{R}
 \end{aligned}$$

Die Bedeutung der einzelnen Elemente ist:

- c_i : Aktionskette, d.h. ein Individuum
- V : Verwaltungsinformation für ein Individuum
- l_i : Kennzahlen oder Adressen (Zeiger), z.B. Länge der Segmente
- S_i : Segment, bestehend aus einer Folge von Aktionen
- a_{ik} : Aktion
- ak : Aktionskennung (Integer)
- p_i : Parameter einer Aktion (Integer oder Real)

Bildlich kann man eine Aktionskette entsprechend ihrer Implementierung darstellen als eine dynamisch verzeigerte Liste:



Der Listenkopf enthält Verwaltungsinformationen, wie Länge oder Fitness der Aktionskette. Die Aktionen bestehen aus Verbunden von Daten mit der entsprechenden Aktionskennung sowie ihren Parametern, der Speicherplatz für eine Aktion wird dynamisch nach Bedarf erzeugt und wieder gelöscht.

GLEAM enthält als völlig neues Strukturelement für den Genetischen Code eine Segmentierung. Dies bedeutet, dass die Aktionsketten in Abschnitte, eben diese Segmente unterteilt werden, über die auch bestimmte Genetische Operatoren, wie Segmente löschen/einfügen/vertauschen oder in der Aktionsfolge invertieren, ausgeführt werden. Wenn die Startpopulation durch Auswürfeln initialisiert wird, werden die Segmente willkürlich gesetzt. Sollte schon ein Vorwissen vorhanden sein, sollte man sinnvolle Teilbewegungen bzw. die entsprechenden Aktionen dazu in Segmente einteilen.

Nach einer gewissen Evolutionszeit bilden sich auch bei anfangs willkürlicher Einteilung Segmente heraus, die verschiedene Teillösungen enthalten. Da die Rekombination über die Segmente und nicht wie sonst durch willkürliche Unterteilung des Codes erfolgt, steigt die Wahrscheinlichkeit, dass durch die Kombination zweier guter Teillösungen schneller eine sehr gute Gesamtlösung eines Problems entsteht.

Die Segmentierung hat ihr entferntes Vorbild in der Biologie durch die Zusammenfassung von Genen zu Sequenzen und Chromosomen. Auch in diesem Fall dient die

Einteilung in Unterstrukturen der Beschreibung von Teillösungen bzw. Einheiten im Bauplan von Lebewesen.

3 Aktionsmodell

Nach dem Erzeugen einer Aktionskette durch die Evolution (Initialisieren per Zufall, Anwenden von Rekombination und Mutation) wird sie zum Feststellen ihrer Fitness „ausgeführt“. Dieses „Ausführen“ bedeutet im statischen Fall, dass eine Bewertungsfunktion angewandt wird, die entsprechend den Parameterangaben im Genetischen Code eine Formel auswertet. Bei GLEAM wird grundsätzlich von einer Aktion ausgegangen, d.h. der Aktionscode wird durch einen Interpreter ausgewertet und entsprechende (physikalische bzw. simulierte) Aktionen zu einem bestimmten Zeitpunkt im Prozessablauf angestoßen oder beendet. Während dieser Aktionsausführung werden Daten für die Berechnung der Fitness gesammelt und am Ende der Ausführung aller Aktionen des Genetischen Codes eine Fitness für diese Aktionskette ermittelt.

Der Simulator kann also als ein Interpreter des Aktionscodes betrachtet werden, wobei auch der zeitliche Ablauf der Aktionsausführung eine Rolle spielt und modelliert wird. Sollte die Zeitdauer einer Aktion keine Rolle spielen und beispielsweise nur die Anordnung der Aktionen von Bedeutung sein, kann dies natürlich ebenso behandelt werden, der Simulator berücksichtigt bzw. benötigt dann eben keine Zeitangaben. Dies wäre z.B. bei der Optimierung einer Anordnung von Schnittmustern auf einer Fläche der Fall. Eine „Aktion“ wäre in diesem Fall etwa „Ordne Teil A an“, dann könnte die Aktion „Ordne Teil C an“ folgen, wobei keine Zeitbetrachtung notwendig ist. Wichtig ist, dass die jeweilige Implementierung eines Moduls „Simulator“ zu dem definierten Genetischen Code passt.

3.1 Definition des Aktionsmodells

Der Zeitbezug wird in GLEAM durch die Definition eines Zeittaktes ähnlich wie bei einer Robotersteuerung erreicht. Die Dauer des Zeittaktes kann vom Anwender frei festgelegt werden. Für die Simulation einer Aktionskette (mit Zeittakt) gilt dann folgendes: Die erste Aktion einer Aktionskette startet zum Zeitpunkt t_0 . Die nächste Aktion startet nach einer (simulierten) Zeitdauer von einem Zeittakt Δt , usw. Daraus ergibt sich folgender Bezug:

Aktion a_1	t_0
Aktion a_2	$t_0 + \Delta t$
.....
Aktion a_i	$t_0 + (i-1)*\Delta t$

Es gibt zwei wichtige Elemente im Genetischen Code von GLEAM, die für die Zeitsteuerung zuständig sind:

Die Aktionen von GLEAM enthalten die beiden nicht spezifisch anwendungsbezogenen Aktionstypen BLOCK_BEGIN und BLOCK_END. Alle Aktionen, die in der Aktionskette zwischen BLOCK_BEGIN und BLOCK_END vorkommen, starten zum gleichen Zeitpunkt t_n , erst die Aktion nach der BLOCK_END-Aktion startet zum nächsten Zeitpunkt $t_n + \Delta t$:

Aktion a_i : BLOCK_BEGIN	t_n
Aktion a_{i+1}	t_n

Aktion a_{i+2}	t_n
.....	
Aktion a_{i+m}	t_n
Aktion a_{i+m+1} : BLOCK_END	t_n
Aktion a_{i+m+2} :	$t_n + \Delta t$

Die ebenfalls nicht anwendungsbezogene Aktion UNCHANGED enthält einen Parameter n vom Typ Integer. Sie bewirkt, dass die nächsten n Zeittakte keine neue Aktion beginnt, die folgende Aktion in der Aktionskette nach UNCHANGED beginnt also nach n Zyklen:

action a_i	t_i
action a_{i+1} : UNCHANGED n	$t_i + \Delta t$
action a_{i+2} :	$t_i + n * \Delta t$

Mit den Aktionen BLOCK_BEGIN und BLOCK_END sowie UNCHANGED können beliebige Abläufe und Zeitpunkte zum Starten und Beenden von Aktionen modelliert werden.

3.2 Aktionsmodell für Programme zur Roboterbewegung

Die Anwendung von GLEAM auf die Robotertechnik erfolgte durch Definition von primitiven Achsbewegungen des Roboters als Aktionen einer Aktionskette, welche dann die gesamte Bewegung (oder mehrere Bewegungen) beschreibt. Ein derartige einfache Aktion bzw. Bewegungsbefehl lautet im Klartext: „*Bewege Achse 3, indem die Achse mit einer Beschleunigung von 48 Grad/Sekunde² auf eine Geschwindigkeit von 12 Grad/Sekunde gebracht wird*“. Anschließend wird eine UNCHANGED-Aktion ausgeführt, wobei 10 Takte lang keine neue Aktion begonnen, jedoch die Achse 3 weiter bewegt wird.. Beendet wird diese Bewegung durch eine Stop-Aktion oder durch das Ende der Aktionskette (sog. Auslaufen der Bewegung). Die zugehörige Stop-Aktion (oder Stop-Befehl) lautet beispielsweise „*Stop Achse 3 mit einer Verzögerung von 17 Grad/ Sekunde²*“. Formal würde als Genetischer Code angegeben:

1003, 48.0, 12.0 (*Kennung 1003 für Bewegung Achse 3, dann folgen die Parameter*)
 1015, 10 (*Kennung 1015 für UNCHANGED, dann folgt der Parameter*)
 1009, 17.0 (*Kennung 1009 für Stop Achse 3, dann folgt der Parameter*)

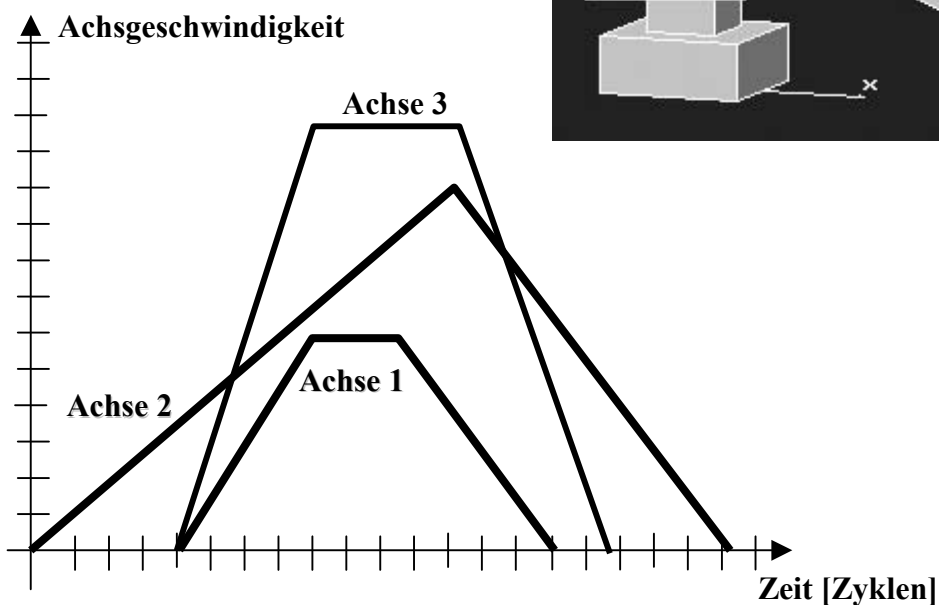
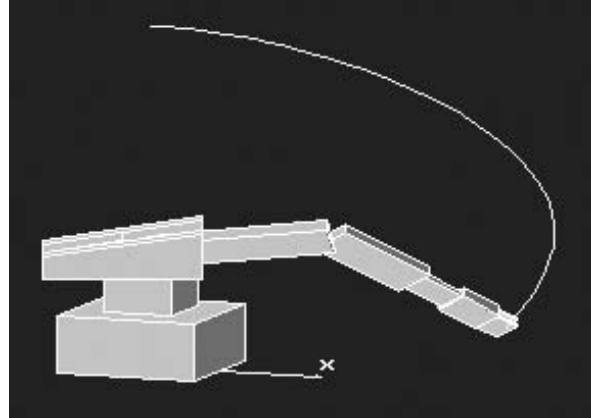
Die Evolution hat jetzt z.B. eine Aktionskette mit diesen Aktionen erzeugt, sie werden dann gemäß dem folgendem Zeitablauf ausgeführt.

Zeitpunkt 0 sei der Bewegungsstart, d.h. der Simulator führt die Aktion „Bewege Achse 3“ aus. Nach einem Zeittakt, z.B. nach 100 ms, führt der Simulator die nächste Aktion aus, also 10 Zeittakte oder 1 s lang keine neue Aktion beginnen. Allerdings wird weiter das Anfahren der Achse 3 mit einer Beschleunigung von 48 Grad/Sekunde² ausgeführt. Die Achse 3 wird nun solange beschleunigt, bis nach 250 ms die vorgegebene Geschwindigkeit von 12 Grad/Sekunde erreicht wurde. Dann behält die Achse die Geschwindigkeit von 12 Grad/Sekunde bei, bis nach 1 s oder 10 Zeittakten die Stop-Aktion begonnen wird (vom Start nach 1.1 s). Die Bewegung der Achse wird dann mit einer Verzögerung von 17 Grad/Sekunde² beendet.

Wenn mehrere Achsen gleichzeitig bewegt und in ihrer Geschwindigkeit verändert werden, können Roboterbewegungen mit beliebiger Dynamik und Geometrie erzeugt werden. Das folgende Beispiel des Genetischen Codes umfasst die Steuerung der ersten drei Achsen des Mitsubishi-Roboters RV-M1:

1. : Start_Motor 2, Rampe = 1°/s², Geschwindigkeit = 10°/s

- 2.: Unveraendert: Takt-Anzahl = 15
- 3.: Block_Beginn
- 4.: Start_Motor 1, Rampe = $2^\circ/s^2$, Geschwindigkeit = $5^\circ/s$
- 5.: Start_Motor 3, Rampe = $4^\circ/s^2$, Geschwindigkeit = $10^\circ/s$
- 6.: Block_End
- 7.: Unveraendert: Takt_Anzahl = 20
- 8.: Motor_aus 1, Rampe = $2^\circ/s^2$
- 9.: Unveraendert Takt-Anzahl = 5
- 10.: Block_Beginn
- 11.: Motor_aus 3, Rampe = $5^\circ/s^2$
- 12.: Motor_aus 2, Rampe = $2^\circ/s^2$
- 13.: Block_End
- 14.:Unveraendert: Takt-Anzahl = 20

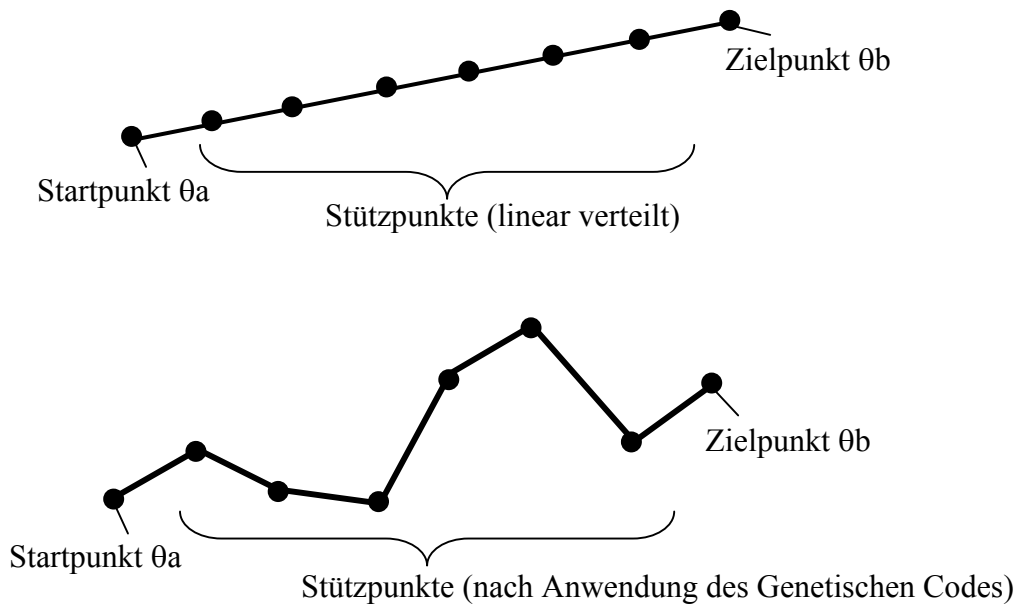


Die resultierende Roboterbewegung ist auf dem Bild mit der Mitsubishi-Robotersimulation zu sehen. Die Bewegungsbahn des Greifers des Mitsubishi-Roboters (nur sie ist für den Anwender von Interesse) ist durch eine Linie gekennzeichnet. Die Dynamik, d.h. die Geschwindigkeitsänderungen sind so leider nicht zu sehen, sie werden durch die obigen Geschwindigkeitsprofile angedeutet.

3.3 Alternativer Genetischer Code ohne direkten Zeitbezug

Die obige Definition von Aktionen als Bewegungsprimitive basiert auf dem Aktionsmodell mit seinem Zeitbezug. Zum Vergleich sei hier ein Genetischer Code angegeben, der jedoch ohne direkten Zeitbezug nur auf geometrischen Überlegungen basiert.

Die zurückzulegende Strecke einer Roboterachse ist durch den Startwert θ_a und dem Zielwert θ_b definiert. Es werden nun eine bestimmte Anzahl von Stützpunkten erzeugt, zu denen zunächst die lineare Änderung der Achsenwerte führt, also ein Betrag von Gesamtstrecke der Achse geteilt durch die Stützpunktzahl. Der Genetische Code beschreibt nun die Anzahl, den Betrag (in Prozent) und „Änderungsgeschwindigkeit“ der Abweichung über die betroffenen Stützpunkte. Die folgende Abbildung zeigt dies für eine Roboterachse.



Durch die Anwendung des Genetischen Codes entsteht ein neues Bewegungsprofil, das zwar auch von der Start- zur Zielposition führt, jedoch willkürliche Ausweichbewegungen umfasst. Experimente mit diesem Genetischen Code ergaben, dass damit selbst bei langen Evolutionszeiten sich keine kollisionsfreien Bewegungsbahnen generieren ließen (bei nicht trivialer Aufgabenstellung).

Es wurden noch geometrische Codierungen der Änderung der Bewegungsbahn des Roboters im dreidimensionalen Raum getestet, das Ergebnis war jedoch ebenso wenig erfolgversprechend. Wenn auch nicht durch längere Testreihen belegt, zeigen diese ersten Experimente ähnlich wie bei Arbeiten anderer Autoren (siehe auch [12], [13], [14]), dass eine rein geometrisch orientierte Genetische Codierung ohne Zeitmodell kaum Chancen auf eine akzeptable Problemlösung hat.

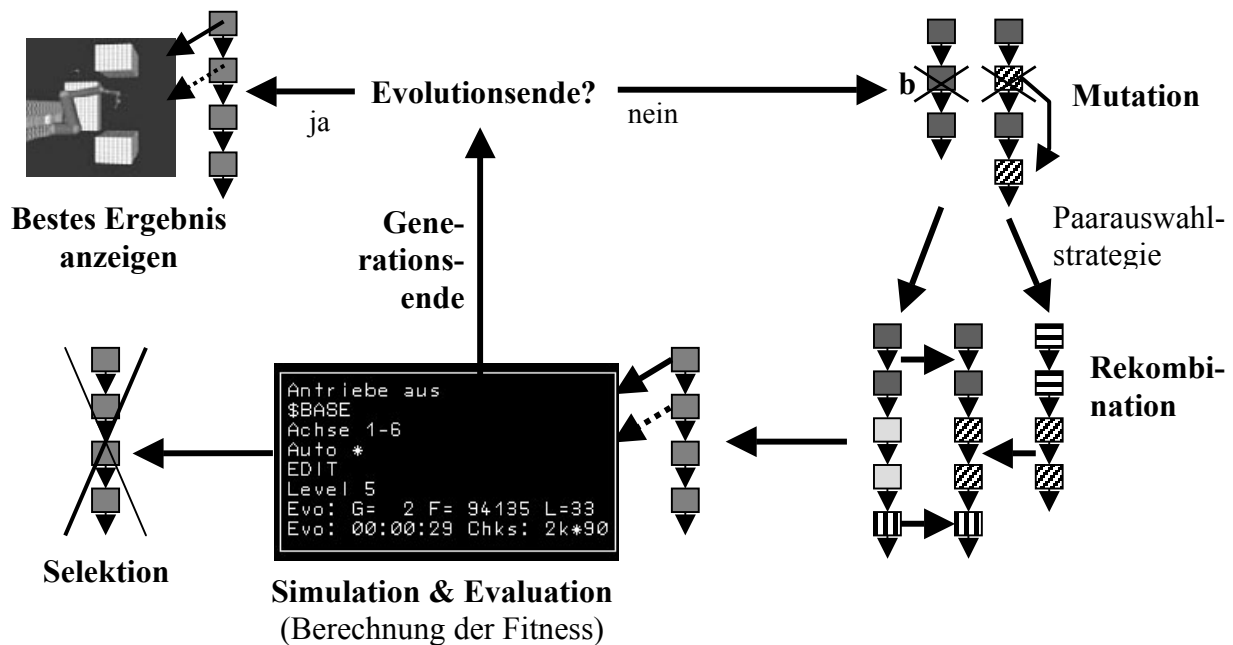
3.4 Aktions- und Aktionskettentypen

Die Implementierung des Aktionskonzepts von GLEAM erfolgte mit Hilfe der Definition abstrakter Datentypen. Daher kann eine Aktion eine beliebige Folge von Parametern der Datentypen Real und/oder Integer umfassen, wichtig ist nur, dass auch der Simulator diese Parameter entsprechend auswertet. Da mit einem Integerwert auch ein Zeichen (Typ char) oder ein Typ Bool dargestellt werden kann, sind diese Typen bzw. ihre Werte auch als Aktionsparameter möglich. Außerdem wird in GLEAM der Bereich eines Aktionsparameters vom Anwender festgelegt, damit die Evolution Parameterwerte nur über diesen Bereich variiert.

Ansonsten gibt es keine Einschränkungen (außer der Speichergröße) für die Definition von Aktionen. Der Anwender kann entsprechend dem zu lösenden Problem beliebige Aktionen definieren (er muss natürlich auch einen entsprechenden Interpreter für die Simulation implementieren). Die Gesamtheit aller definierten Aktionen und ihre semantische Bedeutung für eine Anwendung von GLEAM wird Aktionsmodell genannt. Der Genetische Code umfasst die formale bzw. syntaktische Beschreibung der Aktionen.

4 Genetische Operatoren und Evolutionsablauf im Überblick

Die hier vorgestellte Definition der Genetischen Codierung der Aktionsketten erlaubt die Formulierung eines Satzes neutraler genetischer Operatoren, die unter Einhaltung der bei der Codedefinition hinterlegten Informationen (z.B. Wertebereichsgrenzen von Parametern) an die Aufgabenstellung angepasste Mutationen durchführen können. Außerdem kann durch geeignete Zusammenfassung von Parametern in einer Aktion erreicht werden, dass sie nicht durch Rekombinationsoperatoren aufgespaltet werden können. Zusätzlich können anwendungsspezifische Operatoren implementiert werden, da die Aktionsketten in Verbindung mit dem Aktionsmodell genügend problembezogene phänotypische Informationen beinhalten. Der Evolutionsablauf entspricht dem Schema Evolutionärer Algorithmen, siehe Abbildung.



Alle Individuen einer Population werden der Reihe nach zum Elter und die Auswahl des Partners für das Crossover erfolgt durch rangbasierte Selektion. Bei ihr werden die zur Selektion anstehenden n Individuen entsprechend ihrer Fitness sortiert und erhalten dann gemäß ihrem Rang eine feste Selektionswahrscheinlichkeit zugeordnet: Die Selektion des Partners erfolgt innerhalb einer vorgegebenen Nachbarschaft des Elter in einem Deme.

Die Gesamtpopulation zerfällt nämlich in mehr oder weniger getrennte Teilpopulationen, den Demes. Dieses Nachbarschaftsmodell geht auf Gorges-Schleuter [15] zurück und verwendet eine ringförmige Topologie, auf der die Individuen linear angeordnet sind. Jedes Individuum hat eine gleichgroße Nachbarschaft zur rechten und zur linken, seinen Deme. Da die Nachbarschaften benachbarter Individuen sich überlappen, findet trotz der Isolation der Reproduktion auf das Deme ein mehr oder weniger langsamer Informationsaustausch über die Demegrenzen hinweg statt. Die Geschwindigkeit des Informationsflusses innerhalb der Gesamtpopulation und damit des Verhältnisses von *exploration* zu *exploitation* kann damit auch über die Demegröße gesteuert werden [16].

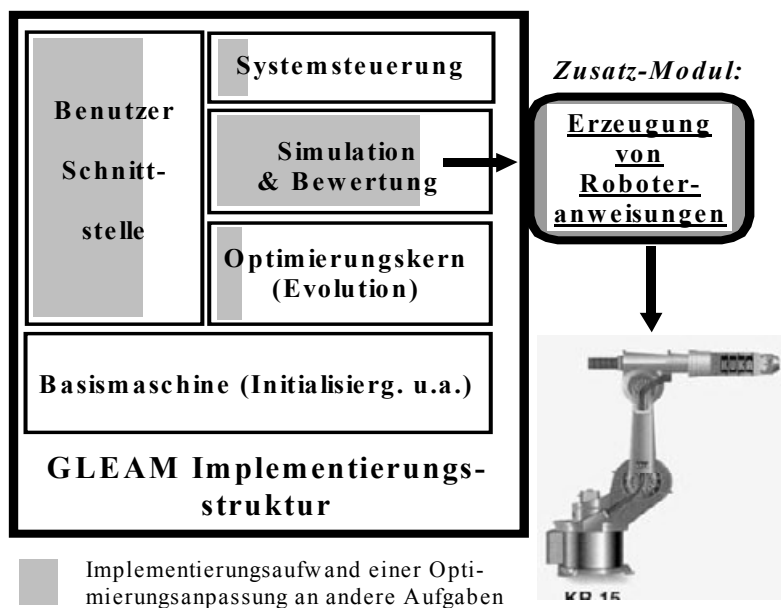
Die Wahrscheinlichkeit, ob und wie oft ein Genetischer Operator auf eine Aktionskette angewandt wird, kann in Abhängigkeit von der Länge der Aktionskette und der Fitness

des Elter durch eine GLEAM-Steuerdatei für die Evolution parametrisiert werden. Darüberhinaus können beliebige Mutationen zu einer hintereinander auf den gleichen Offspring angewandten Operatorsequenz zusammengefasst werden, wobei jedem Operator eine eigene Ausführungswahrscheinlichkeit zugeordnet wird. Jede Sequenz erzeugt einen (bei Mutationen) oder zwei Offsprings (beim Crossover). Damit entstehen pro Paarung in der Regel mehr als ein Offspring. Nur der beste Nachkomme einer Paarung konkurriert mit seinem Elter gemäß alternativen Akzeptanzregeln um den Fortbestand in der nächsten Generation:

5 Erzeugung von Roboterprogrammen

Für industrielle Roboteranwendungen müssen vor allem beim Punktschweißen, Handhaben und in der Montage Bewegungsbahnen erzeugt werden, die zwischen Hindernissen, wie Werkstücke, Magazine oder Anlagenaufbauten, zum Bewegungsziel führen. Als Ergebnis der GLEAM-Software soll ein fertiges Roboterprogramm bzw. Programmabschnitt ausgegeben werden, das direkt auf die Robotersteuerung geladen und dort ausgeführt werden kann.

Dazu ist es notwendig, die (internen) allgemeinen Bewegungsbeschreibungen für das Evolutionsverfahren in konkrete Bewegungsbefehle einer Roboterprogrammiersprache umzusetzen. Dies wird bisher durch eine Art Postprozessor vorgenommen, der die Bewegung des Roboters während der Simulation „abtastet“ und daraus ein entsprechendes Roboterprogramm erzeugt, siehe Abbildung der GLEAM-Implementierungsstruktur. Derartige Postprozessoren wurden für mehrere Industrieroboter von ABB, KUKA, REIS und Mitsubishi realisiert [17], [18].



So besteht beispielsweise die derart generierte Befehlsfolge für den IRB 2400-Roboter von ABB aus einem Datendefinitionsteil, wo die Zwischenpositionen der Bewegung definiert werden (`jointtarget Pi`), und den Bewegungsbeehlen zum Ausführen der kompletten Bewegung mit Überschleifen zwischen den einzelnen Bewegungsbefehlen zu den Zwischenpunkten (`MoveAbsJ Pi`).

Die folgenden Bilder zeigen als Beispiel das generierte Roboterprogramm in der ABB-Roboterprogrammiersprache RAPID und die kollisionsfreien Roboterbewegungen in einem Hindernisraum, die von GLEAM erzeugt wurden.

```

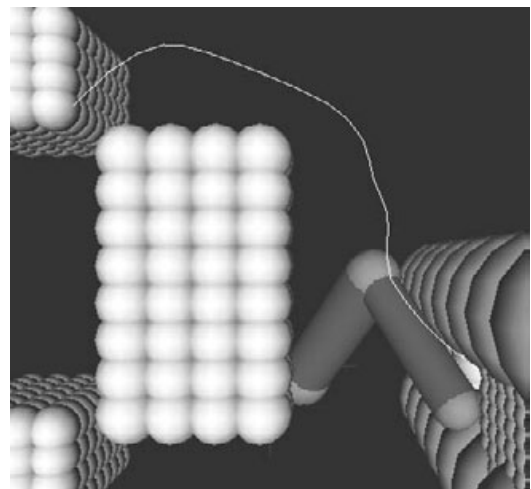
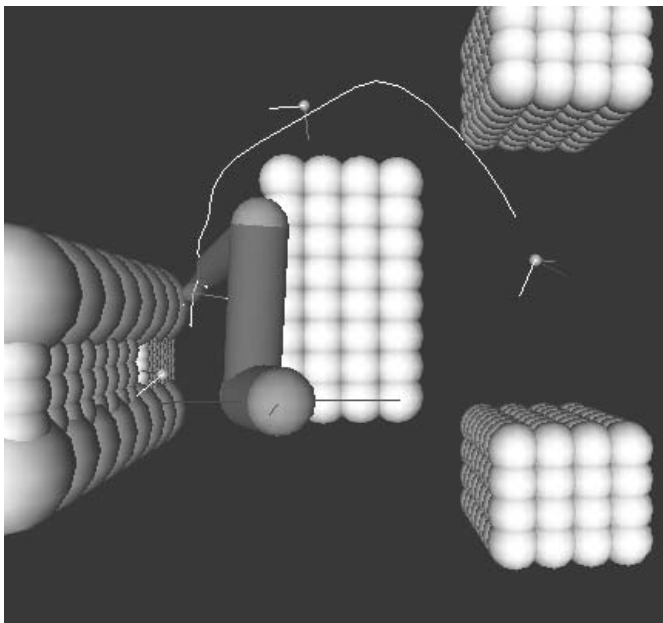
! COMMENT   Position list with
! COMMENT   the move position for every cycle (representing the trajectory):

MODULE demol
  VAR robtargt target:= [[-1647.0,-604.2,1374.7],[0.12367,0.69621,0.12367,
                        -0.69621],[-2,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST jointtargt P1:= [[0.5,45.3,45.3,0.0,-45.3,0.0], [9E9, ... ,9E9]];
  CONST jointtargt P2:= [[0.7,45.3,45.3,0.0,-45.3,0.0], [9E9, ... ,9E9]];
  CONST jointtargt P3:= [[0.7,45.3,45.3,0.0,-44.6,0.0], [9E9, ... ,9E9]];
  .....
  CONST jointtargt P74:= [[-90.7,45.2,0.6,96.9,4.1,-97], [9E9, ... ,9E9]];

! COMMENT   Move statements for the RAPID program:

PROC main()
  MoveAbsJ P1,v400,z20,toolx;
  MoveAbsJ P2,v400,z20,toolx;
  MoveAbsJ P3,v400,z20,toolx;
  .....
  MoveAbsJ P74,v400,z20,toolx;
  MoveL target,v100,z20,toolx;
ENDPROC
ENDMODULE

```



6 Implementierung auf einer industriellen Robotersteuerung

Zum Nachweis der bisherigen Überlegungen und Implementierungen wurde die GLEAM-Software auf der Steuerung eines namhaften deutschen Roboterherstellers implementiert. Dies bedeutete einen erheblichen Anpassungsaufwand, der vor allem durch die spezifischen Gegebenheiten der Steuerungshardware und -software verursacht wurde.



So fiel beispielsweise die komplette Grafik sowie Dateioperationen weg (die sind auf der Steuerung nicht möglich). Außerdem musste die gesamte Bedienoberfläche von GLEAM abgeschaltet werden, da die Bedienung über die Bedienoberfläche der Robotersteuerung erfolgt. Inhaltliche Änderungen am Programmcode verursachte auch die Tatsache, dass GLEAM nun ein Steuerungsmodul und kein eigenständiges Programm mehr ist. Was jedoch keine Probleme und kaum Aufwand hervorrief, war die Umstellung von Borland C auf Visual C, da GLEAM selbst fast nur in ANSI-C geschrieben wurde.

Eine besonders enge Zusammenarbeit und Abstimmung mit dem Roboterhersteller erforderte das Ausblenden der bisherigen Erfassung von Simulationsergebnissen zur Bewertung der Population in GLEAM und deren Erfassung und Berechnung in der Steuerungssoftware. In diesem Zusammenhang wurde eine neue Schnitt-

stelle zur Robotersimulation definiert und implementiert.

Da der Roboterhersteller nur Positionsangaben benötigt (Geschwindigkeits- und Beschleunigungsvorgaben erzeugt die Robotersteuerung intern), wurde zunächst ein Genetischer Codierung definiert, der die lineare Strecke Start – Ziel willkürlich abwandelt, siehe 3.3. Dies führte jedoch zu keinem vertretbaren Resultat, d.h. die Evolutionszeiten für einfache Beispiele waren schon äußerst hoch, für die geforderten komplexen Hindernisszenarien führten sie zu keinem Ergebnis. Daher wurde auf das bewährte Aktionsmodell zurückgegriffen und sein Ergebnis entsprechend den Anforderungen des Roboterherstellers angepasst.

Als konkrete Anwendungen wurden Beispielszenarien für drei kinematisch verschiedene Robotertypen realisiert. Denn bedingt durch die unterschiedliche Simulation (und damit Bewertung) sowie eine (hier nicht behandelte) Heuristik ist für jeden Robotertyp eine gewisse Anpassung notwendig.

7 Ergebnisse und Ausblick

Die bisherigen Ergebnisse der Implementierung auf einer industriellen Robotersteuerung lassen sich wie folgt charakterisieren:

- Gelungene Software-Integration von GLEAM in die Robotersteuerung durch beiderseitigen modularen Aufbau und klare Schnittstellen
- Nachweis der Funktionstüchtigkeit von GLEAM als Laborprototyp
- Einbau sog. Heuristiken, z.B. Suchbewegungen, die zu einer erheblichen Verkürzung der Evolutionszeiten führten
- Erzeugung von Roboter-Bewegungsprogrammen zur Simulation und zur Ausführung auf der Robotersteuerung innerhalb eines akzeptablen Zeitrahmens

Natürlich bedeutet die bisherige Implementierungsarbeit noch nicht, dass GLEAM bereits vollständig tauglich wäre für „den Verkauf von der Stange“. Es sind sicher

einige Arbeiten notwendig, um für Forschungsarbeiten nicht notwendigen Bedienungs- und Ausführungskomfort sowie –sicherheit zu erreichen. Die folgenden Arbeiten sind daher vor allem beim Roboterhersteller ins Auge gefasst worden:

- Einsatzprüfung der gesamten Software beim Roboterhersteller
- Optimierung der Simulation des Roboterherstellers (benötigt bisher ca. 90% der gesamten Rechenzeit)
- Feintuning der Evolutionsparameter, Verbesserung der Bedienungsoberfläche in der Robotersteuerung
- Weiterer Test von GLEAM und Ausbau der Heuristiken (Forschungsthema)

Die im letzten Punkt angesprochenen Heuristiken beinhalten sowohl die künstliche Erzeugung von Aktionsketten bei der Generierung einer Anfangspopulation als auch die Veränderung einzelner Aktionsketten (Individuen) durch Einfügen von Suchbewegungen während der Evolution. Diese Forschungsarbeiten ergänzen auf Anwendungsebene die im Forschungszentrum Karlsruhe von Jakob vorgenommene generelle Weiterentwicklung von GLEAM zu einem allgemein anwendbaren hybriden Verfahren HyGLEAM [19], [20] und [21].

Neben dem Einsatz von GLEAM für Industrieroboter kann die Software durch Parameter in Dateien relativ schnell auf neue Anwendungen angewandt werden, die sich vor allem auf die Planung und Optimierung von Prozessabläufen beziehen.

8 Literatur

- [1] Pohlheim, H.: *Evolutionäre Algorithmen*, Springer Verlag, Berlin, 1999
- [2] Hafner, S. (Hrsg.): *Industrielle Anwendungen Evolutionärer Algorithmen*, R. Oldenbourg Verlag, München, Wien, 1998
- [3] Blume, C.: *Allgemeine Methoden der Roboterprogrammierung*, in Messen, Steuern Regeln MSR, Jahrgang 1991 (Ergänzung), Interest-Verlag, Kissing
- [4] Blume, C., Jakob, W.: *Programming Languages for Industrial Robots*, Springer-Verlag, Berlin, Heidelberg, 1986
- [5] Blume, C., Früauf, P.: *Standardization of Programming Methods and languages for Manipulating Industrial Robots*, In 27th International Symposium on Industrial Robots, Oktober 1996, Mailand
- [6] Blume, C.: *GLEAM – A System for Intuitive Learning*. In Schwefel, H.P., Männer, R. (eds): *Conf. Proc. of PPSN I. LNCS 496*, Springer Verlag, Berlin, 1990, S. 48-54
- [7] Blume, C.: *Planning and Optimization of Scheduling in Industrial Production by Genetic Algorithms and Evolutionary Strategy*. In Proc. of the Second Biennial European Joint Conference on Engineering Systems Design and Analysis (ESDA), July 4-7, 1994, London, England
- [8] Blume, C., Jakob, W., Kaltwasser, J.: *Kosten senken durch verbesserte Nutzung der Ressourcen*. Im Tagungsband der ASIM - Fachausschuß 4.5 "Simulation" der Gesellschaft für Informatik, 10.-13. Oktober 1994, Stuttgart
- [9] Blume, C.: *Optimization in Concrete Precasting Plants by Evolutionary Computation*. in Proc. of the 2000 Genetic and Evolutionary Computation Conference GECCO 2000, 8th July 2000, Las Vegas Nevada, USA, pp 43 - 50

- [10] Jakob, W.; Quinte, A.; Scherer, K.P.; Eggert, H. *Optimization of a micro fluidic component using a parallel evolutionary algorithm and simulation based on discrete element methods*. In Hernandez, S. [Hrsg.] *Computer Aided Optimum Design of Structures VII : OPTI 2001 ; 6th Internat.Conf.*, Bologna, I, May 28-30, 2001 Southampton [u.a.] : WIT Press, 2001 S.337-46
- [11] Jakob, W.; Peters, D. *Designoptimierung mit HyGADO*. In Laur, R. [Hrsg.] *Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen : Statusseminar zum Verbundprojekt OMID*, Bremen, 9.-10.Oktober 2001 Tagungsband S.15-24 Bremen : Universität Bremen, 2001
- [12] Dai, F.: *Collision-Free Motion of an Articulated Kinematic Chain in a Dynamic Environment*, IEEE Computer Graphics & Application, January 1989, pp. 70-74
- [13] Heine, R., Schnare, T.: *Kollisionsfreie Bahnplanung für Roboter*. In *Robotersysteme 7*, (1991). pp. 17-22
- [14] Popovic, D., Heine, R., Schnare, T., Wolter, F.: *Suchverfahren zur kollisionsfreien Bahnplanung für Industrieroboter*. *Robotersysteme 8*, (1991). pp. 67-73
- [15] Gorges-Schleuter, M.: *Genetic Algorithms and Population Structures - A Massively Parallel Algorithm*. Doktorarbeit, Universität Dortmund, 1990
- [16] Gorges-Schleuter, M., Sieber, I., Jakob, W.: *Local Interaction Evolution Strategies for Design Optimization*. In: Conf. Proc. CEC 99, IEEE press, Piscataway, N.J., S.2167-2174, 1999
- [17] Blume, C.: *Automatic Generation of Collision Free Moves for the ABB Industrial Robot Control*. In: Proc. of the First International Conference on Knowledge-Based Intelligent Electronic Systems (KES '97), 21-23 May 1997, Adelaide, Australia, pp. 672-683
- [18] Blume, C.: *Programming of Industrial Robots by the GLEAM Evolutionary Algorithm*. In 5th World Multiconference on Systems, Cybernetics and Informatics (SCI 2001) and 7th Internat. Conference on Information Systems Analysis and Synthesis (ISAS 2001), July 22 - 25, 2001, Sheraton World Resort, Orlando, Florida, U.S.A.
- [19] Jakob, W.: *HyGLEAM – An Approach to Generally Applicable Hybridization of Evolutionary Algorithms*. In Merelo, J.J., et al. (eds.): Conf. Proc. PPSN VII, LNCS 2439, Springer, Berlin (2002) 527–536.
- [20] Jakob, W.: *Eine neue Methodik zur Erhöhung der Leistungsfähigkeit Evolutionärer Algorithmen durch die Integration lokaler Suchverfahren*, Dissertation, Universität Karlsruhe, FZKA 6965, März 2004, siehe auch: <http://www.iai.fzk.de/~jakob/HyGLEAM/main-gb.html>
- [21] Jakob, W.: *Towards an Adaptive Multimeme Algorithm for Parameter Optimisation Suiting the Engineers' Needs* T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, pp. 132 – 141, 2006. Springer-Verlag Berlin Heidelberg 2006

Moderne Methoden zur experimentellen Analyse evolutionärer Verfahren

Thomas Bartz-Beielstein und Mike Preuß

Universität Dortmund

Lehrstuhl Algorithm Engineering

Arbeitsgruppe Computational Intelligence

44221 Dortmund

Tel.: (0231) 755-7705

Fax: (0231) 755-7740

E-Mail: {thomas.bartz-beielstein@udo.edu, mike.preuss@cs.uni-dortmund.de}

Zusammenfassung

Die Beschreibung einer Methodik zur Analyse der Industrietauglichkeit evolutionärer Algorithmen steht im Mittelpunkt dieses Beitrags. Dazu werden moderne statistische Verfahren wie “Design and Analysis of Computer Experiments” mit klassischen Verfahren der experimentellen Versuchsplanung (Design of Experiments) kombiniert. Diese Methoden eignen sich sehr gut, um den experimentellen Vergleich verschiedener Algorithmen zu objektivieren und praxisnah durchzuführen. Sie wurden bereits für unterschiedliche Simulations- und Optimierungsszenarien eingesetzt, z.B. für die Optimierung von Transportproblemen in Gebäuden (Fahrstuhlsteuerungsprobleme). Aus diesen Szenarien lassen sich allgemeine und zugleich praxisrelevante Tests definieren, die die Grundlage für den Vergleich von CI-Methoden (Fuzzy-Systeme, neuronale Netze und evolutionäre Algorithmen) mit klassischen Verfahren der Optimierung bilden können.

Des Weiteren stellen wir eine frei verfügbare Sammlung von Programmen zur sequenziellen Analyse von Optimieralgorithmen zur Verfügung (SPOT: Sequential Parameter Optimization Toolbox), die für die oben genannten Zwecke entwickelt wird. Sie beinhaltet klar definierte Schnittstellen, so dass gängige Optimierungstools mit geringem Aufwand eingebunden und analysiert werden können. Die Analysemöglichkeiten gehen über klassische Vergleiche (Bericht von Mittelwerten, Standardabweichungen, etc.) weit hinaus und ermöglichen ein tiefer gehendes Verständnis der Funktionsweise der Algorithmen.

1 Einleitung

Naturalanaloge Optimierverfahren wie genetische Algorithmen (GA), Evolutionsstrategien (ES), Particle Swarm Optimization (PSO) und Differential Evolution (DE) haben in den letzten Jahren an Popularität gewonnen. Der Einfachheit halber fassen wir diese Verfahren als *Heuristiken* zusammen, wohingegen die mathematischen Algorithmen als *exakte Verfahren* referenziert werden. Heuristiken haben die akademischen Spielwiesen verlassen und werden mehr und mehr zur Lösung praktischer Problemstellungen eingesetzt. Es existiert ein ganzer Zoo unterschiedlicher Heuristiken und beinahe täglich werden neue Verfahren veröffentlicht. Jede dieser Heuristiken erfordert die Einstellung so genannter exogener Parameter wie der Populationsgröße oder der Rekombinationswahrscheinlichkeit bei ES/GA, die den Erfolg der Verfahren wesentlich beeinflussen. Zusätzlich stehen diese Verfahren in Konkurrenz zu den klassischen, exakten Verfahren der Optimierung.

Anwender, die auch nur halbwegs objektiv einen Vergleich durchführen möchten, sind durch die Komplexität dieses Unterfangens überfordert. So werden erfolgreiche Anwendungen als Einzelfälle in der Literatur beschrieben, diese sind zumeist stark problemspezifisch und nicht ohne Weiteres auf andere Probleminstanzen oder gar Problemklassen übertragbar.

An dieser Stelle kommt die sequentielle Parameteroptimierung (SPO) zum Einsatz. Mittels statistischer Verfahren werden gute Parametereinstellungen für die Verfahren bestimmt, so dass ein objektiver Vergleich ermöglicht wird. Bartz-Beielstein et al. haben in einer Reihe von Veröffentlichungen den entscheidenden Einfluß exogener Parameter auf die Güte (hier verstanden als Performance) der Heuristiken wie auch der exakten Verfahren demonstriert [1]. In vielen Fällen konnte die Laufzeit der Heuristiken mit SPO um den Faktor 100 verkürzt werden, ohne dass Qualitätseinbußen in Kauf genommen werden mussten.

Wir empfehlen daher eine Anpassung der exogenen Parameter für zeit- und kostenintensive praktische Optimierprobleme. Die hierzu erforderlichen statistischen Grundlagen werden in Abschnitt 2 eingeführt. Das eigentliche Verfahren wird in Abschnitt 3 beschrieben. Der Artikel schließt mit einer Zusammenfassung und einem Ausblick.

2 Klassische und moderne Verfahren

2.1 Klassisches Design of Experiments

Design of Experiments besitzt eine lange Tradition in der Statistik. Grundlegende Konzepte wurden bereits von 1935 Fischer [2] entwickelt. Sie fanden in den darauf folgenden Jahrzehnten Anwendung in der industriellen Optimierung[3] und im Bereich der computergestützten Simulation [4].

Eingabegrößen, die während des Experiments kontrolliert variiert werden, werden als Faktoren oder Designvariablen bezeichnet. Im folgenden bezeichnet x den Vektor dieser Eingabegrößen. Ihre unterschiedlichen Einstellungen werden als Stufen (Level) bezeichnet, ein Sample ist eine spezielle Einstellung. Ein experimentelles Design ist eine Auswahl bestimmter Level für die einzelnen Faktoren, die im Rahmen der Experimente ausgewertet (simuliert) werden. Der erhaltene Ausgabewert, y , wird als Response (abhängige Variable) bezeichnet. Hängt dieser Wert vom Zufall ab, z.B. bei stochastischen Simulationen oder Heuristiken, wird er als Zufallsvariable Y interpretiert. Der (Haupt-) Effekt eines Faktors A kann intuitiv als die durch Variation der Einstellungen von A hervorgerufene Änderung der Response Y verstanden werden, wobei über alle Einstellungen der verbleibenden Faktoren gemittelt wird.

Basierend auf diesen Annahmen lassen sich Varianz- (ANOVA) und Regressionanalysen durchführen, um die Effekte der einzelnen Faktoren zu schätzen. Ein einfaches Regressionsmodell hat in diesem Zusammenhang die Form:

$$y = X\beta + \epsilon, \quad (1)$$

wobei ϵ den Fehler, β die Regressionskoeffizienten und X die Regressionsmatrix darstellen. Zur weiteren Erläuterung verweisen wir auf Standardwerke der experimentellen Versuchsplanung [5]. Anwendungen in der Simulation werden anschaulich in [4, 6] dargestellt, [7] beschreibt beispielhaft die Analyse für evolutionäre Algorithmen.

2.2 Stochastische Prozessmodelle

In den letzten Jahren haben stochastische Prozessmodelle zur Analyse von Computerexperimenten große Popularität gewonnen. Diese auf Sacks et al. [8] beruhenden Verfahren können als eine Erweiterung des klassischen Regressionsmodells (1) interpretiert werden. Sie werden im folgenden als Design and Analysis of Computer Experiments (DACE) Verfahren bezeichnet. Im Gegensatz zu den klassischen Regressionsmodellen gehen sie nicht von einem konstanten Fehler ϵ im gesamten Suchraum aus und benutzen andere, sogenannte raumfüllende (space filling), Designs. Beliebte auf Grund ihrer Einfachheit sind Latin Hypercube Designs (LHD). Sie platzieren Designpunkte unter geringen Einschränkungen zufällig im Inneren des Suchraums, wohingegen klassische Verfahren die Designpunkte an den Rändern (niedrige und hohe Level) wählen. Die Erzeugung eines LHDs wird als Latin Hypercube Sampling (LHS) bezeichnet.

Nach unserer Erfahrung eignen sich stochastische Prozessmodelle besser zur Analyse von Heuristiken, wobei ein großer Vorteil vermutlich durch die Designauswahl gegeben ist. Weiterführende Untersuchungen stehen hierzu allerdings noch aus. Santner et al. [9] haben wichtige Ergebnisse zusammengetragen.

3 Sequentielle Parameteroptimierung

Nachdem im vorigen Abschnitt die grundlegenden Begriffe eingeführt wurden, wird nun gezeigt, wie diese Methoden zur Verbesserung von Heuristiken (Tuning) eingesetzt werden. Die sequentielle Parameteroptimierung (SPO) stellt einen Versuch dar, statistische Verfahren möglichst effizient für das Tuning zu benutzen. SPO wird ausführlich in [1] beschrieben. Es existieren Anwendungen in folgenden Bereichen:

1. Maschinenbau: Temperierbohrungen [10]
2. Luft- und Raumfahrt: Tragflächenoptimierung [11]
3. Simulation und Optimierung: Fahrstuhlsteuerung [12, 13, 14]
4. Verfahrenstechnik: Entwurf von Destillationsanlagen [15]
5. Wirtschaftswissenschaft: Modellierung eines Bodenmarktes [16]
6. Touroptimierungs- und Tourzuordnungsprobleme [17]

Anwendungsfelder im Bereich der Grundlagenforschung sind:

1. Algorithm Engineering: Graph drawing [18]
2. Statistik: Selektionsverfahren optimal computational budget allocation (OCBA) für Partikel-Schwarm-Verfahren [19, 20]
3. Evolutionsstrategien: Threshold Selektion und Schrittweitensteuerung [21]
4. Computational Intelligence: Algorithmische Chemie [15]

5. Partikel-Schwarm-Optimierung: Analyse und Anwendung [22]
6. Numerische Mathematik: Vergleich und Analyse klassischer und moderner Optimierungsalgorithmen [23]

SPO verwendet eine sequentiellen Vorgehensweise bei der Bestimmung geeigneter Designpunkte. Ausgangspunkt ist ein Initialdesign, das zumeist durch LHS generiert wird. Diese Designpunkte bestimmen eindeutig die Parametrisierung der Heuristik. Die Heuristik wird mit diesen Einstellungen ausgeführt, so dass ein Regressionsmodell gebildet werden kann. Dieses Modell wird zur Vorhersage guter Designpunkte herangezogen. Weitere Einstellungen, mit denen die Heuristik ausgeführt wird, sollen einerseits (i) die Güte der Heuristik und (ii) andererseits auch das Modell verbessern. Um diesen Tradeoff zu modellieren, bedient sich SPO des Ansatzes der erwarteten Verbesserung (expected improvement) nach Schonlau [24]. Es werden also sequentiell vielversprechende Designpunkte erzeugt, die das Modell und die Heuristik verbessern.

Da die meisten Heuristiken stochastische Suchverfahren sind, wird durch ein geschicktes Erhöhen der Wiederholungen im Laufe des Tunings die Fehlerwahrscheinlichkeit (Wahrscheinlichkeit, dass ein schlechter Designpunkt nur zufällig ein gutes Ergebnis liefert) verringert.

Für den Anwender verläuft dieser Tuningprozess im Verborgenen, allerdings können die dabei berechneten Statistiken jederzeit abgerufen werden und stehen somit für weiterführende Analysen zur Verfügung. Wir beschreiben nun SPO aus der Sicht des Anwenders, d.h. die Schritte, die ein Anwender für ein Tuning durchführen muss.

Schnittstellen

Die zu SPO gehörige Toolbox SPOT (Sequential Parameter Optimization Toolbox) ist vollständig in MATLAB implementiert und erfordert neben dem MATLAB Grundsystem die Statistik Toolbox. SPOT arbeitet dateibasiert, die Schnittstellen sind einfache Textdateien, die im folgenden beschrieben werden.

SPOT benötigt zwei Dateien:

1. `roi` und
2. `conf` Dateien.

`roi` Dateien spezifizieren den Suchraum (region of interest, ROI) der Einstellungen für die Heuristik. Bild 1 zeigt eine `roi` Datei. Die `conf` Datei beschreibt SPO interne Einstellungen wie z.B. die Wahl des stochastischen Prozessmodells oder die Anzahl initialer Designpunkte für das LHD. Wir verweisen hierzu auf die SPOT Dokumentation [25]. Nachdem diese Dateien erstellt worden sind, kann SPOT aufgerufen werden. SPOT erzeugt ein initiales Design, das in eine sog. `des` (Design) Datei geschrieben wird. Die Heuristik muss in der Lage sein, die Werte der `des` Datei auszulesen und mit den darin spezifizierten Designs zu starten. Für jeden Designpunkt erzielt die Heuristik einen Funktionswert, der zusammen mit diesem Designpunkt zeilenweise in eine `res` (result) Datei geschrieben wird. Die Werte aus der `res` Datei werden von SPOT zur Verbesserung des Regressionsmodells und zur Vorhersage neuer Designpunkte benutzt.


```

name low high isint pretty
NPARENTS 1 10 TRUE 'NPARENTS'
NU 1 5 FALSE 'NU'
TAU1 0.5 3 FALSE 'TAU1'

```

Bild 1: Region of interest (ROI) Datei.

Bild 2 illustriert das Zusammenspiel der einzelnen SPO Komponenten.

Neben der Darstellung des Regressionsmodells und des zugehörigen mittleren erwarteten Fehlers zur Abschätzung der Modellgüte stehen nach Beendigung der SPO eine Vielzahl von Auswertungsmöglichkeiten und Ergebnisdarstellungen (grafisch wie auch textuell) zur Verfügung. Wir empfehlen Methoden der explorativen Datenanalyse (EDA). Momentan existieren bereits mehr als ein Dutzend unterschiedlicher EDA Tools, von denen einige im folgenden dargestellt werden (Bild 3 und 4).

4 Zusammenfassung und Ausblick

Die sequentielle Parameteroptimierung ist ein erster Versuch, ein anwendungsfreundliches Tool zum Tunen zur zur experimentellen Analyse heuristischer Suchverfahren bereitzustellen. Der Anwender muss ein Minimum an Informationen über den zu untersuchenden Algorithmus zusammenstellen und erhält fast automatisch eine verbesserte Einstellung.

Da die theoretische Analyse in der Regel nicht in der Lage ist, sinnvolle Einstellungen für konkrete Optimierprobleme bereitzustellen, schließt SPO diese Lücke durch eine geschickte Kombination moderner statistischer Verfahren. Der Anwender erhält zusätzlich die Möglichkeit, verschiedene Heuristiken (oder auch klassische Optimierverfahren) auf eine möglichst objektive Art und Weise zu vergleichen. Durch die dateibasierten Schnittstellen ist SPO ohne großen Aufwand und unabhängig von Betriebssystemen oder Programmiersprachen einsatzfähig. Interessant ist für uns weiteres Feedback von industriellen Anwendern, um sicherzugehen, dass möglichst viele Anwendungsszenarien durch SPO abgedeckt werden.

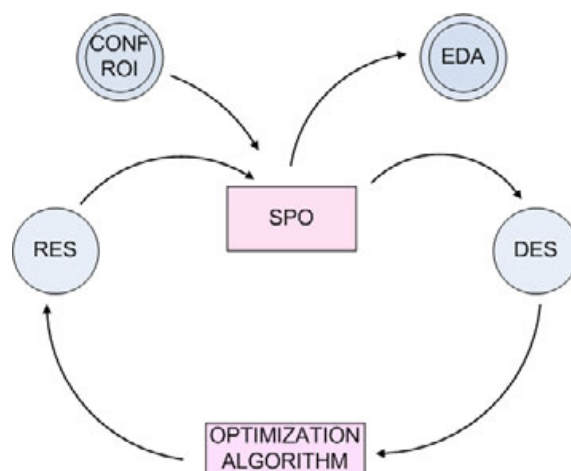


Bild 2: SPOT Datenfluss

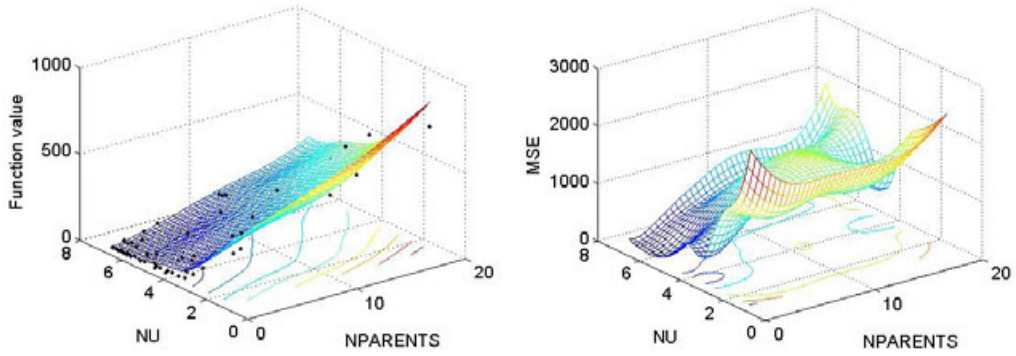


Bild 3: Regressionsmodell (li.) und mittlerer quadratischer Fehler (re.)

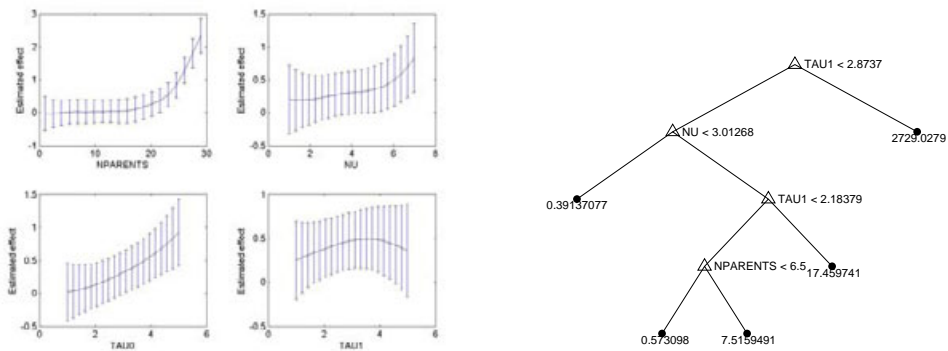


Bild 4: Haupteffekte (li.) und Darstellung der Effekte durch einen Regressionsbaum (re.)

Literatur

- [1] Bartz-Beielstein, T.: *Experimental Research in Evolutionary Computation—The New Experimentalism*. Berlin, Heidelberg, New York: Springer. 2006.
- [2] Fisher, R. A.: *The Design of Experiments*. Edinburgh: Oliver and Boyd. 1935.
- [3] Box, G. E. P.; Hunter, W. G.; Hunter, J. S.: *Statistics for Experimenters*. New York NY: Wiley. 1978.
- [4] Kleijnen, J. P. C.: *Statistical Tools for Simulation Practitioners*. New York NY: Marcel Dekker. 1987.
- [5] Montgomery, D. C.: *Design and Analysis of Experiments*. New York NY: Wiley, 5th Aufl. 2001.
- [6] Kleijnen, J. P. C.; Van Groenendaal, W.: *Simulation—A Statistical Perspective*. Chichester, U.K.: Wiley. 1992.
- [7] Bartz-Beielstein, T.: Experimental Analysis of Evolution Strategies—Overview and Comprehensive Introduction. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–157/03, Universität Dortmund, Germany. 2003.
- [8] Sacks, J.; Welch, W. J.; Mitchell, T. J.; Wynn, H. P.: Design and analysis of computer experiments. *Statistical Science* 4 (1989) 4, S. 409–435.
- [9] Santner, T. J.; Williams, B. J.; Notz, W. I.: *The Design and Analysis of Computer Experiments*. Berlin, Heidelberg, New York: Springer. 2003.
- [10] Mehnen, J.; Michelitsch, T.; Bartz-Beielstein, T.; Henkenjohann, N.: Systematic analyses of multi-objective evolutionary algorithms applied to real-world problems using statistical design of experiments. In: *Proceedings Fourth International Seminar Intelligent Computation in Manufacturing Engineering (CIRP ICME'04)* (Teti, R., Hg.), Bd. 4, S. 171–178. Naples, Italy. ISBN 88-87030-79-0. 2004.
- [11] Bartz-Beielstein, T.; Naujoks, B.: Tuning Multicriteria Evolutionary Algorithms for Airfoil Design Optimization. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–159/04, Universität Dortmund, Germany. 2004.
- [12] Beielstein, T.; Markon, S.: Threshold Selection, Hypothesis Tests, and DOE Methods. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–121/01, Universität Dortmund, Germany. 2001.
- [13] Beielstein, T.; Ewald, C.-P.; Markon, S.: Optimal Elevator Group Control by Evolution Strategies. In: *Proceedings Genetic and Evolutionary Computation Conf. (GECCO 2003), Chicago IL, Part II* (Cantú-Paz, E.; et al., Hg.), Bd. 2724 von *Lecture Notes in Computer Science*, S. 1963–1974. Berlin, Heidelberg, New York: Springer. ISBN 3-540-40603-4. 2003.
- [14] Markon, S.; Kita, H.; Kise, H.; Bartz-Beielstein, T. (Hg.): *Modern Supervisory and Optimal Control with Applications in the Control of Passenger Traffic Systems in Buildings*. Berlin, Heidelberg, New York: Springer. 2006.

- [15] Bartz-Beielstein, T.; Lasarczyk, C.; Preuß, M.: Sequential Parameter Optimization. In: *Proceedings 2005 Congress on Evolutionary Computation (CEC'05)*, Edinburgh, Scotland (McKay, B.; et al., Hg.), Bd. 1, S. 773–780. Piscataway NJ: IEEE Press. ISBN 0-7803-9363-5. 2005.
- [16] de Vegt, M.: Einfluss verschiedener Parametrisierungen auf die Dynamik des Partikel-Schwarm-Verfahrens: Eine empirische Analyse. Interner Bericht der Systems Analysis Research Group SYS-3/05, Universität Dortmund, Fachbereich Informatik, Germany. 2005.
- [17] Bartz-Beielstein, T.; Chmielewski, A.; Janas, M.; Naujoks, B.; Scheffermann, R.: Optimizing Door Assignment in LTL-Terminals by Evolutionary Multiobjective Algorithms. In: *Proceedings 2006 Congress on Evolutionary Computation (CEC'06) Within Fourth IEEE World Congress on Computational Intelligence (WCCI'06)*, Vancouver, Canada (Fogel, D. B.; et al., Hg.). Piscataway NJ: IEEE. 2006.
- [18] Tosic, M.: *Evolutionäre Kreuzungsminimierung*. Diploma thesis, University of Dortmund, Germany. 2006.
- [19] Bartz-Beielstein, T.; Blum, D.; Branke, J.: Particle Swarm Optimization and Sequential Sampling in Noisy Environments. In: *Proceedings 6th Metaheuristics International Conference (MIC2005)* (Hartl, R.; Doerner, K., Hg.), S. 89–94. Vienna, Austria. 2005.
- [20] Bartz-Beielstein, T.; Blum, D.; Branke, J.: Particle Swarm Optimization and Sequential Sampling in Noisy Environments. In: *Metaheuristics—Progress in Complex Systems Optimization* (Hartl, R.; Doerner, K., Hg.), Springer Operations Research/Computer Science Interfaces Book Series. Berlin, Heidelberg, New York: Springer. 2006.
- [21] Bartz-Beielstein, T.: Evolution Strategies and Threshold Selection. In: *Proceedings Second International Workshop Hybrid Metaheuristics (HM'05)* (Blesa Aguilera, M. J.; Blum, C.; Roli, A.; Sampels, M., Hg.), Bd. 3636 von *Lecture Notes in Computer Science*, S. 104–115. Berlin, Heidelberg, New York: Springer. ISBN 3-540-28535-0. 2005.
- [22] Bartz-Beielstein, T.; Parsopoulos, K. E.; Vrahatis, M. N.: Analysis of Particle Swarm Optimization Using Computational Statistics. In: *Proceedings International Conference Numerical Analysis and Applied Mathematics (ICNAAM)* (Simos, T.-E.; Tsitouras, C., Hg.), S. 34–37. Weinheim, Germany: Wiley-VCH. ISBN 3-527-40563-1. 2004.
- [23] Bartz-Beielstein, T.; Parsopoulos, K. E.; Vrahatis, M. N.: Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis & Computational Mathematics (ANACM)* 1 (2004) 2, S. 413–433.
- [24] Schonlau, M.: *Computer Experiments and Global Optimization*. Dissertation, University of Waterloo, Ontario, Canada. 1997.
- [25] Bartz-Beielstein, T.; Lasarczyk, C.; Preuß, M.: Sequential Parameter Optimization Toolbox. <http://ls11-www.cs.uni-dortmund.de/people/tom/ExperimentalResearchPrograms.html>. 2006.

Efficient Product Development of Car Bodies Using Multi-disciplinary Optimization

Prof. Dr. Thomas Bäck

NuTech Solutions GmbH, Dortmund, Thomas.Baeck@nutechsolutions.de

Dr.-Ing. habil. Fabian Duddeck

Queen Mary, London University, f.duddeck@qmul.ac.uk

Formerly with BMW AG, München

Dipl.-Inform. Martin Schütz

NuTech Solutions GmbH, Dortmund, Martin.Schuetz@nutechsolutions.de

Abstract

The usage of multi-disciplinary optimization strategies is rendering the standard product development process (PDP) of car bodies more efficient. This is due to a holistic concept, which takes simultaneously into account all functions required for the design of the structure of a car body. Therefore parts and materials can be used to their full load bearing capacity. In the past, the true industrial applications of the multi-disciplinary optimization (MDO) have been questionable because of the high computational effort and the related long time needed for optimization. Hence in most cases, only methodological studies were realized. In the paper presented here, it is shown that more sophisticated optimization algorithms can accelerate the MDO remarkably. In addition, the automatization of pre- and post-processing plays an important role for the integration of the MDO-method into real life product developments. Completed with a method to control the distribution of the computational loads on the processors, the MDO was applied successfully for the first time at BMW in the real PDP.

1 Required characteristics of optimization for process integration

Automated optimization methods constitute an important part in the development of car body structures. Up to now optimizations have been performed separately for each structural discipline. This implicates the major drawback, that results of each discipline, which has not been considered, have to be additionally validated. As a consequence, a common optimum for all disciplines can never be achieved. Using multi-disciplinary optimization (MDO), all design criteria may be covered and optimized at the same time, such that time consuming coordinations between the respective disciplines may be omitted and a common optimum can be determined.

Until now the disadvantage of a multi-disciplinary optimization is the extreme computational time in case that criteria originating from passive safety are integrated into the optimization process. This extreme effort for multi-disciplinary optimizations may result in turn around times of four weeks or more, a time horizon which is not acceptable for standardized applications in the product development context. An economically justified

use of the MDO approach within this context can only be guaranteed, if the overall time required for the MDO, including pre- and post-processing, remains under two weeks, cf. Lescheticky et al. (2004).

Beside the temporally aspect, an integration of the MDO into the PDP is only practicable when not only optimization experts are able to use a corresponding tool. Based on this fact the necessity may be deduced, that the optimization algorithm must be applicable to all prevalent problems within the development. Additionally, the algorithm should automatically adapt itself to a particular problem. The optimization method, based on the Monte Carlo Analysis, used so far at the car development group at BMW, was not able to fully satisfy all these condition required for an effective integration into the PDP. Hence, in the remaining years, attention was laid on the topic 'Alternative Optimization Algorithms' within the scope of methods development.

Additionally, the usability of the software makes an essential contribution to the successful application of the optimization method within the different projects. Multi-disciplinary optimizations will not be used on a daily basis, but only at particular time slots within the PDP. Consequently, it is necessary to design an optimization software in such a way that it is appealing for the user and invites him for immediate use.

2 Selection of appropriate algorithms for the MDO

2.1 Requirements on the optimization algorithm

Usually, departments of car body development at major car manufacturers define three basic requirements on the algorithms for multi-disciplinary optimization, which will be briefly discussed in the following.

Basically, each employee must be able to use autonomously the optimization software without the help of an optimization expert. Hence, two main criteria for algorithms can be deduced:

- The optimization algorithm should be constructed in such a way that all optimization problems like one- or multi-criteria problems, single- or multi-disciplinary problems, linear or non-linear phenomena, etc. may be handled with the same algorithm.
- Generally, for each problem formulation, optimization algorithms require an individual setting of numerical parameters. The algorithm must be able to adapt these parameters autonomously and optimally to the optimization problem. Consequently, the user may restrict him- or herself to the adjustment of parameters, which are simple and understandable.

Beside the simple handling, the efficiency of the algorithm is a fundamental criterion. Due to the existing requirement of the optimization by non-experts, the algorithm must be able to treat all problems typically found in the context of car body development. However, this means that a potential algorithm can never be as efficient as an algorithm which has been designed especially for a single application. Therefore, the following requirement on the algorithm can be deduced:

- The algorithm must be as general as possible while simultaneously guaranteeing maximal efficiency at given resources.

2.2 Investigated Algorithms

2.2.1 Evolutionary Algorithms

Evolutionary Algorithms (cf. Bäck 1996) belong to a class of optimization algorithms that are inspired by the paradigm of Darwinian Evolution. These methods are characterized by the fact that a number of offspring individuals are produced from one or several parents. Parents of the next generation will be chosen from those off-springs, which possess the highest fitness, i.e. the highest objective function values.

Due to the fact that Evolutionary Algorithms have no special requirements on the problem formulation, e.g. the existence of derivatives, these methods may be used as pure ‘Black-Box’ methods. The only condition, which must be fulfilled for a completely automatic optimization procedure, is the unique evaluation of each solution through one or several objective function values. In contrast to many other methods, Evolutionary Algorithms may compensate the loss of some partial results, which might occur in case of a hard- or software failure, without degrading the complete optimization process. Additionally, these algorithms can easily be parallelized, such that a speed-up of the optimization process may be achieved with minor effort.

Over the last 30 years different evolutionary optimization approaches have been developed nearly independently from each other. In North America, Genetic Algorithms (GA) depict the most widespread and therefore most recognized variant of Evolutionary Algorithms. Beside Evolutionary Programming (EP) as alternative US variant, Evolution Strategies (ES) constitute the mainstream development in Europe. The later strategy turned out to be very successful in case of extremely time consuming real-valued engineering applications, like in the case of the crash simulations considered here, which only allow for a small number of design evaluations.

Beside classical variants of Evolution Strategies alternative optimization algorithms were implemented and/or evaluated in the context of this work. Among them were extensions of Evolution Strategies, Genetic Algorithms, Simulated Annealing and others.

2.2.2 Evolution Strategies

Evolution Strategies are Evolutionary Algorithms, which - based on parent individuals - generate offspring designs by the use of mutation and recombination operators. These operators directly work on the real-valued design variables which are subject to optimization. The most interesting characteristic of Evolution Strategies is the simultaneous optimization of design variables and so-called strategy parameters. The later specify the random distribution with which offspring individuals will be mutated. Since a fundamental paradigm of Darwinian Evolution is the preference of small variations over larger ones, i.e. the mutation process favors gradual changes of individuals, mutation within standard Evolution Strategies usually is carried out with the help of an additive normally distributed random process:

$$x' = x + \sigma \cdot N(0, 1) \quad (1)$$

Hereby, $N(0, 1)$ denotes a normally distributed random variable with expectation 0 and variance 1. σ depicts the strategy parameters of the Evolution Strategy controlling the mutation of the design variables. Consequently, the resulting design variables correspond to normally distributed random variables with expectation x and variance σ^2 . The efficiency of the Evolution Strategy is mainly dominated by an adequate adaptation of the strategy parameters. This adaptation process is autonomously controlled by the strategy and therefore termed self-adaptation. With the help of this mechanism the Evolution Strategy is able to dynamically adapt to the fitness landscape over the course of optimization. Up to now a number of different self-adaptation schemes have been developed which, among others, differ in the number of degrees of freedom of the resulting normal distribution and the goal orientation of the adaptation process.

In the context of this work two self-adaptation mechanisms have been tested, a mutative step-size control and a derandomized variant. The mutative self-adaptation mechanism makes use of one common step size for all design variables which may be increased or decreased by a constant factor in every iteration. This mechanism guarantees that the variance will be increased, decreased or left unmodified each for one third of the offsprings. In contrast, the derandomized variant uses for each design variable one separate variance; it's modification is deduced from successful step-sizes in the past.

Generally, the use of mutation is sufficient for an efficient optimization, such that also strategies with a single parent individual may be successfully utilized. Most notably this is advisable if a small number of objective function evaluations may only be conducted per iteration. Furthermore, the use of a single parent individual leads to a faster convergence in contrast to the use of several parents. The advantage of several individuals lies in an increased stability, i.e. increased global convergence property of the method, reflecting the trade-off between convergence velocity and reliability.

Usually, Evolution Strategies are characterized by the number of parents μ , number of off springs λ , and the maximal number of reproduction cycles κ for each parent. Thus in a (μ, λ, κ) -ES in each generation λ aspirins will be generated from μ parents, whereas the parents may only survive in κ successive generations. For the parameter κ the values 1 and ∞ are very common, whereas a $(\mu, \lambda, 1)$ -ES usually is qualified as (μ, λ) -ES and a (μ, λ, ∞) -ES as $(\mu + \lambda)$ -ES. Over the course of optimization a (μ, λ) -ES temporarily excepts deteriorations of the parents individuals, leading to an increased self-adaptation process of the strategy parameters. In contrast parent individuals will only by substituted in case of a $(\mu + \lambda)$ -ES if better aspiring have been produced. Although this so-called elitist strategy guarantees that in each generation the best known solution will be used as origin, this method may mislead the self-adaptation mechanism.

2.2.3 Alternative Algorithms

Additionally to the Evolution Strategies, alternative optimization algorithms like Genetic Algorithms, Simulated Annealing, and a Monte-Carlo Method were evaluated in the context of this work. Similar to Evolution Strategies, Genetic Algorithms belong to the class of Evolutionary Algorithms. In contrast to ES, Genetic Algorithms regard the recombination of parents as the most essential genetic operator. Furthermore, all genetic operators work on a binary representation of the design variables, thus requiring a mapping in case of real-valued optimization problems. Especially, the emphasis on recombination usually

demands for a comparatively great number of parent and offspring individuals, since otherwise the diversity within the population can not be guaranteed, which is essential for successful optimization. In contrast to Evolutionary Algorithms, Simulated Annealing is not biologically inspired but motivated by physical principles. Mainly, the concept of energy minimization of a physical system in the sequence of a heating and annealing cycles is integrated into an optimization algorithm. Based on a starting point, new solutions are generated with the help of a random process, whereas the range of the random process decreases exponentially over the course of the optimization. Whenever a better solution is found, the actual starting point will be substituted. Over the course of optimization inferior solutions will be excepted as new starting point with an exponentially decreasing probability. In contrast to Evolution Strategies the used distribution of the search process will not be adapted to the local topology of the optimization problem at hand. Instead an external schedule is used, which is independent from the underlying problem.

2.3 Implementation of the requirements for the Algorithms

With Evolution Strategies, Simulated Annealing, and Monte-Carlo methods it is relatively easy to integrate initial solutions into the search process. Since the existence of an initial solution generally may be interpreted in such a way that this solution already lies in a well known, promising subspace, a local search near this solution seems to be most promising. Consequently, in this case, the search distributions have to be initialized appropriately.

Due to the focus on recombination, things are a little more complicated in the case of GAs when using initial solutions. Unfortunately, the cross-over operator produces larger jumps in the search space which, especially in the beginning of the optimization, might lead the algorithm out of the interesting subspace. All algorithms evaluated here, do not rely on the complete evaluation of all newly generated solutions and work even in cases when some evaluation fail. This property is extremely important if the optimization process is based on complex, distributed simulations. Both Evolutionary Algorithms and Monte-Carlo Methods can be easily parallelized since all individuals of a population may be evaluated independently. This property does not hold for Simulated Annealing without modification of the annealing schedule. This is due to the fact that the exponential reduction of the search parameters is designed for sequential evaluations only. An essential property of Evolutionary Algorithms is the possibility to simultaneously provide several solutions at the same time for the continuation of the optimization. Usually, this leads to a stabilization of the optimization process since potential wrong decisions during the selection process do not carry too much weight. This is in contrast to methods which iteratively improve a single solution. Even though the population concept might have a negative effect on the convergence velocity, it constitutes a major advantage concerning the convergence reliability of the strategy.

Under all algorithms considered in this evaluation, Evolution Strategies are simplest to parameterize. Only a careful initialization of the step sizes has to be considered. While an initial step size, which is too large, may evoke an erratic behavior of the strategy within the small number of available iterations, the algorithm will perform poorly if the initial step size is too small. Fortunately, the already necessary problem formulations allow for the automatic deduction of a robust initialization with the help of simple heuristics. Since the ES adapts the critical step sizes over the course of the optimization, a heuristic initial adjustment is absolutely sufficient here. In the case of a GA the coding of the design

variables has to be additionally specified, i.e. the number of bits representing a design variable. Thereby it has to be considered that a small number of bits allow for a faster convergence, but with less precise results. In contrast, a higher number of bits leads to a lower convergence velocity with a higher precision. For Monte-Carlo methods the distributions of the stochastic search have to be fixed a-priori. Since these distributions will not be automatically adapted during the optimization, they must be appropriate for both phases of the optimization, the beginning and the end. Unfortunately, it is not always easy to find this compromise. Usually, the discovery of such a compromise requires a lot of knowledge about the problem at hand, something which normally is not available. In the case of Simulated Annealing, beside the initial search distribution, the number of iterations of the optimization must be known in advance, since otherwise the annealing schedule can not be calculated. This is unpractical in many cases.

3 Selection of the algorithm via benchmark problems

3.1 Strategy

For a permanent use of new processes and procedures the general applicability of the approach is important. At first, a study was accomplished to identify possible areas of application of the optimization method. It turned out that both, the method and the process in which the method is integrated, must solve mono- and multi-criteria (cf. Deb 2001) as well as mono- and multi-disciplinary optimization tasks. Since typical questions arising in the car body development are not only restricted to a few parameters, algorithms of interest should be applicable to any number of optimization parameters. For all investigations, several tests problems relevant for the praxis were selected.

Over all 20 different optimization algorithms, based on Monte-Carlo simulation, evolution, genetic and other approaches were compared. All these methods are characterized by a probabilistic nature, i.e. they provide different optimizations under identical conditions. Hence, all test problems were optimized several times for each algorithm to assure the significance of the comparisons. Three different examples will be presented in the following sections.

3.2 Benchmarks

3.2.1 Mono-criteria MDO Crash / Stiffness / Dynamic

First of all, a simplified model for concept studies was defined. The NVH (noise, vibration, and harshness) model consisted of approximately 90,000 elements, the crash model of approximately 130,000 elements. All component thicknesses were used as optimization parameters with a simultaneous retention of the symmetry. The optimization target was the minimization of the body mass while respecting 18 constraints. Since at the beginning of the optimization some of the constraints were violated, each optimization algorithm had not only to reduce mass but at the same time to eliminate constraint violations.

Independent of the used algorithm, all optimizations were conducted with 10 iterations at 28 simulations each. In order to be able to provide simple test statistic the optimizations

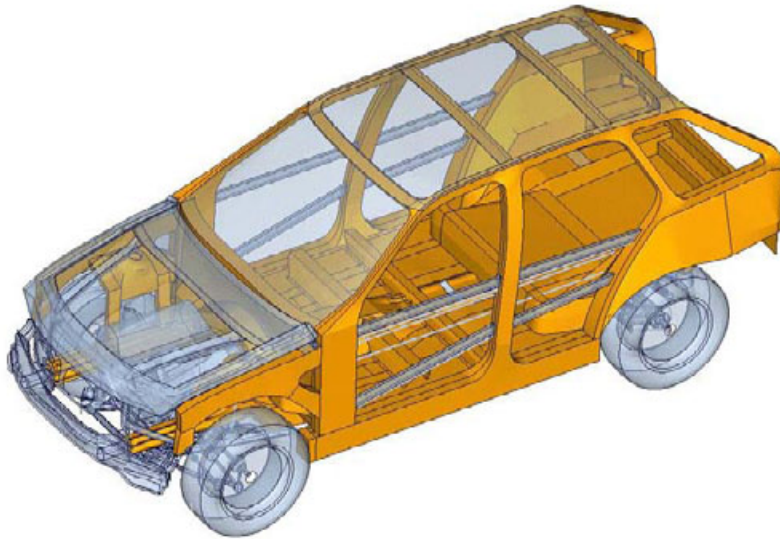


Figure 1: Mono-criteria MDO Crash / Stiffness / Dynamic example problem.

were repeated at least 9 times. Table 3.2.1 shows the results of the Monte-Carlo method and the finally selected Evolution Strategy. It clearly turned out that the ES performed significantly better than the so far used Monte-Carlo approach. Even the smallest mass reduction produced by the ES lies at a comparable level with the mean mass reduction found by the Monte-Carlo method.

Algorithm	Mean mass reduction	highest mass reduction	smallest mass reduction
Monte-Carlo	-6.6 kg	-8,3 kg	-3.3 kg
Evolution Strategy	-9.0 kg	-13,4 kg	-6.3 kg

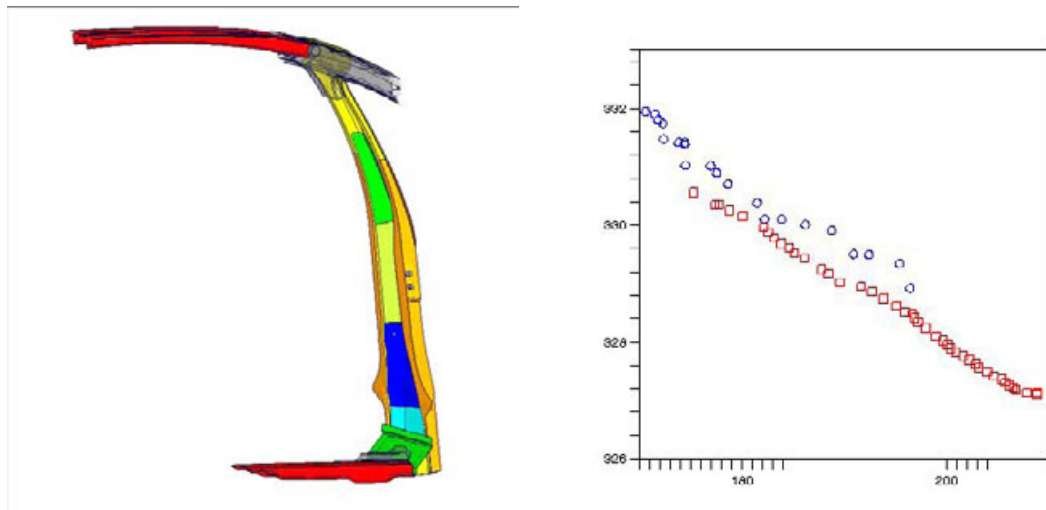
Table 1: Results of the mono-criteria MDO Crash / Stiffness / Dynamic example.

3.2.2 Multi-criteria optimization based on a side crash model

Until now, a true multi-criteria optimization could not be performed with the actual Monte-Carlo approach. In order to determine correlations between the two target objective functions, approaches using regression models were applied. These models have the disadvantage that they only work for a relatively small number of parameters. With an increasing number of parameters they tend to fail very easily.

Here a sub-structure model for lateral impact was chosen. The aim was the identification of the correlation between the mass of the B-pillar and the intrusion. Over all 10 components were subject to variation, where for each of them the corresponding gauge thickness could be modified.

Figure 2 (b) shows a comparison of two pareto fronts, the front evolved by the ES (circles) and the corresponding front generated via regression. Both fronts approximately possess the same characteristics. In contrast to the regression based pareto front, the front detected



(a) The side crash model.

(b) Results of the Multi-criteria optimization based on the side crash model - Evolution vs. Regression.

Figure 2: Multi-criteria optimization based on a side crash example.

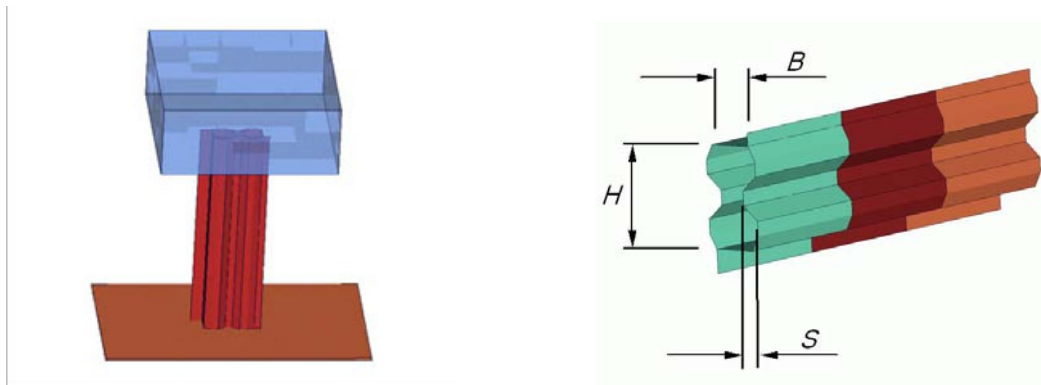


Figure 3: Benchmark example: Fall tower experiment for engine mount.

by the ES tends to have a slightly higher weight level. Concerning computing costs for this example, the ES requires a higher effort in contrast to the regression model. This relation reverses with an increasing number of design parameters.

This result clearly shows that a pareto front analysis can reasonably be carried out with the help of Evolutionary Algorithms, thereby obtaining a similar quality as with an usually used regression approach. The main advantage of the new method in contrast to conventional approaches lies in the scalability of the number of parameters.

Additionally, regression models are usually more exact for the lateral crash than for other crash load cases. This is due to the fact, that the underlying physical effects are more linear and very rarely produce bifurcations. This implies that Evolutionary Algorithms tend to have a better performance in case of the other load cases.

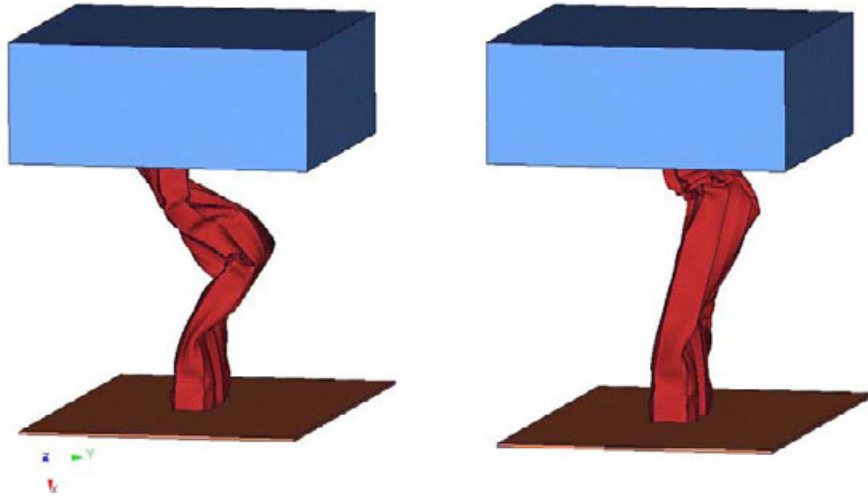


Figure 4: Final deformation state of the engine mount before (left) and after (right) optimization.

3.2.3 Multi-criteria shape optimization

Besides the question for variations in wall thicknesses and materials, shaping is a very essential topic in early phases of vehicle projects. Consequently, interesting optimization algorithms must be able to find a solution, even if the geometry changes, which probably may result in bifurcations of the search space. Large geometry changes require re-meshing of the finite element model, which renders the objectives more noisy.

As additional algorithm benchmark a fall tower experiment for an engine mount was used. In this example a falling mass hits onto the engine mount. A mass optimal form of the engine mount was searched for, whereas the axial load level had to be greater than 90 kN. Additionally, the folding of the mount should take place from front to the end. Overall 9 free geometry variables, 7 dependent variables, and 3 constraints were defined.

Figure 4 shows two different deformation patterns of the engine mount. The deformation of the geometrically optimized mount showed a much better and stable folding than the initial model.

Figure 5 represents the improvement of the optimization with respect to mass. While the Monte-Carlo optimization is not able to find a significant mass reduction within a limited time, the Evolution Strategy shows a continuous and significant reduction in mass. The figures clearly depict the difference originating from the adaptation of the Evolution Strategy.

With this example the proof was given that the chosen Evolution Strategy is able to manage non-linearities and bifurcations more efficiently than the Monte-Carlo method even in cases of shape optimizations.

4 Requirements on the Software

The success of a multi-disciplinary and/or multi-criteria optimization not only depends on the used algorithms. Likewise the time-frame necessary for the execution of the complete

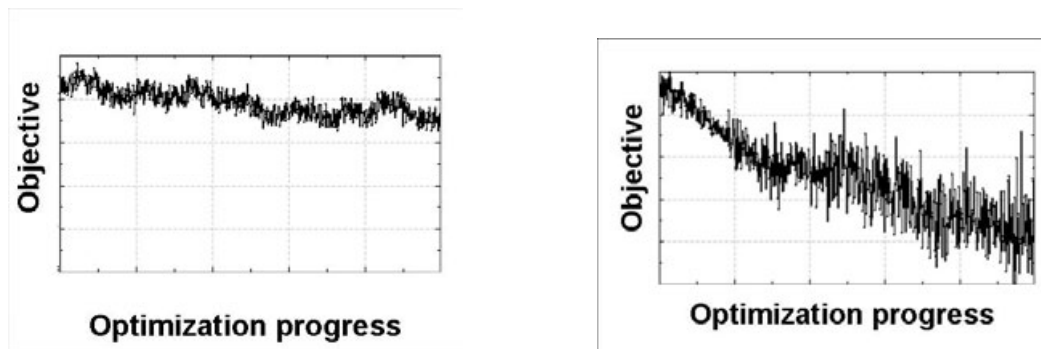


Figure 5: Comparison of the optimization improvement: Monte Carlo (left) vs. Evolution (right). Both plots show the mass reduction over the number of generations (abscissa). In both cases the same scaling of the axis is used.

optimization project is important in the context of vehicle development. A development project will not stop working for 4-5 weeks in order to wait for the results of a MDO and then initiate adequate actions. The results obtained after the necessary 4-5 weeks refer to a construction stage which is already obsolete after the end of the optimization. According to this the results are no longer relevant for the development project. A MDO may only be reasonably integrated into the development process if the time-frame of a complete analysis, including pre- and post-processing, takes no longer than 2 weeks. This necessary time reduction of 2-3 weeks can not be achieved solely through an efficient algorithm. The complete software environment around the optimization algorithm additionally plays an important role. Especially, a failsafe distribution of all computing jobs, including pre- and post-processing is important.

Within the car body development of BMW a MDO usually requires approx. 4 days of pre-processing time, approx. 20 days for the analysis and additionally approx. 4 days for the complete post-processing. From this the following starting points for the reduction of the necessary time-frame may be deduced:

- In the context of pre- and post-processing many workflows are standardized and may be automated with the help of computer scripts.
- Besides an efficient optimization algorithm the duration of the complete MDO may be drastically reduced through an intelligent use of CPU resources. In this context the varying runtimes of the different crash types build the main starting-point. While the runtime of the front crash usually lies between 1 and 2 days other load cases take 1/2 to 1 day. With the help of a clever parallelization of the different load cases the runtime of the pure MDO can be significantly reduced. Since queueing systems do not have any information about the different load cases and their necessary runtimes, there must exist a possibility for user intervention which allows for the specification of the parallelization of the different load cases.
- Besides the timely partitioning of the load cases CPU time may additionally be saved if several crash or static analyses are cancelled as soon as a restriction is violated.

These requirements call for a powerful GUI for a distributed optimization software. With the help of this GUI the user must be able to easily setup his workflow and integrate

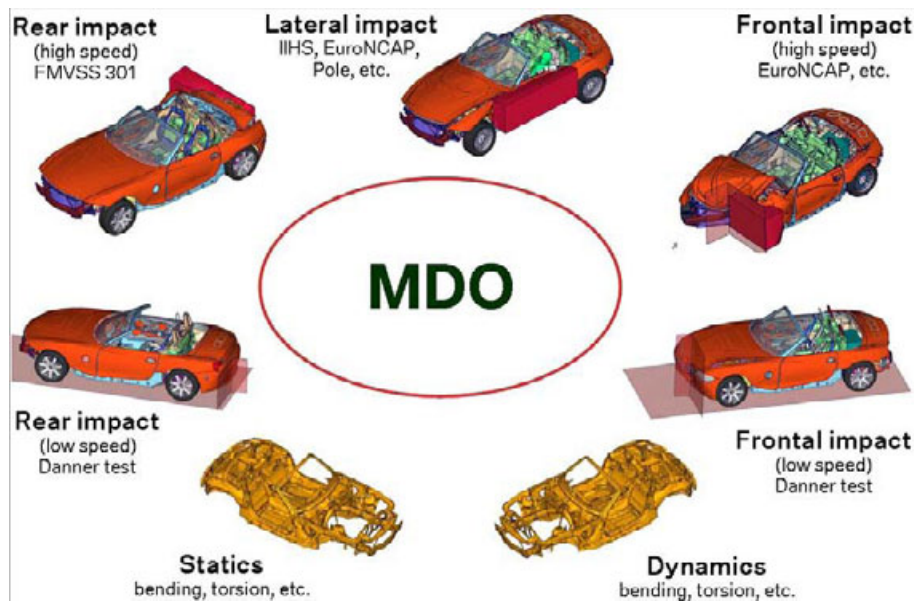


Figure 6: Illustration of the different investigated load cases for the MDO mass reduction.

existing pre- and post-processing scripts. Through the flexibility of the software and the use of an Evolutionary Algorithm as optimizer, the time-frame for a complete MDO was reduced from 4-5 weeks to 2 weeks.

5 Examples of use

After the selection of the best suited algorithm (cf. Wörmann et. al. 2006) two typical examples of use from the car body development were chosen in order to prove the applicability and efficiency of this algorithm under conditions relevant for the praxis.

Since for this example models with approx. 1,000,000 elements were used, the optimization could not be repeated several times. Consequently, the scatter band of the solutions remains unknown and the comparison with the actually used optimization method could not statistically be backed up.

For the example presented here, the high speed load cases front, side, and rear crash were coupled with the assurance load cases (Danner tests) AZT front and rear as well as linear analyses for static stiffness and dynamic frequencies.

While the NVH model uses approx. 300,000 elements, the models for the crash and assurance load cases possess approx. 1,100,000 elements. Most of the gauge thicknesses of the car body were used as free parameters of the optimization, resulting in 136 parameters.

Each discipline and each load case possess their own constraints. Over all 47 constraints had to be respected resulting from the different load cases.

5.1 Problem 1: Multi-disciplinary weight optimization

The first example relevant for the development deals with the question about a possible weight reduction for a given vehicle concept. Although the car body can not only be

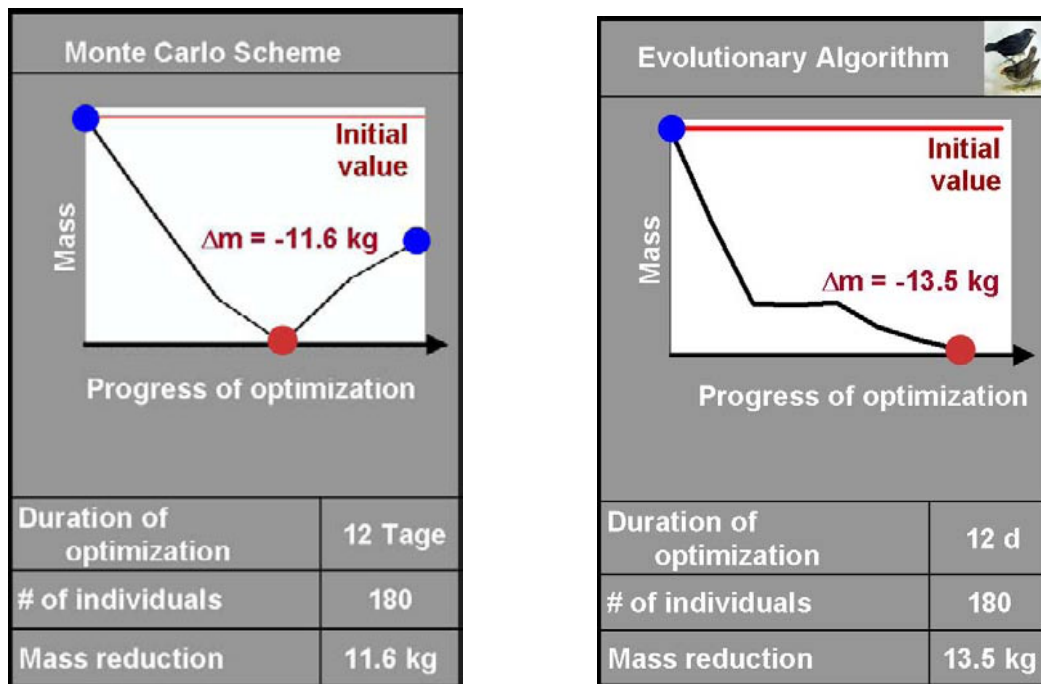


Figure 7: Comparison of the MDO results concerning weight reduction: Monte Carlo vs. Evolution.

constructed for crash or for stiffness or for dynamic, until now a separate optimization for crash and NVH is the standard procedure. Unfortunately, this separate procedure often requires a plausibility check in order not to mislead the respective optimization of crash and NVH respectively. Thus, the complete optimization was carried out in an iterative way, which is work and CPU time intensive.

In order to clearly demonstrate the process advantage of the Evolution Strategy the weight optimization of a car body (cf. Figure 6), including deformation, stiffness and frequency constraints, was used as a first example.

Figure 7 shows the optimization progress for both algorithms, the Monte-Carlo Method and the Evolution Strategy. In order to be able to compare the algorithms, both optimizations were performed with 180 design evaluations.

The direct comparison of both runs clearly indicates that in contrast to the Monte-Carlo method the ES was able to extract approx. two additional kilogram from the body. Additionally, the progression of the fitness curve shows a much faster convergence and a higher mass reduction than the actual standard. As a consequence, the 12 days computing time could be reduced by approx. 50% if a lower mass reduction, comparable to that of the Monte-Carlo approach is acceptable.

5.2 Problem 2: Investigation of the correlation between the body weight and the front wall intrusion in case of the frontal crash

In early phases of a vehicle project the fixation of target values is the main aim. In order to do so, different requirements like functional, operational, weight related, those related to product engineering and others will be posed. Usually, in case of the first target design,

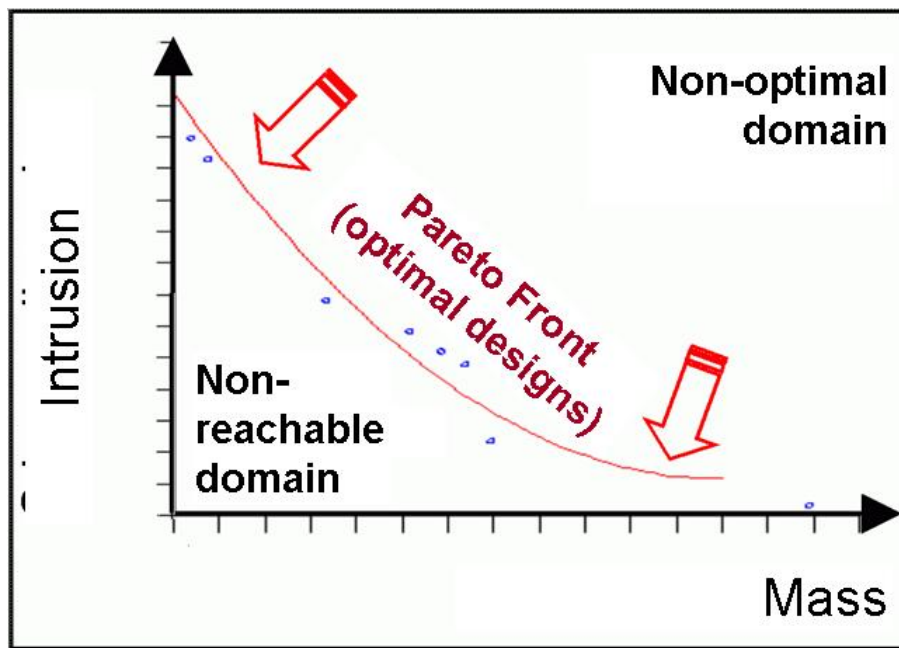


Figure 8: Correlation between the front wall intrusion and the weight of the bodyshell.

all these requirements do not fit together and much time is spent to work out a best compromise. In this working phase support from virtual design is extremely important, since virtual design is able to highlight functional relations of a new vehicle concept. Until now the identification of functional relations was restricted to singular investigation of a single load case, e.g., the correlation between the front wall intrusion and the body weight in case of a frontal crash. Thereby it was not possible to check whether restrictions from other disciplines were violated.

Through the combination of multi-criteria and multi-disciplinary optimization it is now possible to identify functional relations between different guidelines. At the same time restrictions of other disciplines can be met too.

By the means of the example mentioned before, the relation between the intrusion into the footwell and the body mass was determined for an Euro-NCAP frontal impact.

Over all approx. 180 designs were evaluated. With this investigation it was shown that an accentuation of the front wall intrusion by 10 mm implicates a weight increase of approx. 5 kg. Hereby, all other aims for stiffness, dynamic, etc. were met. Within the target identification process, these technically sound-standing statements about the relation between two quantities mostly build the only reliable argument for the generation of the best possible target system of a vehicle.

6 Summary

Within a perennial development period a methodology and procedure for the multi-criteria and/or multi-disciplinary optimization was developed at BMW. Beside usability and flexibility of a software, the optimization algorithm itself is an essential building block. Firstly, by means of benchmark examples 20 different algorithms have been tested with respect to their universal suitability in the context of car body development (passive safety, static

stiffness and dynamic frequencies). Within these tests the Evolution Strategy turned out to be the most appropriate approach which could be successfully applied to all examples. In any case the ES delivered similar or superior results compared to those produced by the standard Monte-Carlo approach.

After these tests, the selected Evolution Strategy was used for the solution of two typical daily problems from the current car body development. In connection with a semi-automatic pre-and post-processing the new method allowed for a reduction of the usually necessary 5 week development time to only 2 weeks. With the help of this new time-frame of just two weeks for a multi-criteria / multi-disciplinary optimization with more than 100 design variables, a successful integration of the method into the development process of BMW was possible.

The focus of future work will lie on the further shortening of the necessary time-frame. Intelligent resource allocation and a fine tuning of the optimization algorithm will build the main potential for achieving this goal.

7 Bibliography

- Bäck T. (1996): *Evolutionary Algorithms in Theory and Practice*. New York: Oxford University Press.
- Blumhardt, R. (2002): *Numerische Optimierung des Crashverhaltens von Fahrzeugstrukturen und -komponenten*. Herzogenrath: Shaker-Verlag.
- Deb K. (2001): *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester: John Wiley & Sons.
- Duddeck, F.; Heiserer, D.; and Lescheticky, J. (2003): *Stochastic methods for optimization of crash and NVH problems*. In: Proc. of the Second M.I.T Conf. on Comput. Fluid and Solid Mech., Cambridge, Mass.; Amsterdam: Elsevier, pp. 2265-2269.
- Lescheticky, J.; Duddeck, F.; Girona, S.; and Willmes, L. (2004): *Effizientere Produktentwicklung durch den Einsatz multidisziplinärer Optimierungsverfahren im Entwurfsprozess von Karosserien*. In: Numerical Analysis and Simulation, VDI-Berichte, Würzburg, No. 1846, pp 583-601
- Wörmann et. al. (2006): *ClearVu Global Optimizer for Optimus Version 1.3.1*. Dortmund: NuTech Solutions GmbH

Modellierungsmethodik für Mixed-Signal Automotive Schaltkreise mittels Data-Mining Methoden und Fuzzy-Systemen

Holger Mielenz¹, Rolando Dölling¹ und Wolfgang Rosenstiel²

¹ Robert Bosch GmbH, EIM3, Abteilung für Methoden, Tools und Technologie im IC-Entwurf, Reutlingen
{holger.mielenz, rolando.doelling}@de.bosch.com

² Universität Tübingen, Lehrstuhl für Technische Informatik, Tübingen
rosenstiel@informatik.uni-tuebingen.de

Hintergrund. In der Automobilelektronik stehen die Entwickler zunehmend vor der Aufgabe, Lösungen für Probleme steigender Komplexität in kürzeren Entwicklungs- und Produktionszyklen zu finden, um auf dem Weltmarkt wettbewerbsfähig zu bleiben. Diese Tendenz manifestiert sich auch in den Anforderungen an anwendungsspezifische, integrierte Schaltkreise (ASICs), deren Realisierung verstärkt als *one-chip* Implementierung der einzelnen Komponenten (*System-on-Chip*, SoC) erfolgt. Für die funktionale Verifikation der entworfenen Schaltungen wird meist eine ausgiebige Überprüfung per Simulation angestrebt, was aufgrund der zunehmenden Komplexität der Schaltkreise zu schwer handhabbaren Randbedingungen und langen Simulationszeiten führt.

Problemstellung. Vor diesem Hintergrund ist es ersichtlich, dass neue Konzepte für die Simulation von ASICs benötigt werden, um den gestiegenen Anforderungen gerecht zu werden. Daher werden unterschiedliche Ansätze für die Modellierung von ASIC-Komponenten untersucht, wobei der Fokus auf eine Automatisierung der Modellgenerierung und eine Verkürzung der Simulationszeit gelegt wird. Insbesondere Analog-Digital Mixed-Signal (A/MS) Schaltkreise stellen die Entwickler vor grosse Herausforderungen, da ereignis-diskrete und zeit-kontinuierliche Signalformen integriert betrachtet werden müssen. Die vorliegende Arbeit diskutiert einen datenbasierten Ansatz, der für derartige Schaltungen eine weitestgehend automatisierte Generierung vereinfachter mathematischer Beschreibungen der Schaltungsfunktionalität ermöglicht, womit ein Gewinn an Simulationszeit erzielt werden kann.

Methodik. Die verwendete Methodik stützt sich auf grundlegende Konzepte der Fuzzy-Theorie zur Modellierung nichtlinearer dynamischer Systeme, die um Funktionalitäten zur Handhabung diskreter und kontinuierlicher Signalverläufe erweitert werden. Um die Eignung der einzelnen Ansätze zu untersuchen, werden insbesondere Mandami-, Takagi-Sugeno- bzw. Takagi-Sugeno-Kang-Systeme auf ihre Leistungsfähigkeit bezüglich der geschilderten Problemstellung analysiert. Da die notwendige Modellgenauigkeit dieser Fuzzy-Systeme meist nur über umfangreiche Regelbasen zu erhalten ist, die deutlich zu einer Verlängerung der Simulationszeiten beitragen, wird ein Entscheidungsbaum basierter Ansatz vorgestellt, der eine partielle Modellierung einzelner Zustände des Gesamtsystems ermöglicht und zu einem Gewinn an Simulationszeit führt.

Ergebnisse und Schlussfolgerung. Um die Eigenschaften der geschilderten Methodik zu illustrieren, wird die Modellierung und Simulation einer *Charge-Pump* Schaltung vorgestellt, die als grundlegende A/MS-Schaltungsstruktur in modernen Anwendungen des Automobil- und Telekommunikationsbereichs weit verbreitet ist. Insbesondere wird der Einfluss der Tiefe des Entscheidungsbaumes auf die Modellgenauigkeit und den Gewinn an Simulationszeit diskutiert, indem dessen Nutzen für den A/MS-Schaltungsentwurf aufgezeigt wird.

The Effect of Kernelization in Relational Fuzzy Clustering

Thomas A. Runkler
Siemens AG Corporate Technology
Information and Communications, CT IC 4
81730 München
Tel. 089/636-45372
E-Mail Thomas.Runkler@siemens.com

Abstract

This paper deals with finding fuzzy clusters in relational (as opposed to object) data that quantify relations between objects, for example similarities between languages, or the degrees of people liking each other. A popular method for finding fuzzy clusters in relational data is non-Euclidean relational fuzzy c -means (NERFCM).

Due to the progress in support vector machines, *kernelization* has gained a lot of attention in the scientific community. The idea is to map the data to a very high-dimensional space in order to obtain more convenient data distributions. In classifier design, for example, the goal is to obtain classes that can be (approximately) separated linearly in the high-dimensional space. This transformation does not have to be performed explicitly. Using the so-called kernel trick this transformation is implicitly done by replacing dot products by kernel functions.

Recently, the kernel trick has been successfully applied to many clustering models, including NERFCM. In this paper we analyze the reasons why kernelized relational fuzzy clustering algorithms often produce better results than their unkernelized variants. In this analysis we consider a simple artificial three point data set. The experiments with KNERFCM show that kernelization leads to a limited visibility of the data and thus implicitly ignores outliers. These effects provide a good explanation of the improvements achieved by kernelization.

1 Introduction

Clustering partitions a set of objects into subsets (clusters). Fuzzy clustering produces fuzzy subsets that allow each object to be partially contained in several clusters. Often the objects are quantified by numerical feature vectors, so that we associate the feature vectors with the objects themselves. Hence, clustering of feature vectors is often denoted as *object data clustering*. A popular model for fuzzy object data clustering is *fuzzy c -means (FCM)* [1]. In many applications, however, it is hard or even impossible to specify features that reasonably describe the objects under examination. In these cases it is often more convenient to quantify the objects by pairwise *relations*. For example, if the objects are languages, it might be convenient to represent the objects by a matrix of similarities between each pair of languages, or if the objects are people, it might be convenient to represent the objects by a matrix of degrees of sympathy for each other. The relations (similarities, sympathies, etc.) may be subjectively and manually specified, or they may

be computed from some features of the objects. In this paper we do not consider the generation of the relation matrices but assume that they are simply given. Relational data can in principle be clustered by objective data clustering algorithms, for example using the fuzzy c -means model [9]; in this paper, however, we will focus on fuzzy clustering algorithms that are actually tailored for relational data and are therefore denoted *relational fuzzy clustering* [14]. Many heuristic algorithms were proposed for clustering, such as self-organizing maps [10], alternating cluster estimation [16], relational alternating cluster estimation [15, 17], or swarm intelligence methods [21]. The disadvantages of these heuristic algorithms is that they produce clustering results, but nothing can be said about the quality of the results or about convergence properties. In contrast to these heuristic methods many clustering algorithms use objective functions that are optimized, so the clustering results are always (at least local) optima of these objective functions [1]. So, this paper deals with *relational fuzzy clustering using objective functions* that can be formulated as the following problem: Given a relational data set $R \subset \mathbb{R}^{n \times n}$, an objective function $J : (U, R) \rightarrow \mathbb{R}$, and a number of clusters $c \in \{2, \dots, n - 1\}$, find a partition matrix $U \in M_{\text{fcn}}$, where

$$M_{\text{fcn}} = \left\{ U \in [0, 1]^{c \times n} \mid \sum_{i=1}^c u_{ik} = 1, k = 1, \dots, n, \sum_{k=1}^n u_{ik} > 0, i = 1, \dots, c \right\}, \quad (1)$$

so that the objective function J is minimized. Fuzzy c -means can be extended to relational data by reformulation [6], yielding the *relational fuzzy c -means (RFCM)* model [2]. RFCM works well only for relational data that represent Euclidean distances between pairs of feature vectors. In general, this is not the case, so RFCM often produces non-fuzzy results $U \notin M_{\text{fcn}}$. In particular, some memberships might even be negative or larger than one. This can be avoided by the so-called β -spread transform, which yields *non-Euclidean relational fuzzy c -means (NERFCM)* [5]. NERFCM assumes that the clusters are approximately hyper-spherical and well separated. Again, this is not the case in many data sets. To avoid this, *kernelized* clustering has become popular recently. Kernelization uses a nonlinear transformation of the data that, according to Mercer's Theorem [11, 18], does not have to be explicitly computed. This so-called *kernel trick* has been successfully applied in *support vector machines (SVM)* [12, 20] to produce classifiers for classes with highly nonlinear class boundaries. In the same way, the kernel trick has been applied to clustering, in particular to fuzzy c -means [23, 24, 22] and to non-Euclidean relational fuzzy c -means (NERFCM) [8], yielding *kernelized non-Euclidean relational fuzzy c -means (NERFCM)*.

The effect of kernelization is often described in terms of nonlinear mappings and kernel functions. In this paper, however, we investigate the *practical* effects of kernelization in relational clustering. We show *why* kernelization often produces superior results, and derive a guideline to determine suitable kernel parameters.

This paper is structured as follows: Section 2 briefly reviews the fuzzy c -means model (FCM). Section 3 describes the *relational* fuzzy c -means model (RFCM). Section 4 presents the *non-Euclidean* relational fuzzy c -means model (NERFCM). Section 5 introduces the *kernelized* non-Euclidean relational fuzzy c -means model (kNERFCM). And Section 6 presents our experiments with kernelization and discusses their results.

2 Fuzzy c–Means (FCM)

The *fuzzy c–means model* (FCM) [1] is defined as the following problem: Given an object (feature) data set $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$, a number of clusters $c \in \{2, \dots, n-1\}$, a fuzziness index $m > 1$, and a norm $|\cdot|$ on \mathbb{R}^p , find a set of prototypes (cluster centers) $V = \{v_1, \dots, v_c\} \subset \mathbb{R}^p$ and a partition matrix $U \in M_{\text{fcn}}$ (1) by minimizing the objective function

$$J_{\text{FCM}}(U, V; X) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m |x_k - v_i|^2. \quad (2)$$

Optimization of the FCM clustering model can be done by random initialization of $V \subset \mathbb{R}^p$ and then alternating optimization through the necessary conditions for extrema of $J_{\text{FCM}}(U, V; X)$.

$$u_{ik} = 1 \left/ \sum_{j=1}^c \left(\frac{|x_k - v_i|}{|x_k - v_j|} \right)^{\frac{2}{m-1}} \right., \quad (3)$$

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m}, \quad (4)$$

$i = 1, \dots, c, k = 1, \dots, n$. This algorithm terminates after a fixed number of $t \in \mathbb{N}$ steps, or when a termination condition on V holds. In a similar way, initialization and termination can be done on U .

3 Relational Fuzzy c–Means (RFCM)

As mentioned earlier, fuzzy c–means (FCM) is actually designed for object (feature) data, but can also directly be applied to relational data [9]. In this paper, however, we focus on relational fuzzy clustering models that are *explicitly* designed for relational data $R \subset \mathbb{R}^{n \times n}$. One approach to modify FCM to be applicable to relational data is to avoid the explicit computation of the cluster prototypes V . This is done by inserting the equation to compute the cluster prototypes V (4) into the FCM objective function (2). This *reformulation* [6] yields the *relational fuzzy c–means* (RFCM) [2] model with the objective function

$$J_{\text{RFCM}}(U; R) = \sum_{i=1}^c \frac{\sum_{j=1}^n \sum_{k=1}^n u_{ij}^m u_{ik}^m r_{jk}^2}{\sum_{j=1}^n u_{ij}^m}. \quad (5)$$

Optimization of J_{RFCM} can be done by randomly initializing and then iteratively updating U using the necessary conditions for extrema of RFCM:

$$u_{ik} = 1 \left/ \sum_{j=1}^n \frac{\sum_{s=1}^n \frac{u_{is}^m r_{sk}}{\sum_{r=1}^n u_{ir}^m} - \sum_{s=1}^n \sum_{t=1}^n \frac{u_{is}^m u_{it}^m r_{st}}{2 \left(\sum_{r=1}^n u_{ir}^m \right)^2}}{\sum_{s=1}^n \frac{u_{js}^m r_{sk}}{\sum_{r=1}^n u_{jr}^m} - \sum_{s=1}^n \sum_{t=1}^n \frac{u_{js}^m u_{jt}^m r_{st}}{2 \left(\sum_{r=1}^n u_{jr}^m \right)^2}} \right., \quad (6)$$

$i = 1, \dots, c, k = 1, \dots, n$. Again, this algorithm terminates after a fixed number of $t \in \mathbb{N}$ steps or when a termination criterion on U holds.

Relational duals for other object clustering models can be found in [7]. A relational dual of the Gustafson–Kessel clustering model [4], an extension of the FCM model with local covariance matrices, can be found in [13].

4 Non–Euclidean Relational Fuzzy c–Means (NERFCM)

Consider a given object (feature) data set X and its equivalent relational data set $R(X)$ that is computed using the Euclidean norm $\|\cdot\|$, and assume that the Euclidean norm $\|\cdot\|$ is also used as the norm $|\cdot|$ in the FCM model. Both models $J_{\text{FCM}}(U, V; X)$ and $J_{\text{RFCM}}(U; R(X))$ will then produce the same results. In the general case, however, R might be computed from a data set X using a non–Euclidean norm, or it might even be obtained without any underlying object data set, for example by manual rating. In these cases, RFCM does not match any FCM instance any more. Moreover, the RFCM model might even yield partition matrices $U \notin M_{fc}$, in particular we can even have some $u_{ik} < 0$ or $u_{ik} > 1$. This problem can be fixed by transforming the so–called non–Euclidean distance matrix D into a Euclidean distance matrix D_β by applying a so–called β –spread transform [5]

$$D_\beta = D + \beta \cdot B, \quad (7)$$

with a suitable $\beta \in \mathbb{R}^+$, where $B \in [0, 1]^{n \times n}$ is the off–diagonal matrix with $b_{ij} = 1$ for all $i, j = 1, \dots, n, i \neq j$, and $b_{ii} = 0$ for all $i = 1, \dots, n$. In the *non–Euclidean relational fuzzy c–means* (NERFCM) algorithm [5] the value of β is successively increased, i.e. higher values of β are added to the off–diagonal elements of R , until the Euclidean case is achieved and $U \in M_{fc}$ is asserted.

5 Kernelized Non–Euclidean Relational Fuzzy c–Means (kNERFCM)

The goal of kernelization is to transform a data set $X = \{x_1, \dots, x_n\} \in \mathbb{R}^p$ to a higher dimensional data set $Y = \{y_1, \dots, y_n\} \in \mathbb{R}^q, q > p$, so that the data structure in Y is in some sense simpler than in X . For example, in *support vector machines* (SVM) [12], the kernel transformation is supposed to map not linearly separable classes in X to (approximately) linearly separable classes in Y . Equivalently, in kernelized clustering the kernel transformation is supposed to map clusters in X that are difficult to detect to clusters in Y that are easier to detect.

Mercer’s Theorem [11, 18] states that there is a mapping $\varphi : \mathbb{R}^p \rightarrow \mathbb{R}^q$ such that

$$k(x_j, x_k) = \langle \varphi(x_j), \varphi(x_k) \rangle \quad (8)$$

with the generalized dot product

$$\langle y_j, y_k \rangle = \sum_{i=1}^q y_{ji} y_{ki}. \quad (9)$$

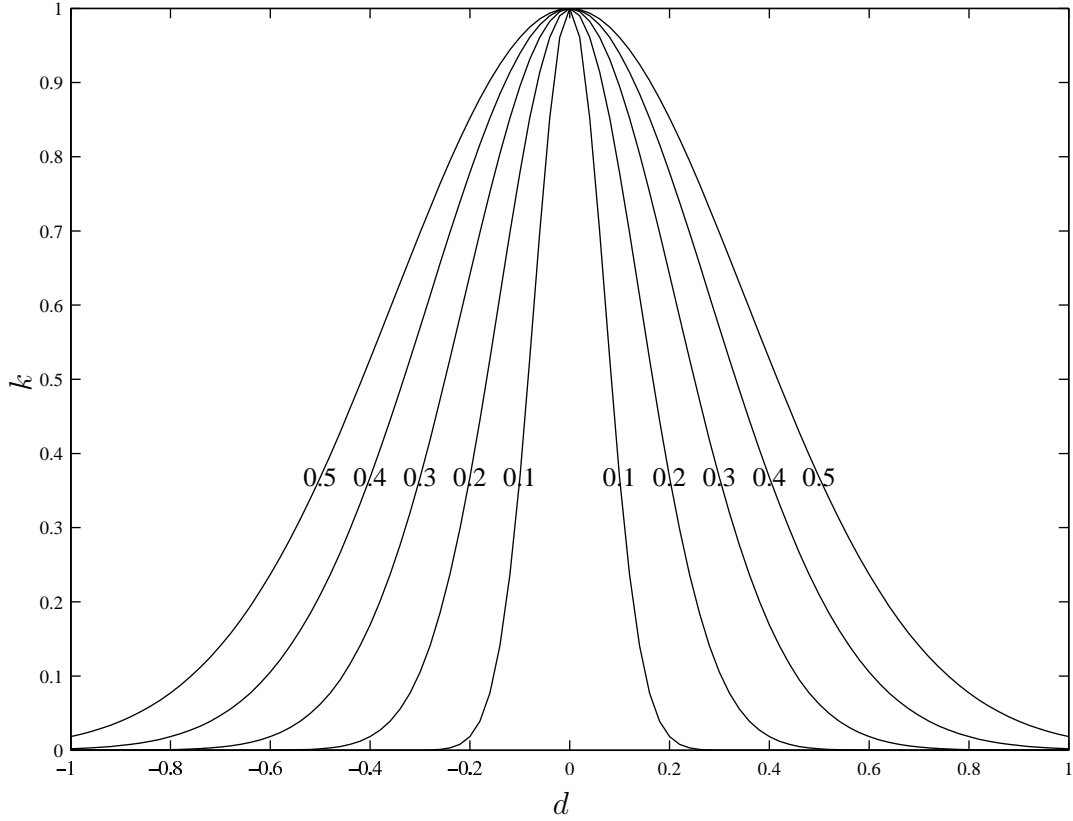


Figure 1: Gaussian kernel.

So practically, using Mercer's theorem a dot product in Y can be simply computed by evaluating a kernel function in X . In general, replacing dot products with kernels is therefore called the *kernel trick*. A popular family of kernel functions are Gaussian kernels

$$k(x_j, x_k) = e^{-\frac{\|x_j - x_k\|^2}{\sigma^2}}, \quad (10)$$

$\sigma \in \mathbb{R}^+$. Fig. 1 shows this function for the parameters $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

Kernelization of object (feature) data clustering models has been done for hard c-means (HCM) [19, 26, 3], fuzzy c-means (FCM) [23, 24, 22], and possibilistic c-means (PCM) [25]. Here, we focus on the kernelization of relational clustering models, in particular on the kernelization of NERFCM [8]. Assume that we can associate the relational data set R with an object data set X . Then, each element of the matrix R can be interpreted a distance between a pair of points in X . Following Mercer's theorem this distance can be computed in Y by a kernel in X :

$$r_{jk}^2 = \|\varphi(x_j), \varphi(x_k)\|^2 \quad (11)$$

$$= (\varphi(x_j), \varphi(x_k))^T (\varphi(x_j), \varphi(x_k)) \quad (12)$$

$$= \varphi(x_j)^T \varphi(x_j) - 2 \varphi(x_j)^T \varphi(x_k) + \varphi(x_k)^T \varphi(x_k) \quad (13)$$

$$= k(x_j, x_j) - 2 \cdot k(x_j, x_k) + k(x_k, x_k) \quad (14)$$

$$= 2 - 2 \cdot k(x_j, x_k), \quad (15)$$

under the assumption of $k(x, x) = 0$ for all $x \in \mathbb{R}^p$. So, kernelization of a relational clustering algorithm is equivalent to transforming the R to R' by (15) and then simply

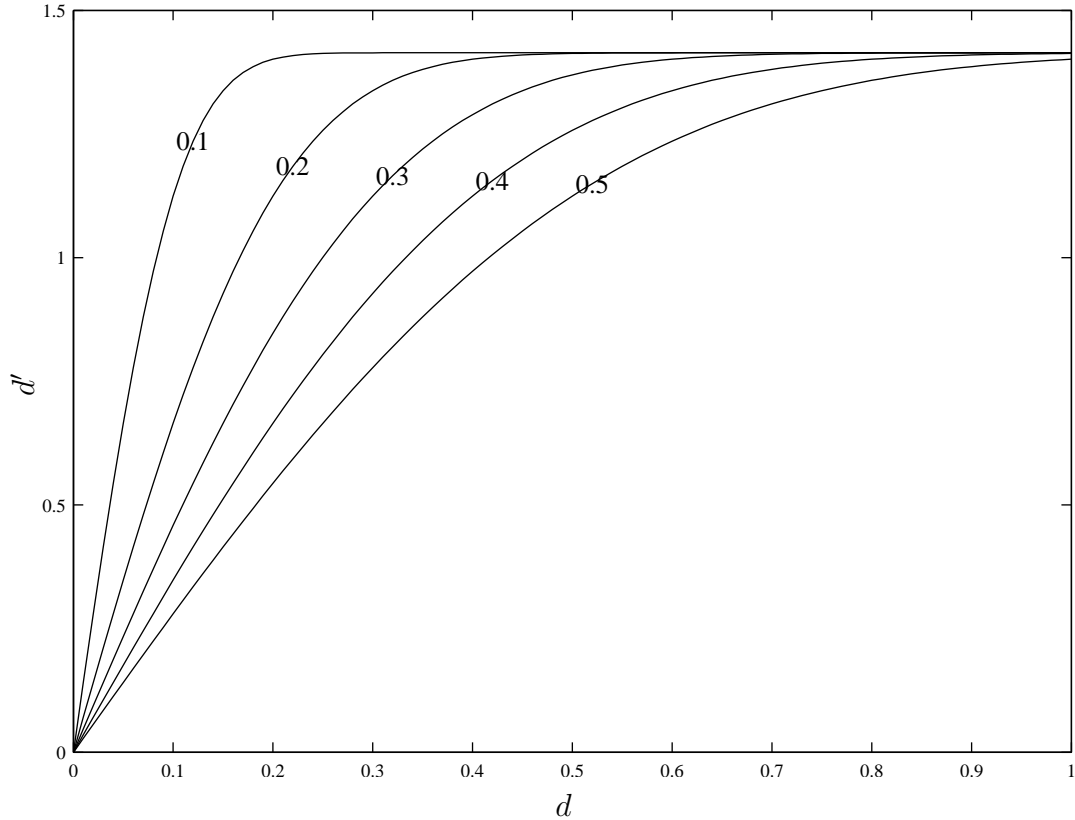


Figure 2: Kernel transformation.

applying the (unchanged) relational clustering algorithm to R' . In particular, for Gaussian kernels (10) we obtain

$$r'_{jk} = \sqrt{2 - 2 \cdot e^{-\frac{r_{jk}^2}{\sigma^2}}}. \quad (16)$$

Fig. 2 shows the Gaussian kernel transformation function $r'(r)$ for the parameters $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Using this transformation, large distances are softly clipped at the threshold of $\sqrt{2}$, and smaller distances are scaled by an almost constant factor. Relational clustering models such as RFCM and NERFCM are invariant against a scaling of the relational data by a constant factor $c \in \mathbb{R}^+$, i.e. we obtain the same results for the original data set R as for a scaled data set R^* with $r_{jk}^* = c \cdot r_{jk}$ for all $j, k = 1, \dots, n$. Therefore, each instance of the kernel transformation function $r'(r)$ can be scaled by an individual constant factor $c(\sigma)$ without affecting the clustering results. We scaled the Gaussian kernel transformation functions $r'(r)$ from Fig. 2 so that the slope at $r = 0$ is approximately equal to one, and obtained the normalized Gaussian kernel transformation functions shown in Fig. 3. These functions show that kernelization keeps small distances in R almost unchanged, but softly clips higher distances at the threshold σ . In the following section we will perform some experiments to investigate the effect of this soft clipping on the clustering results.

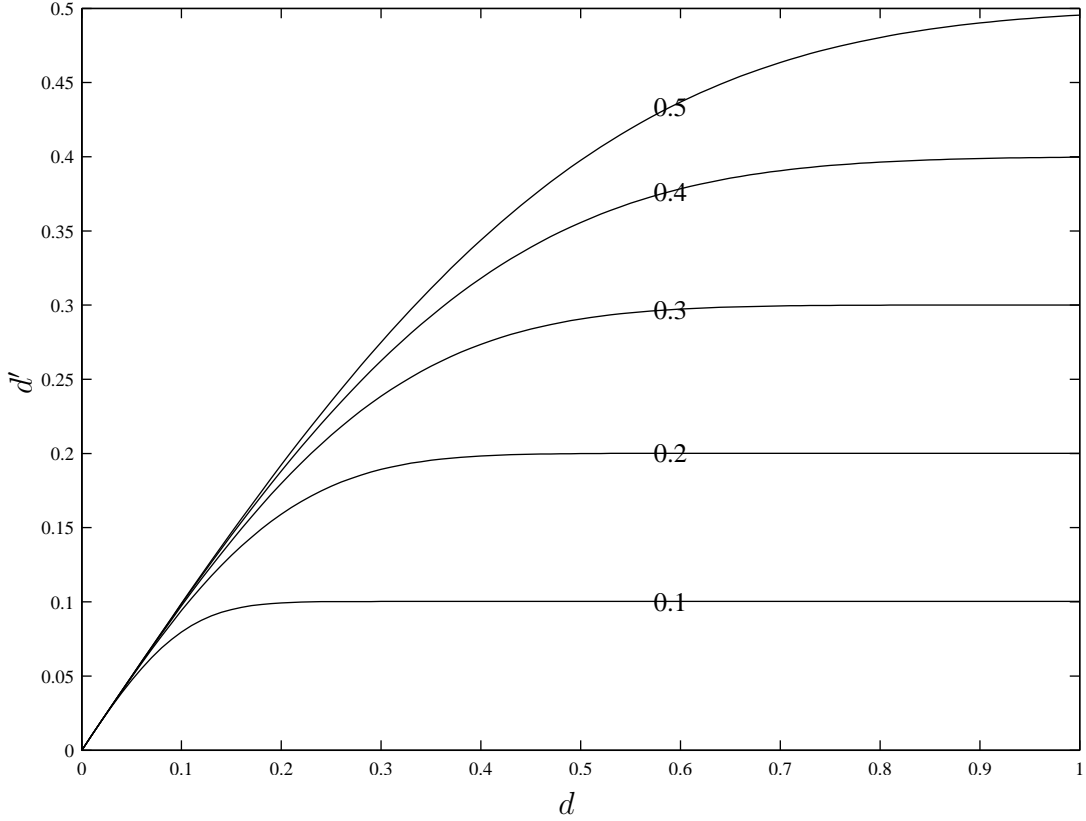


Figure 3: Normalized kernel transformation.

6 Experiments

To illustrate the effect of kernelization on relational fuzzy clustering we consider the following very simple object (feature) data set:

$$X(x, y) = \{(0, 1), (1, 0), (x, y)\}. \quad (17)$$

The first two data points are visualized in Fig. 4. The third data point (x, y) is located on the grid formed by the horizontal or the vertical lines in Fig. 4, more formally $x \in \{-2, -1.9, \dots, 3\}$ and $y \in [-2, 3]$, or $x \in [-2, 3]$ and $y \in \{-2, -1.9, \dots, 3\}$. For the resulting data set $X(x, y)$ from (17) we compute a relational data set $R(x, y)$ using Euclidean distances.

In the first experiment we use non-Euclidean relational fuzzy c -means (NERFCM), $c = 2$, $m = 2$, termination if

$$\max(\max(U_i - U_{i-1})) < 0.0001, \quad (18)$$

where $i \in \mathbb{N}$ is the iteration number, initialization of U as

$$U_0 = \begin{pmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \end{pmatrix}, \quad (19)$$

and apply it to $R(x, y)$. This yields resulting partions

$$U_t \approx \begin{pmatrix} 1 & 0 & u_{13} \\ 0 & 1 & u_{23} \end{pmatrix}. \quad (20)$$

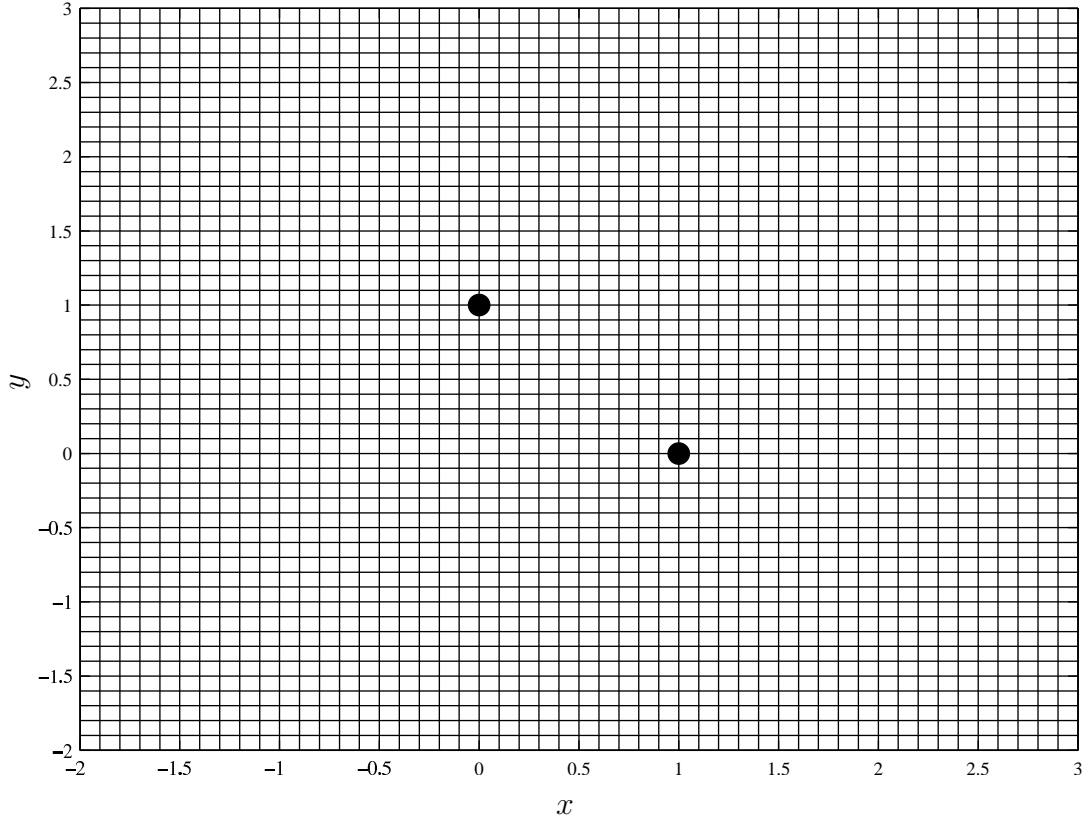


Figure 4: Experimental setup.

As a characteristic indicator of the clustering results we consider the values $u_{13}(x, y)$, i.e. the memberships of the third point in the first cluster associated with the data point $(1, 0)$. Notice that considering the other variable membership values $u_{23}(x, y)$ would capture the same information, since we simply have $u_{23}(x, y) = 1 - u_{13}(x, y)$. Fig. 5a shows the resulting values of u_{13} for (x, y) on the grid from Fig. 4. For data points (x, y) close to the first cluster center at $(1, 0)$ (left) the membership values u_{13} are close to one, and for data points (x, y) close to the second cluster center at $(0, 1)$ (right) the membership values u_{13} are close to zero. At the border line $x = y$ the distances of the data point (x, y) to both cluster centers at $(1, 0)$ and $(0, 1)$ are equal, so at this border line the membership values u_{13} sharply jump between zero and one. These observations exactly meet our expectations.

Next, we take the relational data sets $X(x, y)$ generated as described above and apply the Gaussian kernel transformation function $r'(r)$ from (16), yielding kernelized data sets $R'(x, y)$. These kernelized data sets are then clustered using NERFCM with the same parameters, initialization, and termination as above. Fig. 5b shows the results for $\sigma = 0.1$. For points (x, y) very close to the first cluster center at $(1, 0)$ we obtain memberships $u_{13}(x, y)$ of about one, and for points (x, y) very close to the second cluster center at $(0, 1)$ we obtain memberships $u_{13}(x, y)$ of about zero. For the remaining points (x, y) with a distance of more than about σ to both cluster centers the resulting memberships are about 0.5. Recalling Fig. 3, this effect is caused by the fact that the distances are softly clipped at $\sigma \approx 0.1$, hence the effective distances r_{13} are r_{23} are almost the same, more precisely $r_{13} \approx r_{23} \approx \sqrt{2}$, so these data points (x, y) receive (almost) the same

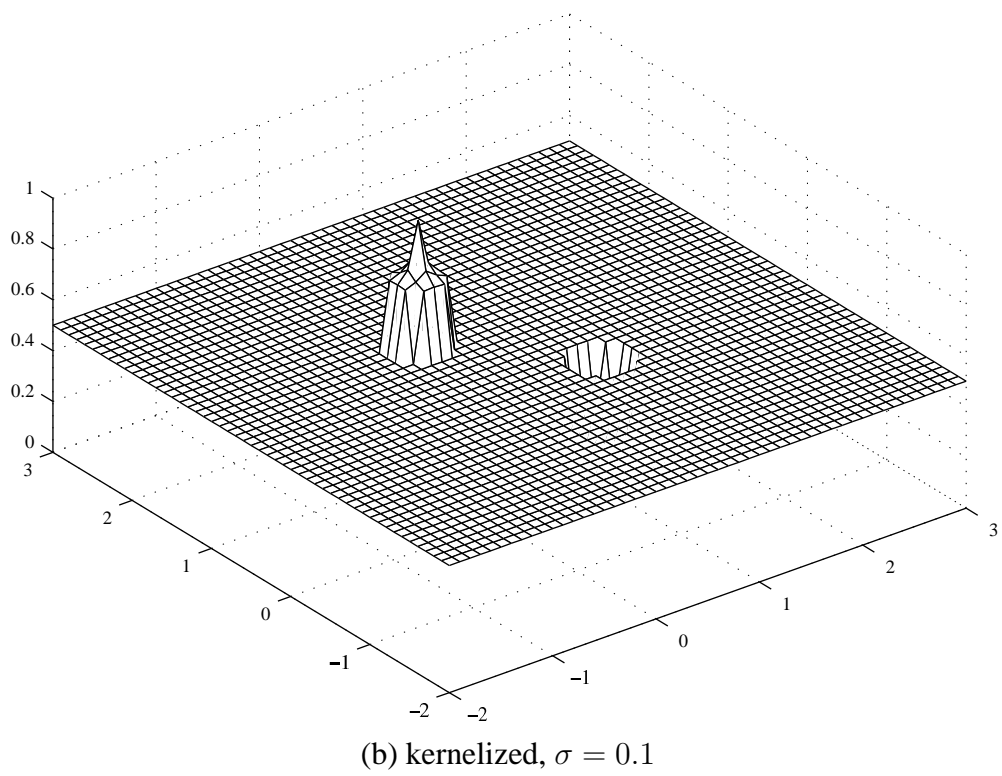
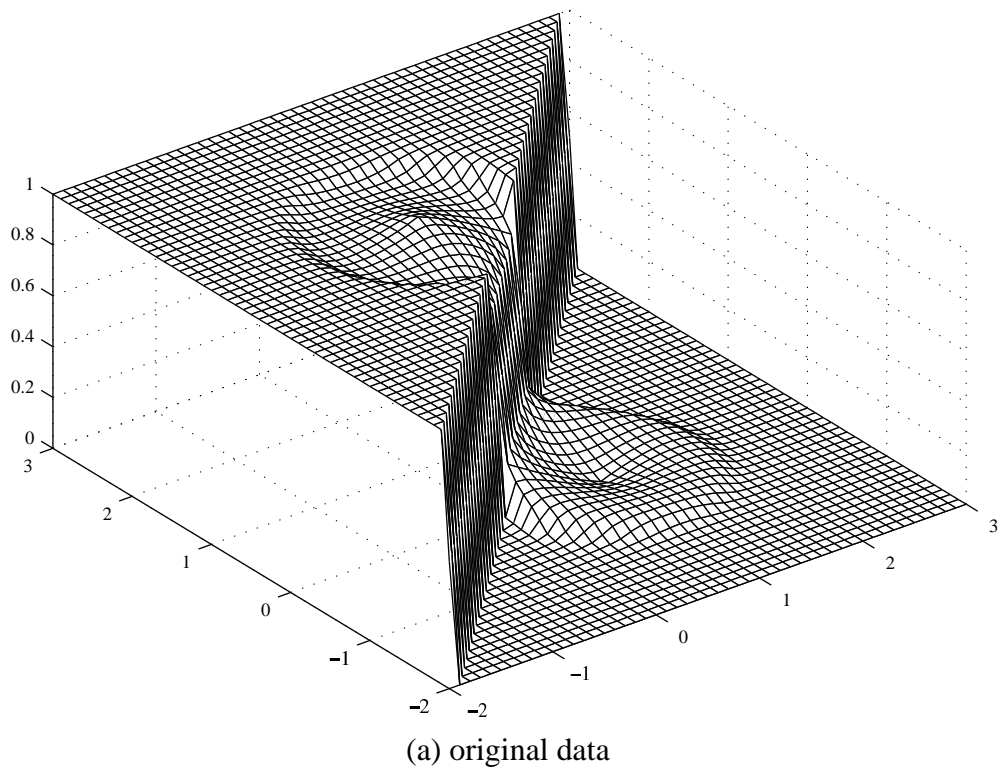
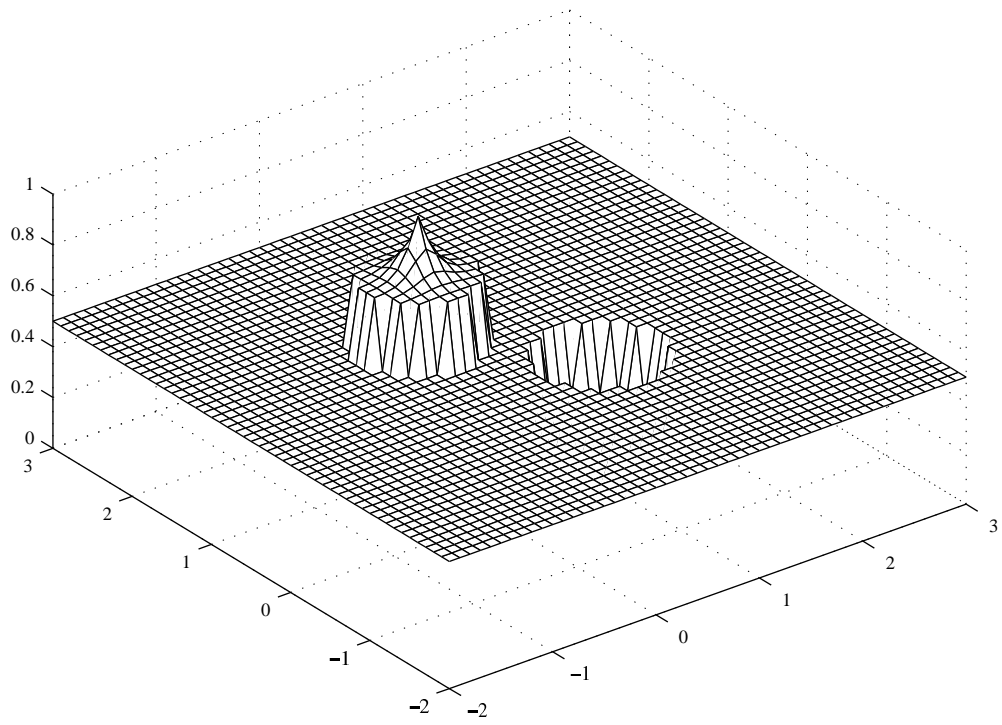
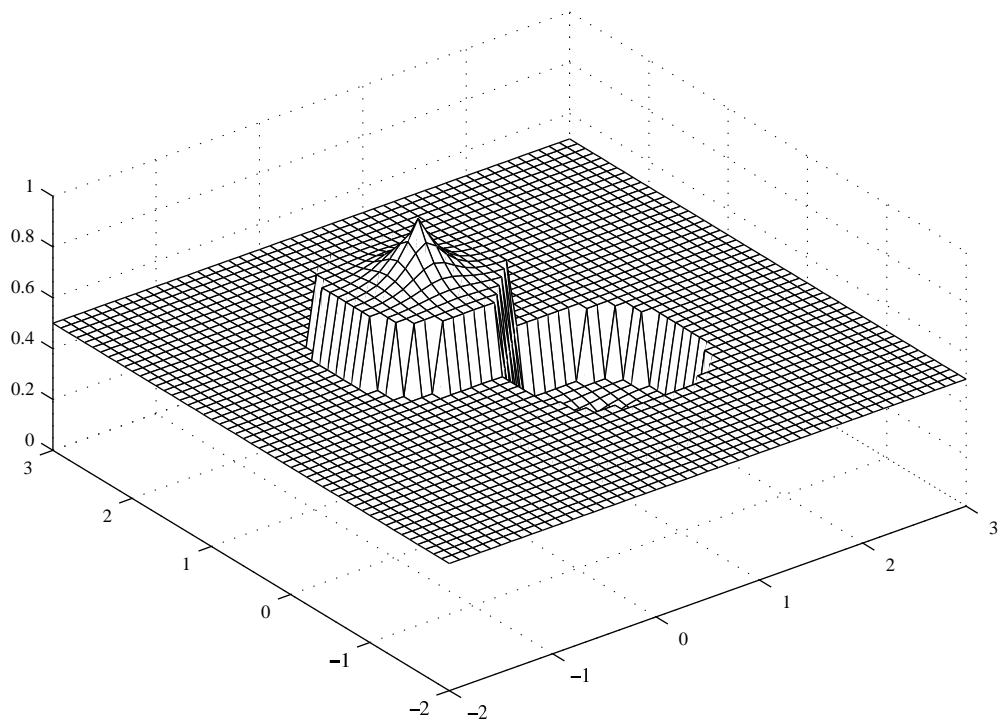


Figure 5: Membership values of an additional data point.

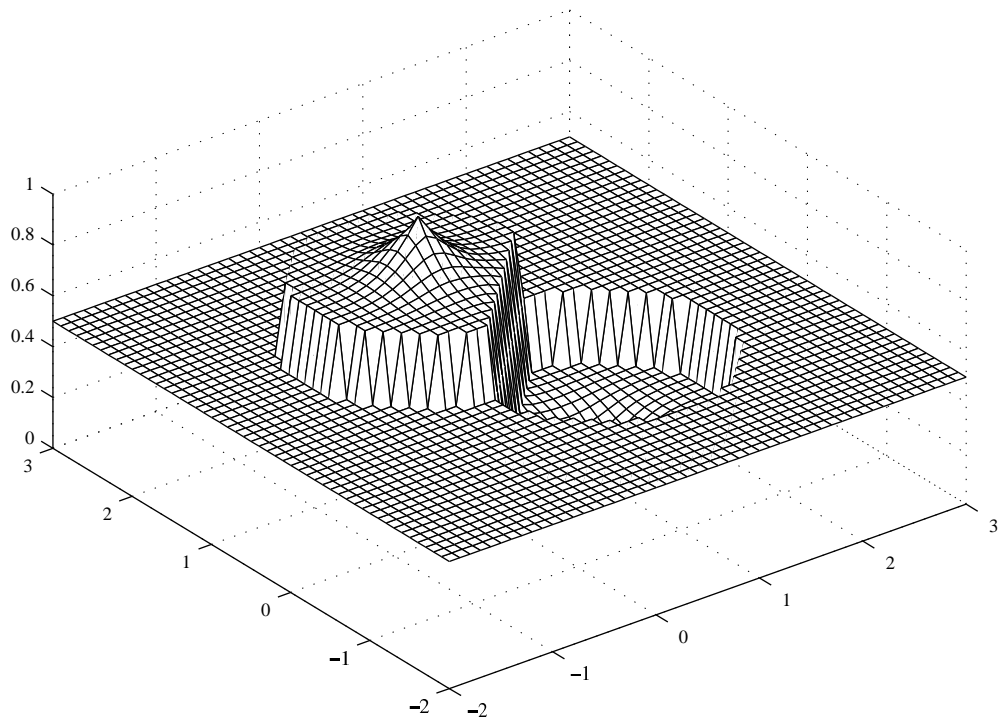


(a) kernelized, $\sigma = 0.2$

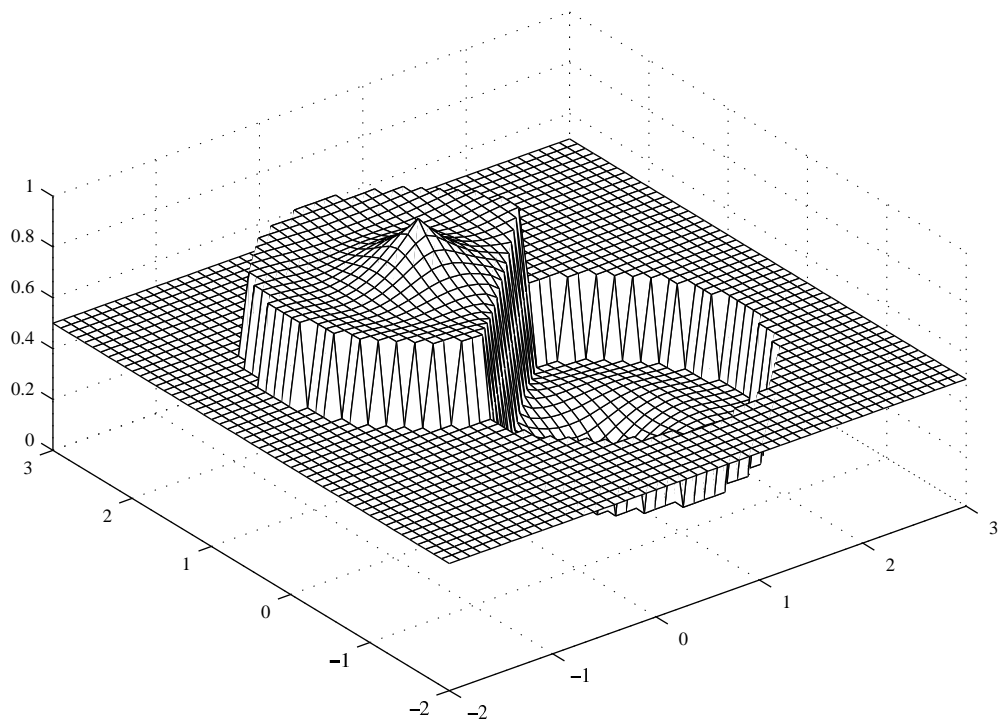


(b) kernelized, $\sigma = 0.3$

Figure 6: Membership values of an additional data point.



(a) kernelized, $\sigma = 0.4$



(b) kernelized, $\sigma = 0.5$

Figure 7: Membership values of an additional data point.

memberships $u_{13} \approx u_{23}$ in both clusters. Since $U \in M_{\text{fcn}}(1)$, i.e. $u_{13} + u_{23} = 1$ we further have $u_{13} \approx u_{23} \approx 0.5$, as shown in Fig. 5b.

Figs. 6 and 7 show the corresponding results for $\sigma \in \{0.2, 0.3, 0.4, 0.5\}$. In all cases unambiguous memberships of zero or one are obtained in circles around the cluster centers, where the radii of the circles are approximately equal to the parameter σ . This means that only data points within these circles around the cluster centers have an effect on the clustering result at all, all other data points are ignored and obtain memberships of $1/c$.

7 Conclusions

Kernelized relational clustering can be viewed as a two-step process: first the relational data set is transformed using a kernel transformation function, and second the transformed data set is clustered using the original relational clustering model. This two-step process allows an elegant analysis of the effect of the kernel transformation function. The kernel transformation function and its normalized version are equivalent, since relational clustering is invariant towards a constant scaling of the relational data. The (normalized) kernel transformation function leaves small distances almost unchanged and performs a soft clipping of larger distances, where the threshold depends of the kernel parameters. For the case of Gaussian kernels this threshold is equivalent to the variance parameter σ .

To analyze the effect of Gaussian kernelization to the actual clustering results we performed some experiments with a simple three point data set, where we examined the membership that the third (variable) data point obtained after kernel transformation and relational fuzzy clustering using the non-Euclidean relational fuzzy c -means (NERFCM) model. Our experiments indicate that data points are effectively only visible for the clustering algorithm if they are closer than σ from any cluster center. To ensure that all points in the vicinity of the cluster centers are considered in clustering, σ can be chosen in the order of magnitude as the expected distances between the cluster centers. Since these distances are hard to estimate a priori, a reasonable estimate for σ seems to be the median value of the distances in R . If σ is set to the median of R then *outliers* which have a large distance $\gg \sigma$ from the other data points (and the cluster centers) will be ignored. Therefore, kernelization in relational fuzzy clustering can be considered as a very elegant and effective way to eliminate outliers.

References

- [1] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [2] J. C. Bezdek and R. J. Hathaway. Clustering with relational c -means partitions from pairwise distance data. *International Journal of Mathematical Modelling*, 8:435–439, 1987.
- [3] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13:780–784, 2002.

- [4] E. E. Gustafson and W. C. Kessel. Fuzzy clustering with a covariance matrix. In *IEEE International Conference on Decision and Control, San Diego*, pages 761–766, 1979.
- [5] R. J. Hathaway and J. C. Bezdek. NERF c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition*, 27:429–437, 1994.
- [6] R. J. Hathaway and J. C. Bezdek. Optimization of clustering criteria by reformulation. *IEEE Transactions on Fuzzy Systems*, 3(2):241–245, May 1995.
- [7] R. J. Hathaway, J. W. Davenport, and J. C. Bezdek. Relational duals of the c-means algorithms. *Pattern Recognition*, 22:205–212, 1989.
- [8] R. J. Hathaway, J. M. Huband, and J. C. Bezdek. Kernelized non-Euclidean relational fuzzy c-means algorithm. In *IEEE International Conference on Fuzzy Systems*, pages 414–419, Reno, May 2005.
- [9] C. Katz, T. A. Runkler, and K. Heesche. Fuzzy clustering using similarity measures: Clustering relational data by object data methods. In *GMA/GI Workshop Fuzzy Systems and Computational Intelligence, Dortmund, Germany*, pages 46–58, November 2005.
- [10] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 2001.
- [11] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209:415–446, 1909.
- [12] K. R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12:181–201, 2001.
- [13] T. A. Runkler. Relational Gustafson Kessel clustering using medoids and triangulation. In *IEEE International Conference on Fuzzy Systems*, pages 73–78, Reno, May 2005.
- [14] T. A. Runkler. Relational fuzzy clustering. In J. Valente de Oliveira and W. Pedrycz, editors, *Advances in Fuzzy Clustering and its Applications*. Wiley, 2007.
- [15] T. A. Runkler and J. C. Bezdek. RACE: Relational alternating cluster estimation and the wedding table problem. In W. Brauer, editor, *Fuzzy-Neuro-Systems '98, München*, volume 7 of *Proceedings in Artificial Intelligence*, pages 330–337, March 1998.
- [16] T. A. Runkler and J. C. Bezdek. Alternating cluster estimation: A new tool for clustering and function approximation. *IEEE Transactions on Fuzzy Systems*, 7(4):377–393, August 1999.
- [17] T. A. Runkler and J. C. Bezdek. Web mining with relational clustering. *International Journal of Approximate Reasoning*, 32(2–3):217–236, February 2003.
- [18] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

- [19] B. Schölkopf, A.J. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [20] V. Tresp. Scaling kernel-based systems to large data sets. *Data Mining and Knowledge Discovery*, 5(3):197–211, 2001.
- [21] D. W. van der Merwe and A. P. Engelbrecht. Data clustering using particle swarm optimization. In *IEEE Congress on Evolutionary Computation*, pages 215–220, Canberra, Australia, 2003.
- [22] Z.-D. Wu, W.-X. Xie, and J.-P. Yu. Fuzzy c-means clustering algorithm based on kernel method. In *International Conference on Computational Intelligence and Multimedia Applications*, pages 49–54, Xi'an, China, 2003.
- [23] D.-Q. Zhang and S.-C. Chen. Fuzzy clustering using kernel method. In *International Conference on Control and Automation*, pages 123–127, 2002.
- [24] D.-Q. Zhang and S.-C. Chen. Clustering incomplete data using kernel-based fuzzy c-means algorithm. *Neural Processing Letters*, 18:155–162, 2003.
- [25] D.-Q. Zhang and S.-C. Chen. Kernel-based fuzzy and possibilistic c-means clustering. In *International Conference on Artificial Neural Networks*, pages 122–125, Istanbul, Turkey, 2003.
- [26] R. Zhang and A.I. Rudnicky. A large scale clustering scheme for kernel k-means. In *International Conference on Pattern Recognition*, pages 289–292, Quebec, Canada, 2002.

Modellierung und Simulation hydrostatischer Getriebe in mobilen Arbeitsmaschinen mittels Takagi-Sugeno-Fuzzy-Systemen

Horst Schulte

Bosch Rexroth AG, Mobilhydraulik, Glockeraustr.4, 89275 Elchingen
Tel.: 07308 82-2101, horst.schulte@boschrexroth.de

Hydrostatische Getriebe, auch bekannt als hydrostatische Fahrtriebe, werden aufgrund der kontinuierlich verstellbaren Übersetzung, der hohen Zugkraft bei niedrigen Antriebsgeschwindigkeiten und der aufgelösten Bauweise zunehmend in mobilen Arbeitsmaschinen eingesetzt.

In der industriellen Praxis werden bisher zur Regelung der Abtriebsdrehzahl und zur Anpassung der Antriebsleistung an die zur Verfügung stehende Dieselmotorleistung in Fahrtrieben mit hydrostatischen Getrieben einschleifige PID-Regelkreise eingesetzt. Zur Verbesserung der Antriebsdynamik wird in dieser Arbeit ein Takagi-Sugeno (TS) Fuzzyansatz zur Vorbereitung eines generalisierten nichtlinearen Reglerentwurfs für hydrostatische Getriebe abgeleitet. Es wird gezeigt, dass der vorgestellte Modellierungsansatz basierend auf Takagi-Sugeno-Fuzzy-Systemen die nichtlineare Dynamik und die Kopplung zwischen dem Druckaufbau im geschlossenen Hydraulikkreis und der Abtriebsdrehzahl adäquat nachbilden kann.

1 Einleitung

Bau-, Land- und Forstmaschinen sowie Flurförderfahrzeuge werden unter dem Begriff *mobile Arbeitsmaschinen* zusammengefasst. Mobile Arbeitsmaschinen sind sowohl stationäre Arbeitsmaschinen als auch Fahrzeuge. Während stationäre Maschinen im industriellen Bereich vorwiegend definierte, feste Arbeitsabläufe zyklisch durchlaufen unterliegen mobile Maschinen anderen Anforderungen:

1. für Fahr- und Arbeitsantriebe muss ein Energievorrat, welcher naturgemäß begrenzt ist, mitgeführt werden,
2. mobile Arbeitsmaschinen durchlaufen keine einheitlichen Arbeitszyklen aufgrund veränderlicher Umweltbedingungen bei wechselnden Aufgaben,
3. aufgrund neuer gesetzlicher Rahmenbedingungen ergeben sich steigende Anforderungen bezogen auf Verkehrssicherheit, Arbeitsschutz und Abgasreduktion.

Im Fahrtrieb bzw. Antriebsstrang mobiler Arbeitsmaschinen werden häufig hydrostatische Getriebe eingesetzt. Diese ermöglichen eine stufenlose Variation der Fahrgeschwindigkeit vom Stillstand bis zur Maximalgeschwindigkeit, ein ruckfreies Umkehren der Fahrtrichtung (Reversierbetrieb) und ein verschleißfreies Abbremsen. Gegenüber Antrieben mit hydrodynamischen Wandlern sind die mit hydrostatischen Fahrtrieben eingestellten Übersetzungen kaum lastabhängig. Im unteren Geschwindigkeitsbereich kann damit ein sehr feinfühliges Traktionsverhalten realisiert werden, welches die Handhabung von mobilen Arbeitsmaschinen erheblich erleichtert. Im Antriebsstrang, bestehend aus

Dieselmotor, Getriebe, Kardanwelle und Antriebsrädern, tragen hydrostatische Getriebe entweder zur direkten stufenlosen Leistungsübertragung bei oder sie sind Teil eines Überlagerungsgetriebes. In Überlagerungsgetrieben wird die Leistung über einen mechanischen und hydrostatischen Pfad verweigert, wodurch bei entsprechender Auslegung der Gesamtwirkungsgrad erhöht wird. In beiden Fällen kann durch die geeignete Wahl der Ansteuerung und die realisierte Reglerstruktur die Fahrdynamik und Energieeffizienz von hydrostatischen Antrieben entscheidend verbessert werden.

In der industriellen Praxis werden bisher zur Anpassung der Antriebsleistung an die zur Verfügung stehende Dieselmotorleistung und zur Regelung der Abtriebsdrehzahl bzw. Fahrzeuggeschwindigkeit eine Kombination aus einschleifigen PID-Regelkreisen eingesetzt. Um den oben genannten steigenden Anforderungen mobiler Arbeitsmaschinen jedoch gerecht werden zu können, müssen die physikalischen Zusammenhänge zwischen den Getriebedrehzahlen, den hydraulischen Drücken und den Schwenkwinkeln systematisch untersucht und im Regelgesetz in Form einer nichtlinearen Mehrgrößenregelung abgebildet werden.

In dieser Arbeit wird zur Vorbereitung des Entwurfs einer nichtlinearen Mehrgrößenregelung zunächst ein vereinfachtes für die Echtzeitregelung angepasstes mathematisches Modell hydrostatischer Getriebe vorgestellt. Dieses wird anschließend in eine Takagi-Sugeno-Fuzzy-System Formulierung überführt. Dabei bedienen wir uns der Methode von [3], analog zum Vorgehen in [7]. Es wird gezeigt, dass das vorgestellte TS-Fuzzy Modell für einen modellbasierten Reglerentwurf die nichtlineare Dynamik und die Kopplung zwischen dem Druckaufbau im geschlossenen Hydraulikkreis und der Hydromotordrehzahl bzw. Abtriebsdrehzahl adäquat nachbilden kann.

2 Funktionsprinzip und physikalische Modellierung hydrostatischer Getriebe

Der prinzipielle Aufbau vom Antriebsstrang in mobilen Arbeitsmaschinen ist in Bild 1 dargestellt. Ein Dieselmotor treibt eine hydraulische Verstellpumpe an, die mit zwei Anschlüssen durch Rohr- oder Schlauchleitungen mit einem Hydromotor verbunden ist. Der Hydromotor ist entweder direkt oder über ein mechanisches Getriebe mit den Antriebsrädern verbunden. Der geschlossene hydraulische Kreis bestehend aus Hydropumpe und -motor bildet den hydraulischen Grundkreislauf eines hydrostatischen Getriebes. Die meist verbreitetste Bauart von Pumpen und Motoren in mobilen Arbeitsmaschinen ist die Axialkolbenmaschine. Wegen der hohen Leistungsdichte und konstruktiv relativ einfachen Realisierung der Verstellbarkeit des Hubvolumens ist diese Bauart besonders für stufenlose hydrostatische Getriebe geeignet.

Neben dem in Bild 1 dargestellten Grundkreislauf sind noch Ein- und Ausspeisungen zum Ausgleich der Leckverluste, die im Modellierungsansatz zunächst nicht berücksichtigt werden, sowie Druckbegrenzungsventile und Rückschlagventile, die jedoch unter normalen Betriebsbedingungen nicht ansprechen, in einem hydrostatischen Getriebe enthalten, siehe auch [4].

Über ein Fahrpedal und den Richtungsschalter gibt der Fahrer die gewünschte Getriebeübersetzung oder Fahrzeuggeschwindigkeit vor (vgl. Bild 1). Zur Regelung wird die Eingangsdrehzahl, welche der Hydropumpendrehzahl n_P entspricht, die Ausgangsdrehzahl, welche der Hydromotordrehzahl n_M entspricht und die Drücke p_A und p_B im geschlossenen Hydraulikkreis erfasst. Dabei wird durch kontinuierliche Verstellung der Pumpe

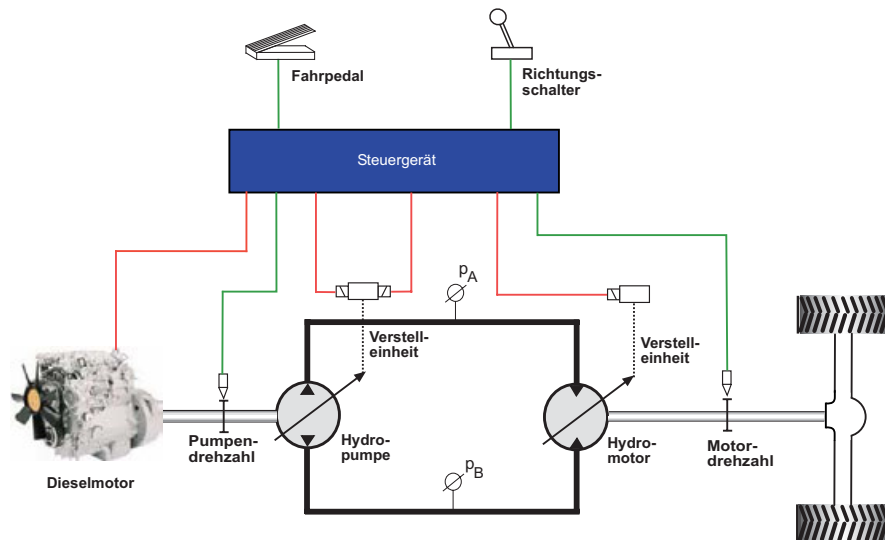


Bild 1: Aufbau und Komponenten elektronisch geregelter hydrostatischer Fahrtriebe

der geförderte Volumenstrom q_P in Menge und Richtung so variiert, dass sich durch Vergleich mit den gemessenen Regelgrößen die gewünschte Fahrzeuggeschwindigkeit oder das gewünschte Übersetzungsverhältnis einstellt. Ergänzend wird zur Verhinderung einer unzulässig hohen Absenkung der Dieseldrehzahl die Pumpenleistung mittels Reduktion der Fördermenge durch Verstellung des Verdrängungsvolumens V_P an die zur Verfügung stehende Dieselmotorleistung angepasst. Die grundlegenden physikalischen Zusammenhänge lassen sich in einfacher Form wie folgt angeben:

Der Volumenstrom q_P der von der Pumpe gefördert ergibt sich aus dem verstellbaren Verdrängungsvolumens $V_P(i_P)$ pro Umdrehung und der Drehzahl n_P

$$q_P = V_P(i_P) n_P \quad (1)$$

mit i_p als Gesamtstrom durch die Proportionalmagnete der Servoventile der hydraulischen Verstell-einheit. Hierbei muss beachtet werden, dass je Schwenkwinkelrichtung (positiv/negativ) immer nur ein Magnet Strom führt und die Zählpfeilrichtung der Ströme ohne Beschränkung der Allgemeinheit entgegengesetzt festgelegt wird. Die Verstellung des Verdrängungsvolumens $V_P(i_P)$ sowohl in positiver wie in negativer Richtung bewirkt eine Förderrichtungsumkehr und erlaubt eine nahezu ruckfreie Umkehrung der Fahrtrichtung. Aufgrund der hydromechanischen Rückkopplung zwischen Servoventil und Verstellzylinder in den Axialkolbenmaschinen kann in erster Näherung der Zusammenhang zwischen i_p und dem Verdrängungsvolumen durch ein Totband (Berücksichtigung von Haftreibungseinflüssen im Stellzylinder und der Schwenkwinkelmechanik) multipliziert mit einem linearen Verstärkungsfaktor k_P dargestellt werden, vgl. Bild 2 Der Volumenstrom der den Motor antreibt ergibt sich aus dem Volumen, welches der Motor pro Umdrehung "schluckt"

$$q_M = V_M(i_M) n_M \quad (2)$$

mit i_M als Strom durch den Proportionalmagnet des Servoventils der hydraulischen Verstell-einheit vom Hydromotor. Im Gegensatz zur Hydropumpe kann der Motor nur in eine Richtung verstellt werden, d.h. das Verdrängungsvolumen des Motors ist stets positiv. Betrachtet man zunächst den verlustfreien Betrieb im geschlossenen Kreis so kann

im statischen Fall mit $q_P = q_M$ ein verstellbares Übersetzungsverhältnis $r(i_P, i_M)$ von Ausgangs- zu Eingangsrehzahl

$$r(i_P, i_M) = \frac{n_M}{n_P} = \frac{V_P(i_P)}{V_M(i_M)} \quad . \quad (3)$$

angegeben werden. Die *Dynamik* in Hydrostatischen Getrieben wird weitestgehend bestimmt durch den Kompressibilitätsvolumenstrom q_c zwischen Pumpe und Motor aufgrund der Kompressibilität des Arbeitsmediums Öl und durch die Aufweitungen der Hydraulikleitungen im geschlossenen Kreis. Dieser wird für den unteren und oberen Hydraulikkreis (bezogen auf Bild 1) mit Hilfe der konzentrierten hydraulischen Kapazitäten C_A und C_B berechnet, welche die Kapazität des Öls in den Leitungen und die Kapazität der Leitungen berücksichtigen:

$$q_{C_A} = C_A \dot{p}_A, \quad q_{C_B} = C_B \dot{p}_B \quad (4)$$

Hinweis: Die hydraulische Kapazität ist im Allgemeinen nicht konstant und nimmt mit zunehmenden Druck p ab. In vielen technischen Anordnungen ist der Einfluss jedoch gegenüber anderen Einflüssen gering und die Linearisierung um einen Arbeitspunkt ist ausreichend für einen weiten Arbeitsbereich womit die Annahme $C_{A,B} = const$ berechtigt ist. Durch die Einbeziehung des Kompressibilitätsvolumenstroms im unteren und oberen Leitungszweig erhalten wir aus den Volumenstrombilanzen die Beziehungen

$$q_P - q_M = q_{C_A}, \quad (5a)$$

$$-q_P + q_M = q_{C_B}. \quad (5b)$$

Die verlustlos aufgenommene oder abgegebene Leistung am Hydromotor beträgt

$$P = M_M \omega_M = (p_A - p_B) q_M. \quad (6)$$

Anstelle der Verdrängungsvolumina aus (1) und (2) ist es sinnvoll für die Untersuchung des dynamischen Verhaltens

$$q_P = \tilde{V}_P(i_P) \omega_P \quad \text{mit} \quad \tilde{V}_P = \frac{V_P}{2\pi}, \quad (7a)$$

$$q_M = \tilde{V}_M(i_M) \omega_M \quad \text{mit} \quad \tilde{V}_M = \frac{V_M}{2\pi} \quad (7b)$$

einzuführen, wobei die Volumenströme nicht mehr mit $[\text{cm}^3/\text{min}]$ sondern durch die Einheit $[\text{cm}^3/\text{s}]$ quantifiziert werden. Dann erhält man nach dem Einsetzen von (7b) in (6) das Getriebeausgangsmoment, welches dem Motormoment

$$M_M = \tilde{V}_M(i_M) (p_A - p_B) \quad (8)$$

entspricht. Somit verursacht die Druckdifferenz $p_A - p_B$ im Hydraulikkreis bei einem positiven Leistungsfluss von der Pumpe zum Motor ein Moment am Getriebeausgang, welches durch die Schwenkwinkeländerung des Hydromotors verstellt werden kann. Die Hydromotorwelle sei starr über ein Differentialgetriebe mit der Antriebsachse verbunden (vgl. Bild 1). Die Massenträgheitsmomente der Antriebsräder und der Einfluss der Fahrzeugmasse wird wie in [2, Kap.7] durch einen Parameter J_F zusammengefasst. Die Fahrzeugdämpfung in longitudinaler Richtung sei in dem Parameter d_F berücksichtigt. Aufgeprägte äußere Momente werden mit Hilfe des zeitabhängigen Lastmoments $M_L(t)$

repräsentiert. Basierend auf der Drehimpulsbilanz wird die Lastmechanik im Antriebsstrang angegeben mit

$$J_F \dot{\omega}_M + d_F \omega_M = M_M - M_L . \quad (9)$$

Nach dem Einsetzen des Motormoments (8) in (9) erhalten wir die Bewegungsgleichung des hydrostatischen Fahrtriebs

$$J_F \dot{\omega}_M + d_F \omega_M = \tilde{V}_M(i_M) (p_A - p_B) - M_L . \quad (10)$$

Die einzelnen Beziehungen werden nun in einem Gleichungssystem zusammengefügt. Einsetzen von (7a), (7b) in (5a), (5b) ergibt zunächst

$$\begin{aligned} \tilde{V}_P(i_P) \omega_P - \tilde{V}_M(i_M) \omega_M &= q_{C_A} \\ -\tilde{V}_P(i_P) \omega_P + \tilde{V}_M(i_M) \omega_M &= q_{C_B} \end{aligned} \quad (11)$$

Berücksichtigt man nun noch die Kompressibilitätsflüsse erhalten wir durch einsetzen in (4) die Druckaufbaugleichungen des geschlossenen Hydraulikkreises

$$\begin{aligned} \dot{p}_A &= \frac{1}{C_A} \left(\tilde{V}_P(i_P) \omega_P - \tilde{V}_M(i_M) \omega_M \right) \\ \dot{p}_B &= -\frac{1}{C_B} \left(\tilde{V}_P(i_P) \omega_P - \tilde{V}_M(i_M) \omega_M \right) \end{aligned} \quad (12)$$

Wie schon weiter oben erwähnt worden ist, muss zur Quantifizierung der Wirkungskette zwischen dem Magnetstrom $i_{P,M}$ und dem einzustellenden Verdrängungsvolumen $\tilde{V}_{M,P}(i_{P,M})$ ein Totband berücksichtigt werden. In der industriellen Praxis hat es sich bewährt dieses durch die entsprechende inverse Diskontinuität mittels Vorsteuerung, welche typischerweise in einem elektronischen Steuergerät berechnet wird, zu kompensieren [1]. Sei $\tilde{V}_{M,P} = g_{P,M}(i_{P,M})$ mit $u_{P,M}$ als Eingänge der Vorsteuerung, dann wird durch die Berechnung der inversen Diskontinuitäten

$$i_P = g_P^{-1}(u_P) , \quad i_M = g_M^{-1}(u_M) \quad (13)$$

die statischen Nichtlinearitäten der Verstelleinheiten im Steuergerät (bei Vernachlässigung der dynamischen Hysterese) kompensiert.

Im letzten Schritt der Herleitung werden die Gleichungen (10), (12) und (13) zusammengefasst in einem Zustandsraummodell der Form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$:

$$\begin{aligned} \dot{x}_1 &= \frac{1}{C_A} (u_1 \omega_P - u_2 x_3) \\ \dot{x}_2 &= -\frac{1}{C_B} (u_1 \omega_P - u_2 x_3) \\ \dot{x}_3 &= -\frac{1}{J_F} (d_F x_3 - (x_1 - x_2) u_1 + M_L) \end{aligned} \quad (14)$$

mit dem Zustands- und Eingangsvektor

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} := \begin{bmatrix} p_A \\ p_B \\ \omega_M \end{bmatrix} , \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} := \begin{bmatrix} u_P \\ u_M \end{bmatrix} \quad (15)$$

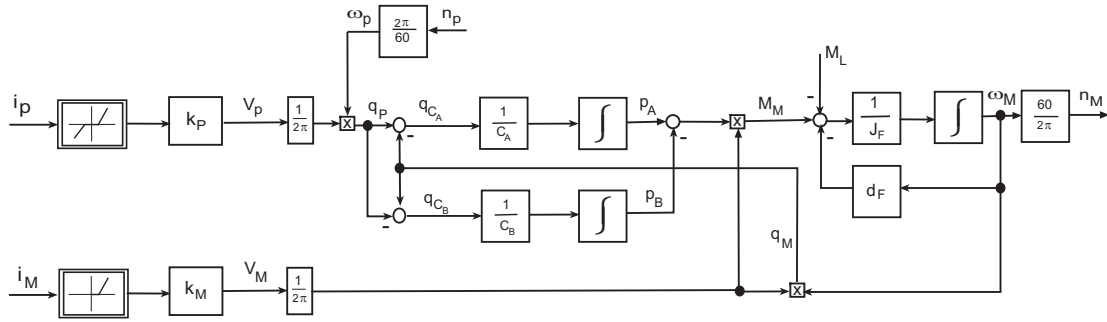


Bild 2: Blockschaltbild hydrostatischer Getriebe basierend auf den Modellgleichungen (1) - (12)

3 Herleitung des Takagi-Sugeno-Fuzzy Modells

Die Umformung eines nichtlinearen Zustandsraummodells in ein Takagi-Sugeno-Fuzzy Modell [8] nach [3] wird in zwei Schritten vollzogen. Im *ersten Schritt* wird das nichtlineare Zustandsraummodell in ein linear parameterveränderliches (LPV-)System überführt. Man erhält nach kurzer Umstellung aus (14) die folgende, für die weitere Umformung in ein TS-Fuzzy-System günstige Realisierung¹

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{1}{C_A} u_2 \\ 0 & 0 & \frac{1}{C_B} u_2 \\ \frac{1}{J} u_1 & 0 & -\frac{1}{J_F} (d_F + \frac{M_L}{x_3}) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} -\frac{\omega_P}{C_A} & 0 \\ -\frac{\omega_P}{C_B} & 0 \\ 0 & -\frac{1}{J_F} x_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (16)$$

Solch ein System wird in der Literatur [5] auch als quasi-LPV-System bezeichnet, da die Änderung der Matrizenkoeffizienten nicht durch physikalische Parameteränderungen sondern, wie z.B. in diesem Fall durch die in den Koeffizienten enthaltenden Systemeingänge u_1 , u_2 , die Pumpendrehzahl ω_P und durch das äußere Lastmoment M_L hervorgerufen wird. Mit dieser Sprechweise grenzt man sich gegenüber den LPV-Systemen ab, die tatsächlich Änderungen physikalischer Parameter in den Beschreibungsgleichungen, wie z.B. veränderliche Massen- oder Dämpfungskoeffizienten, enthalten.

Im *zweiten Schritt* wird mit der Festlegung, dass die Änderung der Dieseldrehzahl um die Nenndrehzahl und das äußere Lastmoment M_L als Störung aufgefasst werden ein Takagi-Sugeno-Fuzzy Modell mit $N_r = 8$ linearen Modellen aufgestellt. Seien die Eingänge u_1 , u_2 und der Zustand x_2 beschränkt

$$u_1 \in [\underline{u}_1, \bar{u}_1], \quad u_2 \in [\underline{u}_2, \bar{u}_2], \quad x_2 \in [\underline{x}_2, \bar{x}_2], \quad (17a)$$

dann gilt mit $\underline{f}_1 := \underline{u}_1$ und $\bar{f}_1 := \bar{u}_1$

$$u_1 = f_1(u_1) = \underline{f}_1 \underbrace{\frac{f_1 - u_1}{f_1 - \underline{f}_1}}_{=w_{11}(u_1)} + \bar{f}_1 \underbrace{\frac{u_1 - \bar{f}_1}{\bar{f}_1 - \underline{f}_1}}_{=w_{12}(u_1)}. \quad (17b)$$

Analog ergeben sich für den Eingang u_2 mit $\underline{f}_2 := \underline{u}_2$ und $\bar{f}_2 := \bar{u}_2$

$$u_2 = f_2(u_2) = \underline{f}_2 \underbrace{\frac{f_2 - u_2}{f_2 - \underline{f}_2}}_{=w_{21}(u_2)} + \bar{f}_2 \underbrace{\frac{u_2 - \bar{f}_2}{\bar{f}_2 - \underline{f}_2}}_{=w_{22}(u_2)}. \quad (17c)$$

¹Denn die Umformung aus (14) in (16) ist nicht eindeutig.

und den Zustand x_2 mit $\underline{g}_1 := \underline{x}_2$ und $\bar{g}_1 := \bar{x}_2$

$$x_2 = g_1(x_2) = \underline{g}_1 \underbrace{\frac{g_1 - x_2}{\bar{g}_1 - g_1}}_{=w_{31}(x_2)} + \bar{g}_1 \underbrace{\frac{x_2 - \bar{g}_1}{\bar{g}_1 - g_1}}_{=w_{32}(x_2)} \quad (17d)$$

die Sektorfunktionen w_{ij} mit $i = 1, 2, 3$ und $j = 1, 2$. Aus der Verknüpfung

$$\{w_{11}, w_{12}\} \times \{w_{21}, w_{22}\} \times \{w_{31}, w_{32}\} \quad (18)$$

ergeben sich mit (17b), (17c) und (17d) die Zugehörigkeitsfunktionen $h_i(u_1, u_2, x_2)$, $i = 1, \dots, 8$ der einzelnen System- und Eingangsmatrizen \mathbf{A}_i und \mathbf{B}_i mit

$$\begin{aligned} h_1(u_1, u_2, x_2) &= w_{11}(u_1) \cdot w_{21}(u_2) \cdot w_{31}(x_2) , \\ h_2(u_1, u_2, x_2) &= w_{11}(u_1) \cdot w_{21}(u_2) \cdot w_{32}(x_2) , \\ h_3(u_1, u_2, x_2) &= w_{11}(u_1) \cdot w_{22}(u_2) \cdot w_{31}(x_2) , \\ h_4(u_1, u_2, x_2) &= w_{11}(u_1) \cdot w_{22}(u_2) \cdot w_{32}(x_2) , \\ h_5(u_1, u_2, x_2) &= w_{12}(u_1) \cdot w_{21}(u_2) \cdot w_{31}(x_2) , \\ h_6(u_1, u_2, x_2) &= w_{12}(u_1) \cdot w_{21}(u_2) \cdot w_{32}(x_2) , \\ h_7(u_1, u_2, x_2) &= w_{12}(u_1) \cdot w_{22}(u_2) \cdot w_{31}(x_2) , \\ h_8(u_1, u_2, x_2) &= w_{12}(u_1) \cdot w_{22}(u_2) \cdot w_{32}(x_2) . \end{aligned}$$

Die System- und Eingangsmatrizen \mathbf{A}_i und \mathbf{B}_i , $i = 1, \dots, N_r$ können kompakt dargestellt mit Hilfe der Notation aus [7]

$$\mathbf{A}_i = \begin{bmatrix} 0 & 0 & -\frac{1}{C_A} *_2 \\ 0 & 0 & \frac{1}{C_B} *_2 \\ \frac{1}{J} *_1 & 0 & -\frac{1}{J_F} d_F \end{bmatrix} , \quad \mathbf{B}_i = \begin{bmatrix} -\frac{\omega_P}{C_A} & 0 \\ -\frac{\omega_P}{C_B} & 0 \\ 0 & -\frac{1}{J_F} *_3 \end{bmatrix} \quad (19)$$

wobei $*_1 \in \{\underline{u}_2, \bar{u}_2\}$, $*_2 \in \{\underline{u}_1, \bar{u}_1\}$ und $*_3 \in \{\underline{x}_2, \bar{x}_2\}$. Die Koeffizienteneinträge der konstanten Matrizen ergeben sich aus den möglichen Kombinationen der Verknüpfung

$$\{\underline{u}_2, \bar{u}_2\} \times \{\underline{u}_1, \bar{u}_1\} \times \{\underline{x}_2, \bar{x}_2\} . \quad (20)$$

Zum Beispiel ist

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 & -\frac{1}{C_A} \underline{u}_2 \\ 0 & 0 & \frac{1}{C_B} \underline{u}_2 \\ \frac{1}{J} \underline{u}_1 & 0 & -\frac{1}{J_F} d_F \end{bmatrix} , \quad \mathbf{B}_1 = \begin{bmatrix} -\frac{\omega_P}{C_A} & 0 \\ -\frac{\omega_P}{C_B} & 0 \\ 0 & -\frac{1}{J_F} \underline{x}_2 \end{bmatrix} \quad (21)$$

und

$$\mathbf{A}_5 = \begin{bmatrix} 0 & 0 & -\frac{1}{C_A} \underline{u}_2 \\ 0 & 0 & \frac{1}{C_B} \underline{u}_2 \\ \frac{1}{J} \bar{u}_1 & 0 & -\frac{1}{J_F} d_F \end{bmatrix} , \quad \mathbf{B}_5 = \mathbf{B}_1 . \quad (22)$$

Damit ist das Takagi-Sugeno-Fuzzy Modell [8]

$$\dot{\mathbf{x}} = \sum_{i=1}^8 h_i(u_1, u_2, x_2) \mathbf{A}_i \mathbf{x} + \sum_{i=1}^8 h_i(u_1, u_2, x_2) \mathbf{B}_i \mathbf{u} \quad (23)$$

mit

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} := \begin{bmatrix} p_A \\ p_B \\ \omega_M \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} := \begin{bmatrix} u_P \\ u_M \end{bmatrix}$$

vollständig, vgl. (15). Im nächsten Kapitel werden erste Simulationsergebnisse in Form von Druck- und Ausgangsdrehzahlverläufen bei gleichzeitiger Hydropumpen- und Hydromotorverstellung (simultaner Primär- und Sekundärverstellung) mit und ohne Drehzahleingangsschwankungen präsentiert.

4 Simulationsergebnisse

Es werden nun erste Simulationsergebnisse mit dem im letzten Kapitel abgeleiteten Takagi-Sugeno Modell präsentiert. Die Modellstruktur (23) wurde hierbei so aufbereitet, dass diese innerhalb der Simulationsumgebung Matlab/Simulink in einer sogenannten S-Function berechnet werden kann. Hierzu wurden für jeden Integrationssschritt die gewichteten Matrixensummen in Abhängigkeit von den aktuellen Eingangswerten- und dem Zustand x_2 ermittelt und dem Integrationsalgorithmus der S-Function übergeben.

Simuliert wurde ein typischer Beschleunigungsvorgang vom Stillstand bis zu einer Abtriebsdrehzahl von ca. $n_M = 4100$ [rpm], vgl. Bild (3). Im oberen Diagramm sind die zwei Eingangsgrößen u_P und u_M zum Verstellen der Verdrängungsvolumen der Hydropumpe (Födervolumen) und des Hydromotors (Schluckvolumen) über die Zeit aufgetragen. Vom Stillstand bis $t = 3$ s wird die Pumpe aus Null bis zum maximalen Verdrängungsvolumen verstellt. Dies verursacht bei konstanter Eingangsdrehzahl n_P einen Differenzdruckanstieg im geschlossenen Kreislauf, wodurch sich eine kontinuierlich ansteigende Abtriebsdrehzahl ergibt. Ab $t = 3$ s wird zur Erweiterung des Geschwindigkeitsstellbereichs das Verdrängungsvolumen des Motors mittels u_M reduziert, wodurch die Hydromotor- bzw. Abtriebsdrehzahl weiter ansteigt.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurde zur Vorbereitung des Entwurfs einer nichtlinearen Mehrgrößenregelung zunächst ein mathematisches Modell eines hydrostatischen Getriebes vorgestellt. Dieses wurde anschließend in eine Takagi-Sugeno (TS) Fuzzy System Formulierung überführt. Dabei bedienen wir uns der Methode von [3] analog zum Vorgehen in [7].

In den nun folgenden Schritten wird zur Modellreduktion ein approximatives Verfahren aus [6] angewandt. Durch weitere Simulationen sollen die beiden hergeleiteten TS Modelle mit einem vorhandenen erheblich detaillierterem Simulationsmodell, welches sich aber aufgrund der hohen Ordnung und der diskontinuierlichen mehrdimensionalen Kennfelder nicht für den Reglerentwurf eignet, verglichen werden.

Literatur

- [1] Bosch Rexroth AG: *Datenblatt zur Axialkolben-Verstellpumpe A4VG (RD 92 003)*, <http://www.boschrexroth.com/brm>, 2006.

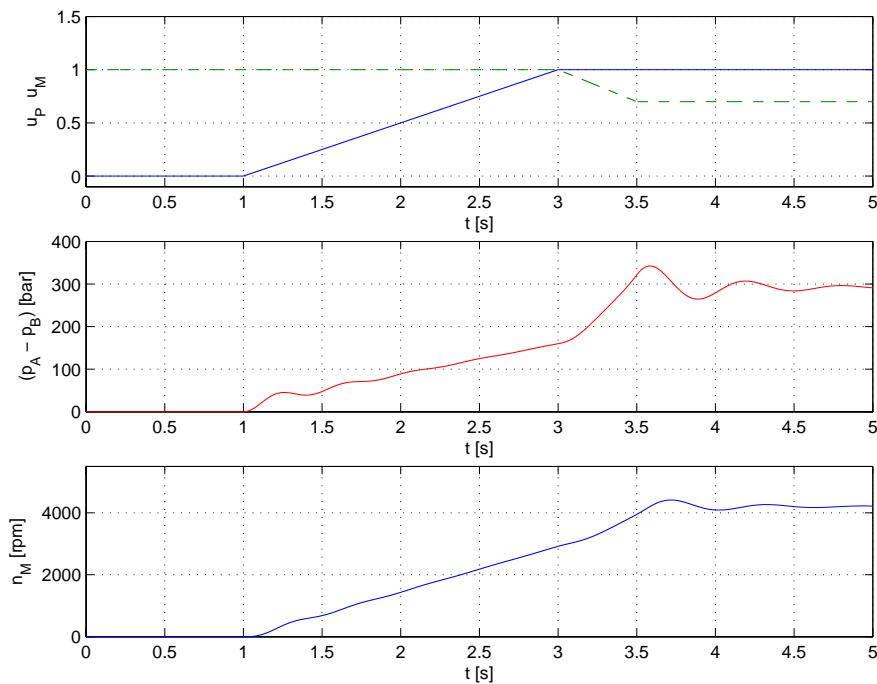


Bild 3: Simulation eines Beschleunigungsvorgangs mit den Stellgrößen $u_P(-)$, $u_M(- - -)$, dem Differenzdruck $p_A - p_B$ und der Abtriebsdrehzahl n_M

- [2] Kiencke, U. und Nielsen, L.: *Automotive Control Systems*, Springer Verlag Berlin, 2. Auflage, 2005.
- [3] Morère Y.: *Mise en oeuvre de lois de commande pour les modèles flous de type Takagi Sugeno*, PhD thesis, University Valenciennes, LAMIH-SF, France, 2001.
- [4] Murrenhoff, H. und Wallentowitz, H.: *Fluidtechnik für mobile Anwendungen. Um-druck zur Vorlesung, RWTH Aachen, Institut für fluidtechnische Antriebe und Steuerungen*, 2. Auflage, 2003.
- [5] Rugh, W. J. und Shamma, J. S. Survey Paper, Research on gain scheduling. *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.
- [6] Schulte, H.: Approximate Modeling of a Class of Nonlinear Oscillators using Takagi-Sugeno Fuzzy Systems and Its Application to Control Design. *In: Proceedings of 44th IEEE Conference on Decision and Control*, Seville, Spain. Dec. 12-15, 2005.
- [7] Schulte, H.: On Fuzzy descriptor systems and nonlinear model following control of two-link robot manipulators. *In: Proceedings of GMA Workshop Computational Intelligence*, Bommerholz, Germany. November 17./18., 2005.
- [8] Takagi, T. und Sugeno, M., Fuzzy identification of systems and its application to modelling in control, *IEEE Transactions on System, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

Methoden und Lösungen für das Energiemanagement im liberalisierten Energiemarkt

Dr.-Ing. Peter Bretschneider

Fraunhofer Anwendungszentrum Systemtechnik

Am Vogelherd 50

98692 Ilmenau

Tel. (03677) 461102

Fax (03677) 461100

E-Mail: peter.bretschneider@ast.iitb.fraunhofer.de

1 Einleitung

Der Wettbewerb auf dem liberalisierten Energiemarkt sowie die gesetzlichen Auflagen zum Klimaschutz führen bei den Versorgungsunternehmen zu einem maßgeblichen Kostendruck und erfordern unter energiewirtschaftlichen Gesichtspunkten fortwährende Maßnahmen für die rationelle Bereitstellung und Nutzung von Energie. Als Beispiele sind die Steigerung der Wirkungsgrade von Technologien und Anlagen oder auch die kostenoptimale Betriebsführung von Erzeugungsanlagen durch die Nutzung von Energiemanagementsystemen zu nennen.

In diesem Beitrag werden Anforderungen, Konzeption und Design sowie Methoden und Lösungen eines Energiemanagementsystems vorgestellt, welches für regionale und kommunale Versorgungsunternehmen konzipiert ist und die liberalisierten Marktverhältnisse berücksichtigt.

2 Aspekte zum liberalisierten Energiemarkt

Mit der Einführung des EU-Binnenmarktes verpflichteten sich die EU-Mitgliedsstaaten Ende des 20. Jahrhunderts, ihre nationalen Energiemärkte zu liberalisieren und für den Wettbewerb in den Bereichen Erzeugung und Vertrieb zu öffnen. Die Umsetzung war mit tiefgreifenden strukturellen Veränderungen verbunden und erforderte entsprechend Abbildung 1 die Entflechtung der Geschäftsbereiche Erzeugung, Übertragung, Transport, Verteilung und Vertrieb.

Neue Marktregeln in Form von Gesetzen, Richtlinien und Vereinbarungen wurden zur Öffnung des Energiemarktes erlassen. Sie führten einerseits zum Wegfall von „festen“ Versorgungsgebieten mit der freien Wahl des Stromlieferanten durch den Endkunden und andererseits zu neuen Marktplätzen und Marktteilnehmern wie z.B. Strombörse, Bilanzkreiskoordinator (BKO), Bilanzkreisverantwortliche (BKV), Lieferanten und Händler (vgl. Abbildung 1).

Der Zugang und die Nutzung der Übertragungs- und Verteilnetze für Strom bzw. der Transport- und Verteilnetze für Gas sind aufgrund der natürlichen Monopolstellung vom Wettbewerb ausgenommen und allen Marktteilnehmern zu gleichen transparenten und diskriminierungsfreien Bedingungen zu gewähren.

Die zu bewältigenden energiewirtschaftlichen Prozesse sind seitdem hinsichtlich Anzahl und Komplexität umfangreicher und zeichnen sich durch die Verzahnung von kaufmänn-

nischen und technischen Prozessen aus. Sie unterscheiden sich für die Geschäftsbereiche Netz, Handel und Vertrieb und benötigen für ihre effiziente Abwicklung geeignete IT-Systeme sowohl für die Zählung, Messung und Abrechnung von Energie als auch für das in diesem Beitrag im Vordergrund stehende Energiemanagement.

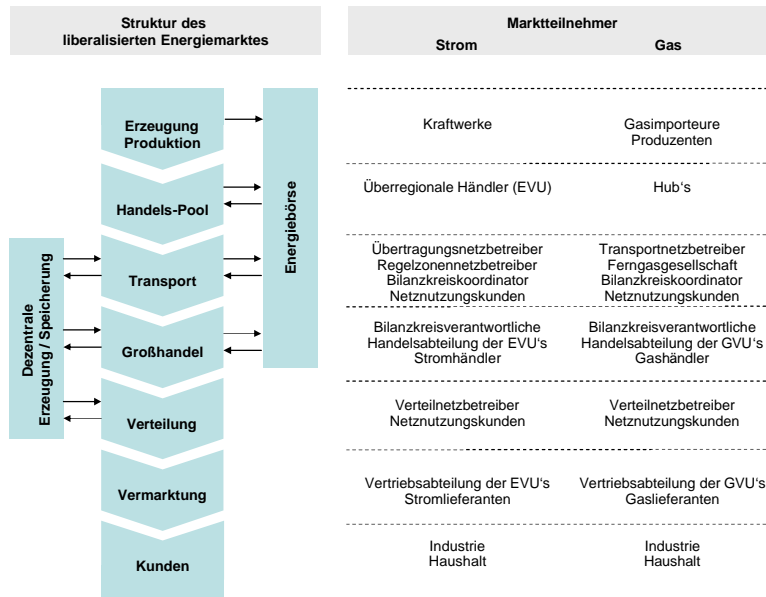


Abbildung 1: Struktur und Marktteilnehmer im liberalisierten Energiemarkt

3 Anforderungen an das Energiemanagement

Die Anforderungen an das Energiemanagement sind Gegenstand der nachfolgenden Ausführungen. Wie bereits in Abschnitt 2 angedeutet, ergeben sich im liberalisierten Energiemarkt zahlreiche neue Aufgaben, die von den einzelnen Geschäftsbereichen zu bewältigen und somit im Energiemanagementsystem umzusetzen sind.

Im Netzbereich erfolgt unter Berücksichtigung der jeweiligen Vertragslaufzeiten die Erfassung, Aufbereitung und Bereitstellung der Messdaten für die berechtigten Marktteilnehmer wie z.B. Bilanzkreisverantwortlicher und Bilanzkreisadministrator. Hierzu sind alle Kunden-Händler-Beziehungen des Netzgebietes auf Basis der Netznutzungsverträge abzubilden und die entsprechenden Bedarfslastgänge zuzuordnen. Dabei ergeben sich die Bedarfslastgänge entweder direkt aus den erfassten Leistungswerten des Zählerfernabfragesystems oder alternativ über ein definiertes Lastprofilverfahren aus den Arbeitswerten des Abrechnungssystems. Abbildung 2 zeigt beispielhaft für die Messdatenerfassung den sich ergebenden Workflow mit Angabe der Verarbeitungssysteme und Marktteilnehmer.

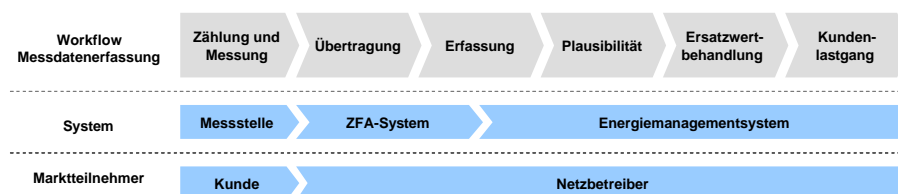


Abbildung 2: Workflow Messdatenerfassung

Eine weitere im Netzbereich benötigte Funktion ist die Bilanzierung aller Bedarfs- und Beschaffungslastgänge. Laut Marktregeln ist sie rückwirkend für den letzten Monat pro

Bilanzkreisverantwortlichen vorzunehmen und sowohl dem Bilanzkreisverantwortlichen als auch dem Bilanzkreiskoordinator zur Verfügung zu stellen. Weitere benötigte Funktionen sind die Mehrmindermengenrechnung und der Lieferantenwechselprozess, die durch das Energiemanagementsystem zu unterstützen sind.

Hauptaugenmerk des Vertriebsbereichs liegt im Endkundengeschäft und beinhaltet die Vermarktung und den Verkauf von Energie an Industrie und Haushalte. Darüber hinaus ist der Vertrieb für die möglichst genaue Vorhersage der zukünftigen Bedarfslastgänge pro Bilanzkreis verantwortlich. Hierzu sind alle zu versorgenden Kunden mit ihren Bedarfslastgängen einzeln zu erfassen und unter Beachtung ihrer Vertragslaufzeiten durch geeignete Vorhersagemethoden in die Zukunft zu prognostizieren. Die vorhergesagten Kundenlastgänge sind anschließend bilanzkreisscharf zu aggregieren und dem Handelsbereich für die Planung der Energiebeschaffung zu überlassen.

Der Handelsbereich ist für die Energiebeschaffung verantwortlich und beinhaltet die zeitgleiche Deckung von Bedarf und Beschaffung. Im Rahmen des Energiehandels werden hierzu gezielte Ein- und Verkäufe getätigt, welche revisionsicher zu erfassen und geeignet darzustellen sind. Offene Positionen sind einfach und übersichtlich anzuzeigen, um wirtschaftliche Risiken schnell erkennen zu können. Die Ein- und Verkäufe sollten nach Möglichkeit das energiewirtschaftliche Optimum widerspiegeln und auch eigene Kraftwerkskapazitäten berücksichtigen. Hierzu sind geeignete Optimierungsverfahren für die kostenoptimale Planung von multikriteriellen dynamischen Beschaffungs- und Versorgungsprozessen zu implementieren. Es ergeben sich dann optimierte Bedarfs- und Beschaffungszeitreihen, die im anschließenden Fahrplanmanagement zu verarbeiten sind. Das Fahrplanmanagement ist dabei für das ESS-Fahrplanformat inklusive der damit verbundenen Workflows zu implementieren und umfasst unter anderem die Verwaltung, Versionierung und Versendung von Fahrplänen pro Bilanzkreiskoordinator.

Insgesamt resultieren im liberalisierten Energiemarkt die in Abbildung 3 wesentlichen Aufgaben und Datenflüsse. Das Zeitreihen-, Netznutzungs- und Bilanzkreismanagement sowie Prognose und Optimierung sind im Energiemanagementsystem umzusetzen. Die Datenflüsse von und zu den externen Systemen sind zu unterstützen und transparent über definierte Workflows abzubilden.

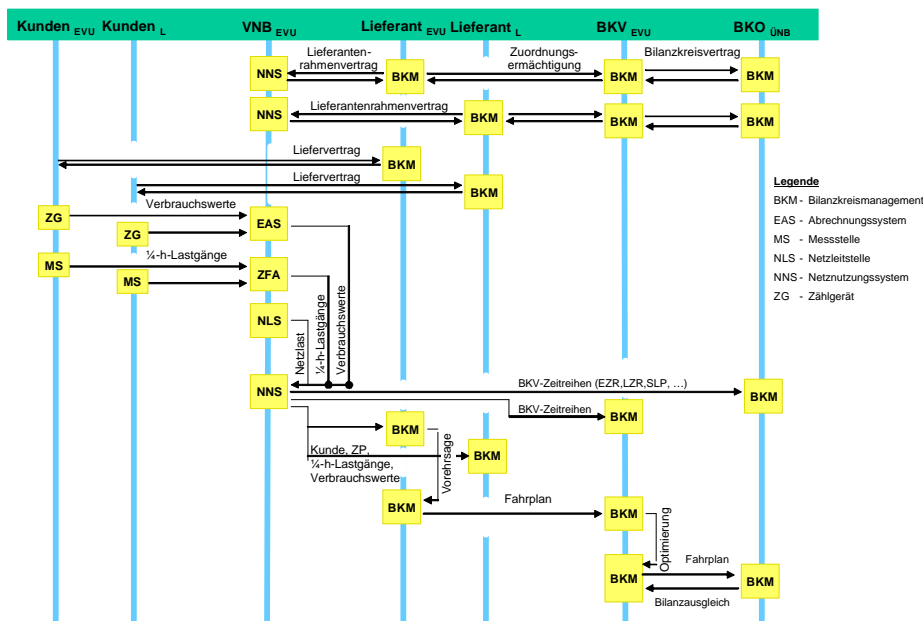


Abbildung 3: Sequenzdiagramm wesentlicher Aufgaben und Datenflüsse

Softwaretechnisch ist das Energiemanagementsystem als Client-Server-Lösung mit einer zentralen Datenhaltung zu gestalten. Die informatorische Trennung der einzelnen Geschäftsbereiche ist sicherzustellen. Grundsätzlich ist die Verwaltung und Verarbeitung von Massendaten zu unterstützen.

4 Konzeption und Entwurf

Der folgende Abschnitt stellt die Konzeption und den Entwurf des Energiemanagementsystems für regionale und kommunale Versorgungsunternehmen vor. Das Energiemanagementsystem wird für den liberalisierten Energiemarkt für die Energiemedien Strom, Gas, Wärme und Kälte konzipiert. Die strategische Ausrichtung erfolgt als Entscheidungshilfesystem zur Generierung von objektiven Entscheidungsvorschlägen. Die Leistungsfähigkeit beinhaltet neben der Lösung von mehrkriteriellen dynamischen Optimierungsaufgaben die effektive und hinreichend genaue Vorhersage des zu erwartenden Energiebedarfs.

Softwaretechnisch gestaltet sich das Energiemanagementsystem als SQL-Client-Server-Lösung und nutzt für die leistungsfähige Datenhaltung eine SQL-Oracle-Datenbank. Die Einbindung in die unternehmensweite IT-Landschaft wird durch Standardschnittstellen unterstützt. Sie dienen zum Datenaustausch mit den vor- und nachgelagerten IT-Systemen wie z.B. mit dem Abrechnungs- oder Zählerfernabfragesystem. Der grundsätzliche Aufbau des Energiemanagementsystems gestaltet sich entsprechend Abbildung 4 modular und stellt die Funktionen Zeitreihen-, Netznutzungs- und Bilanzkreismanagement sowie Prognose, Optimierung und Prozessautomation (Scheduler) bereit. Die Kombination der einzelnen Module wird entweder über die externe Marktkommunikation oder auf Basis von freigegebenen Zeitreihen unterstützt, die Träger der relevanten energiewirtschaftlichen Informationen sind. Auswahl und Verknüpfung der Module sind entsprechend den Projektanforderungen skalierbar und ermöglichen somit die Modellierung vielfältiger energiewirtschaftlicher Fragestellungen.

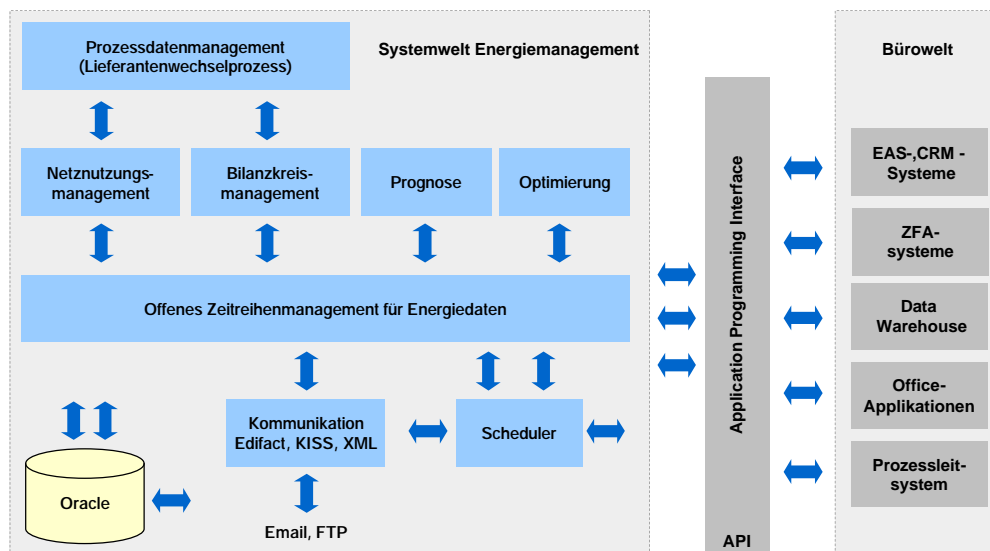


Abbildung 4: Module des Energiemanagementsystems

Das Netznutzungsmanagement unterstützt den Netzbetreiber bei der Bewältigung der ihm zugewiesenen energiewirtschaftlichen Aufgaben. Es gestattet sämtliche Marktbeziehungen zu allen relevanten Marktteilnehmern zu hinterlegen. Hierbei handelt es sich um

Entnahme- und Einspeisekunden, Lieferanten und Bilanzkreisverantwortliche. Arbeitsgemessene Kunden werden über das Normlastprofilverfahren im Rahmen des Bilanzkreismanagements erfasst. Dabei stehen sowohl das synthetische als auch analytische Verfahren zur Verfügung. Die Netznutzungskunden, die Zählpunkte und der Lieferant werden im Netznutzungsmanagement vertraglich erfasst. Über die Zählpunkte erfolgt pro Kunde und Lieferant die verursachungsgerechte Zuordnung der Kundenlastgänge. Die Vertragsbeziehungen sind dabei in der Historie nachvollziehbar. Im Messwertmanagement werden die Kundenlastgänge rückwirkend für den letzten Monat pro Lieferant und Bilanzkreisverantwortlichen aggregiert und dem Bilanzkreiskoordinator bereitgestellt.

Das Bilanzkreismanagement ist sowohl für den Bilanzkreisverantwortlichen als auch für den Lieferanten zur Erfassung und Verwaltung von Handelsgeschäften konzipiert. Die notwendigen Stammdaten wie z.B. Kunden, Verteilnetzbetreiber und Lieferanten werden im System hinterlegt. Somit sind zu jedem Zeitpunkt die Marktbeziehungen von Netzbetreibern, Kunden, Zählpunkten und Lastprofilen nachvollziehbar. Die Kommunikation mit den jeweiligen Marktteilnehmern wird über die Standardformate MSCONS, EDICT, XML und CSV realisiert. Die Verständigung mit dem Bilanzkreiskoordinator erfolgt über das Fahrplanmanagement auf Basis des ESS-Formates und den damit verbundenen Workflows.

Das Prozessdatenmanagement realisiert die externe Marktkommunikation. Es unterstützt die Standardformate EDIFACT, MSCONS und UTILMD-CSV inklusive ihrer Workflows für die automatische Datenverarbeitung. Die Abbildung des Lieferantenwechselprozess erfolgt ebenfalls im Prozessdatenmanagement und stellt eine Erweiterung des Netznutzungs- und Bilanzkreismanagements dar.

Das Prognosemodul stellt zahlreiche Funktionen zur Lastvorhersage für die Ressourcen Strom, Gas, Wärme und Kälte bereit. Die Vorhersagen unterstützen sowohl das operative Lastmanagement als auch die strategische Planung. Wesentliche Grundlage für eine optimale Planung aller verfügbaren Ressourcen ist die genaue Vorhersage der betreffenden Bedarfslastgänge für die Planungshorizonte Tag, Woche, Monat und Jahr. Hierzu werden verschiedene Vorhersagemethoden wie z.B. die Vergleichstagsprognose, die musterbasierte Vorhersage, AR-, ARX- und ARMAX-Modelle oder auch künstliche neuronale Netze in Form eines Methodenpools zur Verfügung gestellt [1], [2].

Das Optimierungsmodul unterstützt den Anwender bei der kostenoptimalen Planung aller verfügbaren Ressourcen unter Beachtung der gesetzlichen, technischen, betriebswirtschaftlichen und ökologischen Randbedingungen. Mit Blick auf den liberalisierten Energiemarkt bietet gerade die Möglichkeit der freien Energielieferantenwahl neue Möglichkeiten zur Optimierung. Neue Komponenten wie z.B. zusätzliche Verbraucher, Kraftwerkskomponenten oder Lieferverträge können erstellt und grafisch in das bestehende Modell eingefügt oder getauscht werden. Zur Modellbildung steht ein grafischer Editor zur Verfügung, der eine symbolische und analytische Beschreibung des Optimierungsproblems gestattet. Als Optimierungsverfahren wird die Gemischt Ganzzahlig Lineare Programmierung (GLP) eingesetzt [3].

Der Scheduler übernimmt die Automation der einzelnen Prozesse. Hierzu stehen verschiedene Jobs wie z.B. für den Im- und Export von Zeitreihen, für die Plausibilisierung und Ersatzwertbehandlung von Zeitreihen, für den Start von Applikationen, für die Ausführung von SQL-Befehlen sowie für den Empfang und Versand von E-Mails zur Verfügung. Die Jobs können zyklisch zu definierten Zeitpunkten oder in Abhängigkeit ei-

nes bereits ausgeführten Jobs gestartet werden. Eine manuelle Ausführung der Jobs ist ebenfalls möglich.

5 Ergebnisse und Zusammenfassung

Die im vorangegangenen Abschnitt skizzierte Lösung wurde softwaretechnisch in einem Energiemanagementsystem umgesetzt und mittlerweile in zahlreichen Projekten erfolgreich eingesetzt. Vorteilhaft erwies sich dabei der modulare Ansatz, der eine breite Palette energiewirtschaftlicher Aufgaben abdeckt. Die hohe Flexibilität und Skalierbarkeit des konzipierten Systems gestattete die Realisierung von zahlreichen kundenspezifischen Energiemanagementlösungen, beginnend mit einem Einzelsystem für die Prognose bis hin zum Gesamtsystem bestehend aus allen Modulen.

Die zukünftigen Arbeiten konzentrieren sich auf die Vorhersage der Energieerzeugung von fluktuierenden Erzeugeranlagen wie z.B. Windkraft- oder Photovoltaik [4], auf die Implementierung linearer und nichtlinearer Zeitreihenanalysemethoden und auf die Planung und Betriebsführung von unsicheren und risikobehafteten Beschaffungs- und Versorgungsprozessen.

6 Literatur

- [1] Bretschneider, P.: „Ein Beitrag zur Vorhersage musterbasierter nichtlinearer stochastischer Signale“, Dissertation A, TU Ilmenau, 2002
- [2] Bunke, D.; Goetting, B.; Kück, H.: „Automatisierte Abwicklung des Lieferantenwechsels und der Fahrplananmeldung“, ew, Frankfurt, 6/2004
- [3] Kleere, Th.; Ortjohann, E.: „Portfolio-Optimierung im liberalisierten Strommarkt“, ew, Frankfurt, 23/2003
- [4] Sielaff, D.: „Untersuchung und Entwicklung von Methoden zur Beschreibung von nichtlinearen stochastischen Zeitreihen für die Vorhersage energiewirtschaftlicher Bedarfslastgänge“, Diplomarbeit, TU Ilmenau, 2004
- [5] Niegsch, C.; Straßberger, F.: „Der europäische Strommarkt“, Studie, DG BANK, Frankfurt am Main, 11/2000

Vergleich nichtlinearer Vorhersagemodelle anhand von Referenzbeispielen

Elmar Garcia, Peter Bretschneider, Thomas Rauschenbach

Fraunhofer Anwendungszentrum Systemtechnik Ilmenau

Am Vogelherd 50, D-98693 Ilmenau

Tel.: +49 - 3677 - 461 - 105

Fax: +49 - 3677 - 461 - 100

Elmar.Garcia@ast.iitb.fraunhofer.de

Zusammenfassung

In den letzten Jahren wurden für die Modellierung und Prädiktion nichtlinearer Systemdynamiken eine Vielzahl von nichtlinearen Vorhersagemodellen entwickelt. Dieser Artikel präsentiert verschiedene aktuelle Prognoseansätze und vergleicht diese anhand von simulativen Untersuchungen.

1 Einführung

Vorhersagen bilden die Grundlage zahlreicher prädiktiver Entscheidungs-, Diagnose und Regelungsstrategien. Die hierbei betrachteten Systeme weisen in der Regel ein komplexes, nichtlineares Verhalten auf und können deswegen nur mit unzureichender Prognosegenauigkeit durch lineare Vorhersagemodelle prädiziert werden. Die Forderung nach präziser und zuverlässiger Bestimmung des zukünftigen Verhaltens motivierte die Erforschung und Entwicklung von Strategien und Methoden zur Modellierung und Vorhersage nichtlinearer Prozesse und Zeitreihen. Der vorliegende Beitrag betrachtet hierbei: Bayes-Filter, künstliche neuronale Netze, muster-, fuzzy- und neuro-fuzzy-basierte Vorhersagestrategien.

In Abschnitt 2 erfolgt zunächst die Vorstellung und Beschreibung der untersuchten Vorhersagemodelle. Anschließend werden im Abschnitt 3 die Ergebnisse der simulativen Untersuchungen präsentiert. Eine Zusammenfassung sowie ein Ausblick auf weitere Forschungsaktivitäten beschliessen den Artikel.

2 Vorhersagemodelle

2.1 Bayes-Filter

Bayessche Filter schätzen stochastisch den Zustand eines dynamischen Systems \mathbf{x}_k aus verrauschten Beobachtungen \mathbf{y}_k . Zur Beschreibung des dynamischen Verhaltens eines Prozesses kommt dabei die Zustandsraumdarstellung

$$\dot{\mathbf{x}}_t = \mathbf{f}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t) \quad (1)$$

$$\mathbf{y}_t = \mathbf{g}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{n}_t) \quad (2)$$

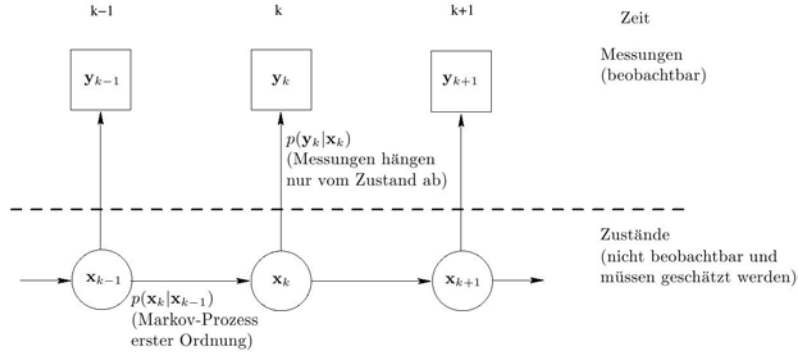


Bild 1: Zustandsraumdarstellung als Hidden-Markov-Modell erster Ordnung

zur Anwendung. Hierbei repräsentieren die Gleichungen 1 und 2 die Zustands- und Beobachtungsgleichung des Systems, \mathbf{x}_t ist der Zustandsvektor, \mathbf{y}_t ist der Beobachtungsbzw. Messwertvektor, \mathbf{u}_t ist die exogene Eingangsgröße des Systems, $\mathbf{f} : \mathbf{R}^{N_x} \mapsto \mathbf{R}^{N_x}$ und $\mathbf{g} : \mathbf{R}^{N_x} \mapsto \mathbf{R}^{N_y}$ sind zwei, eventuell zeitvariante, Vektorfunktionen und \mathbf{v}_t und \mathbf{n}_t stellen das dynamische Prozess- und Messrauschen dar. Für praktische Filterprobleme wird zur zeitdiskreten Beschreibung

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \quad (3)$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{n}_k) \quad (4)$$

übergegangen. Sie entspricht einem Hidden-Markov-Modell erster Ordnung (vgl. Abb. 1) mit

$$p(\mathbf{x}_0), p(\mathbf{x}_k | \mathbf{x}_{k-1}), k \geq 1 \quad (5)$$

$$p(\mathbf{y}_k | \mathbf{x}_k), k \geq 0 \quad (6)$$

wobei die Zustandsübergangswahrscheinlichkeit $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ durch die Zustandsgleichung und die Beobachtungswahrscheinlichkeit $p(\mathbf{y}_k | \mathbf{x}_k)$ durch die Beobachtungsgleichung beschrieben wird. Das Bayes-Konzept zur dynamischen Zustandsschätzung versucht fortlaufend die Posterior-Dichtefunktion des Zustandes aus allen verfügbaren, einschließlich fortlaufend neu eintreffender Beobachtungen zu ermitteln. Der Schätzalgorithmus arbeitet rekursiv in zwei Stufen: Vorhersage (Prädiktion) und Korrektur (Update). In der Prädiktionsstufe erfolgt, unter Verwendung der Übergangswahrscheinlichkeit¹ $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ und der letzten Prior-Dichte $p(\mathbf{x}_{k-1} | \mathbf{Y}_{k-1})$, $\mathbf{Y}_{k-1} = \mathbf{y}_1, \dots, \mathbf{y}_{k-1}$ die Vorhersage der aktuellen Zustandswahrscheinlichkeitsdichte (Prior-Dichte):

$$p(\mathbf{x}_k | \mathbf{Y}_{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Y}_{k-1}) d\mathbf{x}_{k-1}. \quad (7)$$

Da die Vorhersage im Allgemeinen verfälscht ist, wird in der nachfolgenden Korrekturstufe die aktuelle Beobachtung \mathbf{y}_k verwendet, um die prädiizierte Zustandswahrscheinlichkeitsdichte entsprechend zu korrigieren. Mit Hilfe des Bayes-Theorems wird aus der Prior-Dichte und der neuen Beobachtung die Posterior-Dichte

$$p(\mathbf{x}_k | \mathbf{Y}_k) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_{k-1})}{p(\mathbf{y}_k | \mathbf{Y}_{k-1})} \quad (8)$$

¹Die Übergangswahrscheinlichkeit $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ ist vollständig durch Prozessgleichung und die Prozessrauschwahrscheinlichkeit $p(\mathbf{v}_k)$ definiert (Gl. 3).

mit der von der Beobachtungswahrscheinlichkeit² abhängigen Normalisierungskonstante

$$p(\mathbf{y}_k | \mathbf{Y}_{k-1}) = \int p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_{k-1}) d\mathbf{x}_k \quad (9)$$

ermittelt. Dadurch wird eine fortlaufende Nachbesserung der vorherigen Dichteschätzung erreicht. Aus der Kenntnis der Posterior-Dichte $p(\mathbf{x}_k | \mathbf{Y}_k)$ lässt sich für *jedes* beliebige Optimalitätskriterium eine optimale Zustandsschätzung ableiten. So ist zum Beispiel die optimale Schätzung im Sinne des minimalen mittleren quadratischen Fehlers (MMSE) der bedingte Mittelwert von \mathbf{x}_k [1]:

$$\hat{\mathbf{x}}_{k|k}^{MMSE} = E\{\mathbf{x}_k | \mathbf{Y}_k\} = \int \mathbf{x}_k \cdot p(\mathbf{x}_k | \mathbf{Y}_k) d\mathbf{x}_k. \quad (10)$$

Gleichermaßen lassen sich aus $p(\mathbf{x}_k | \mathbf{Y}_k)$ Schätzungen für Median, Kovarianz, Kurtosis, Konfidenzintervalle und andere Größen ermitteln.

Mittels Linearisierungen der nichtlinearen Systemdynamik und weiterer Approximationen, lassen sich aus der bayesschen Schätzung eine Vielzahl rekursiver Algorithmen zur dynamischen Zustandsschätzung ableiten. Die wichtigsten und in dieser Arbeit betrachteten Filter sind:

- das Extended Kalman Filter (EKF),
- die Sigma Point Kalman Filter (SPKF)³
 - Unscented Kalman Filter (UKF)
 - Central Difference Kalman Filter (CDKF)
- die Partikel Filter (PF)
 - Sequential Importance Resampling Filter (SIR Filter)
 - Sigma Point Particle Filter (SPPF)
 - Gaussian Sum Particle Filter (GSPF)
 - Gaussian Mixture Sigma Point Particle Filter (GMSPPF)

2.2 Musterbasierte Vorhersage

In [3] und [4] stellen die Autoren ein hybrides Vorhersagesystem auf Basis eines, um eine Musterkomponente erweiterten Komponentenmodells vor:

$$\mathbf{x}_k = \mathbf{x}_k^T + \mathbf{x}_k^P + \mathbf{x}_k^M + \mathbf{x}_k^S \quad (11)$$

mit \mathbf{x}_k^T - Polynomkomponente,
 \mathbf{x}_k^P - periodische Komponente,
 \mathbf{x}_k^M - Musterkomponente,
 \mathbf{x}_k^S - stochastische Komponente.

²Die Beobachtungswahrscheinlichkeit $p(\mathbf{y}_k | \mathbf{x}_k)$ ist vollständig durch die Beobachtungsgleichung und die Messrauschwahrscheinlichkeit $p(\mathbf{n}_k)$ definiert (Gl. 4).

³auch statistische Linearisierungs-Kalman-Filter oder lineare Regressions-Kalman-Filter genannt [2]

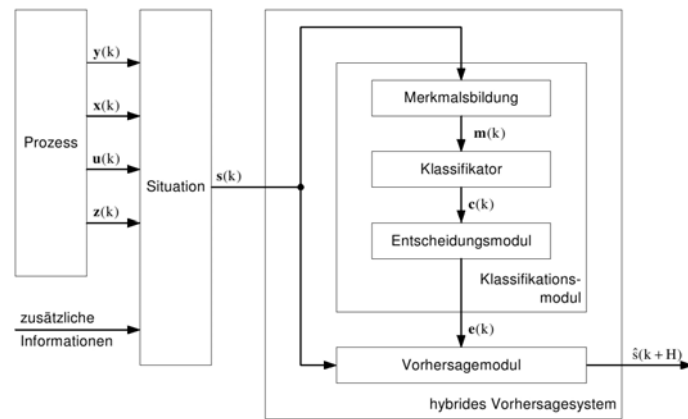


Bild 2: Struktur des hybriden Vorhersagesystems zur Vorhersage musterbasierter Signale (nach [3])

Die Musterkomponente dient dabei zur Beschreibung typischer, zu unterschiedlichen Zeiten wiederkehrender Abschnitte einer Zeitreihe und erspart so den Einsatz komplizierter deterministischer Modelle. Unter einer musterbasierten Zeitreihe wird hierbei nach [3] folgende Definition verstanden:

„Eine Zeitreihe heißt *musterbasiert*, wenn Teilbereiche von ihr durch typische, wiederkehrende Teilverläufe klassifizierbar und beschreibbar sind. Diese Teilverläufe werden als *Muster* oder auch *Musterobjekte* bezeichnet und können linearer oder nichtlinearer Gestalt sein. Ihr Auftreten erfolgt entweder periodisch oder aperiodisch.“

Gelingt die Klassifikation von Musterobjekten in einer Zeitreihe, so brauchen nur noch Änderungen um dieses Muster mittels bereits bekannter Signalmodelle beschrieben werden. Da Musterobjekte oft nichtlineare Zusammenhänge abbilden, wird so eine Signallinearisation (z. B. mittels musterbasierter Differenzenfilter) und infolgedessen ein erheblich vereinfachter Modellentwurf erreicht. Die Gesamtstruktur der musterbasierten Vorhersage wird in Abbildung 2 wiedergegeben.

2.3 Künstliche neuronale Netze

Für die Modellbildung und Vorhersage von Zeitreihen haben sich Multilayer Feedforward Netze als besonders geeignet erwiesen. Verschiedene Modelle künstlicher neuronaler Netze fallen in diese Kategorie. In der Zeitreihenvorhersage häufig verwendet werden:

- Multilayer Perceptrons (MLP),
- Finite Impule Response Networks oder
- Time Delay Neural Networks.

Alle Netzstrukturen lassen sich zur Beschreibung des dynamischen Verhaltens einer nichtlinearen Zeitreihe verwenden. Da sie ineinander überführbar sind und identische Ergebnisse liefern [5], wird in diesem Beitrag nur das MLP weiter betrachtet.

Zur Beschreibung des dynamischen Verhaltens einer nichtlinearen Zeitreihe wird ein NLAR(m)-Prozess

$$y_k = f [y_{k-1}, y_{k-2}, \dots, y_{k-m}] + \varepsilon_k \quad (12)$$

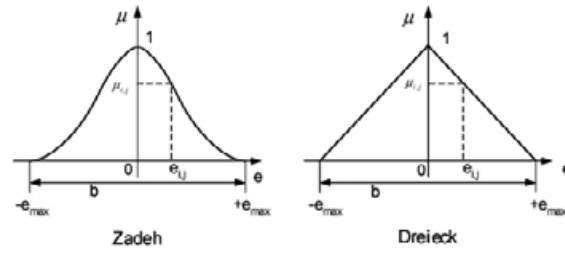


Bild 3: Fuzzy-Zugehörigkeitsfunktionen für die Beurteilung der Differenzen (nach [6])

mit der unbekannte, nichtlineare Dynamikfunktion f und dem White-Noise-Prozess ε_k verwendet. Durch Anlernen eines MLP wird die gesuchte Funktion f durch \tilde{f} approximiert und eine Einschrittvorhersage $\hat{y}_k = \tilde{f}[y_{k-1}, y_{k-2}, \dots, y_{k-m}]$ realisiert.

2.4 Fuzzy-basierte Vorhersage

Eine auf dem Fuzzy-Konzept basierende Vorhersagestrategie wird in [6] vorgestellt. Sie benötigt kein Modell des zugrunde liegenden Prozesses und basiert auf einem Vergleich der aktuellen Situation (Folge von Werten der Zeitreihe) mit allen Situationen der Vergangenheit. Auf der Grundlage dieses Vergleiches werden mit Hilfe eines fuzzy-basierten Algorithmus die der aktuellen Situation ähnlichste Situation bestimmt und die Fortentwicklung der vergangenen Situation als wahrscheinliche Weiterentwicklung der aktuellen Situation angenommen (Analogieprinzip).

Eine Situation s_k zum Zeitpunkt k entspricht einem Zeitfenster $\{y_{k-m+1}, y_{k-m+2}, \dots, y_k\}$ der Breite m aus den Werten der Zeitreihe. Die Breite m hängt dabei von der Dynamik der betrachteten Zeitreihe ab. Die aktuelle Situation s_k wird nun mit allen vorangegangenen Situationen der Zeitpunkte $j = k-1, k-2, \dots, k-n+m$ verglichen, wobei n die Anzahl der Zeitreihenwerte wiedergibt. Die einzelnen Vergleiche liefern einen Differenzenvektor $e_j = s_k - s_j$. Für jedes Element i von e_j werden die Zugehörigkeitswerte $\mu_{i,j}$ mit Hilfe einer Fuzzy-Zugehörigkeitsfunktion entsprechend Abbildung 3 berechnet und zu einem Distanzmaß D_j aggregiert, welches die Übereinstimmungsgüte der aktuellen Situation s_k mit der Situation s_j beschreibt:

$$D_j = \prod_{i=1}^m \mu_{i,j}. \quad (13)$$

Die Breite der Zugehörigkeitsfunktion ($b = 2 \cdot e_{max}$) bestimmt sich entsprechend den Anforderungen an die Güte der Vorhersage. Je genauer die Prognose erfolgen soll, desto kleiner muss e_{max} gewählt werden. In Abhängigkeit der Übereinstimmung zweier Situationen nimmt das Distanzmaß $D(j)$ einen Wert zwischen *Null* (keine Übereinstimmung) und *Eins* (volle Übereinstimmung) an.

Über einen Inferenzmechanismus der Form *WENN ... DANN ...* werden die ermittelten Distanzwerte zur Prognose der zukünftigen Zeitreihenwerte $y_{k+1}, y_{k+2}, \dots, y_{k+h}$ bis zum Vorhersagehorizont h verwendet:

WENN die aktuelle Situation s_k einer in der Vergangenheit aufgetretenen Situation s_j ähnlich ist,

DANN werden die zu prognostizierenden Werte $y_{k+1}, y_{k+2}, \dots, y_{k+h}$ ähnlich den Werten $y_{j+1}, y_{j+2}, \dots, y_{j+h}$ gewählt.

Die Prognose berücksichtigt nur Situationen, deren Distanzmaß ungleich *Null* ist. Das bedeutet, nur auf solche Situationen werden die *WENN ... DANN ...* Inferenzregeln angewendet. Die Berechnung des quantitativen Vorhersagewertes \hat{y}_{k+h} erfolgt für stationäre Zeitreihen über die folgende Vorschrift zur Defuzzifizierung:

$$\hat{y}_{k+1} = \sum_{j=m}^{n-1} D_j \cdot y_{j+1} \cdot \left(\sum_{j=m}^{n-1} D_j \right)^{-1} . \quad (14)$$

Dadurch gehen ähnlichere Situationen der Vergangenheit stärker in den Prognosewert ein als weniger ähnliche. Bei instationärem Prozessverhalten erfolgt die Vorhersage auf der Grundlage der Ähnlichkeit von Differenzen benachbarter Signalwerte. Die Defuzzifizierungsvorschrift aus Gleichung 14 geht dann über in:

$$\hat{y}_{k+1} = y_k + \sum_{j=m}^{n-1} D_j \cdot (y_{j+1} - y_j) \cdot \left(\sum_{j=m}^{n-1} D_j \right)^{-1} . \quad (15)$$

2.5 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Einer der ersten Ansätze zur Kombination neuronaler und linguistischer Methoden zur Funktionsapproximation stellt das in [7, 8] beschriebene ANFIS-Modell dar. Hierbei werden Takagi-Sugeno Fuzzy-Regeln

$$\begin{aligned} R_r : \text{WENN } x_1 = A_{1r}^{(1)} \wedge \dots \wedge x_n = A_{nr}^{(n)} \\ \text{DANN } y = a_0^{(r)} + a_1^{(r)}x_1 + \dots + a_n^{(r)}x_n \end{aligned}$$

in Form einer speziellen vorwärtsgekoppelten Netzwerkarchitektur mit 5 Schichten realisiert (vgl. Abb. 4).

Die erste Schicht repräsentieren die Fuzzy-Zugehörigkeitsfunktionen. Die Knoten der zweiten und dritten Schicht bilden den Prämissenteil einer Fuzzy-Regel ab. Die \prod -Knoten dienen dabei der UND-Verknüpfung auf Basis des algebraischen Produktes und die N -Knoten der Normalisierung der einzelnen Regelausgänge. Der Konklusionsteil jeder Takagi-Sugeno Regel wird in der vierten Schicht des Netzes berechnet. Die abschließende Defuzzifizierung erfolgt in der fünften Schicht (Ausgangsschicht). Hier wird aus den gewichteten Ausgängen der vorhergehenden Schicht der globale Ausgangswert ermittelt.

Zum Training von ANFIS kommt ein hybrider, zweistufiger Lernalgorithmus zum Einsatz. Ein Backpropagation-Algorithmus wird für die Adaption der initialen vorgegebenen Zugehörigkeitsfunktionen verwendet. Die Koeffizienten der linearen Ausgangsfunktionen werden mit Hilfe der Kleinstquadratalgorithmus bestimmt. Die Anwendung dieses hybriden Trainingsprozesses führt zu einer iterativen Parameteroptimierung, wodurch eine datenbasierte Approximation der Systemdynamik gelingt.

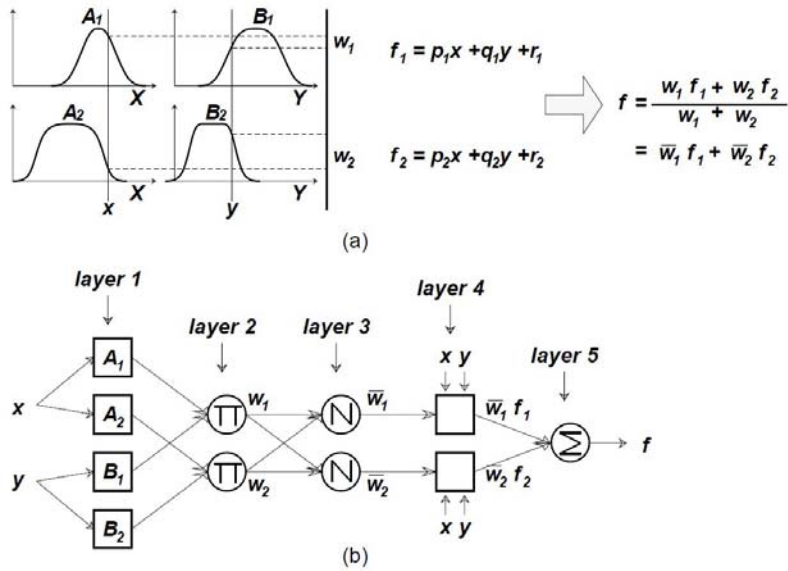


Bild 4: (a) Takagi-Sugeno-FIS erster Ordnung mit zwei Eingangsvariablen, (b) äquivalente ANFIS Implementierung (nach [7])

2.6 Neuro-Fuzzy-Kolmogorov Netze

Die Grundlage der Neuro-Fuzzy-Kolmogorov Netze (NFKN) bildet Kolmogorov's Superpositionstheorem, welches besagt, dass jede kontinuierliche Funktion mit d Variablen exakt durch die Superposition und Addition kontinuierlicher Funktionen mit einer Variablen darstellbar ist:

$$f(x_1, \dots, x_d) = \sum_{l=1}^{2d+1} g_l \left[\sum_{i=1}^d \psi_{l,i}(x_i) \right], \quad (16)$$

wobei $g_l(\cdot)$ und $\psi_{l,i}(\cdot)$ kontinuierliche, univariate Funktionen sind und $\psi_{l,i}(\cdot)$ unabhängig von f .

Unter Verwendung neuronaler Netze in Kombination mit Fuzzy-Methoden entwickeln die Autoren in [9, 10, 11] ein NFKN zur Zeitreihenvorhersage (vgl. Abb. 5). Es besteht aus zwei Schichten von Neuro-Fuzzy-Neuronen (NFN) mit linearen Hidden- und Ausgabeparametern und wird durch folgende Gleichungen beschrieben:

$$\hat{f}(x_1, \dots, x_d) = \sum_{l=1}^n f_l^{[2]}(o^{[1,l]}), \quad (17)$$

$$o^{[1,l]} = \sum_{i=1}^d f_i^{[1,l]}(x_i), \quad l = 1, \dots, n,$$

wobei n der Anzahl der Hidden-Neuronen entspricht, $f_l^{[2]}(o^{[1,l]})$ der l -ten nichtlinearen Synapse in der Ausgabeschicht, $o^{[1,l]}$ der Ausgabe des l -ten NFN in der Hidden-Schicht und $f_i^{[1,l]}(x_i)$ der i -ten nichtlinearen Synapse des l -ten NFN in der Hidden-Schicht. Die Synapsen der Hidden- und Ausgabeschicht (siehe Abbildung 6 und 7) werden durch die

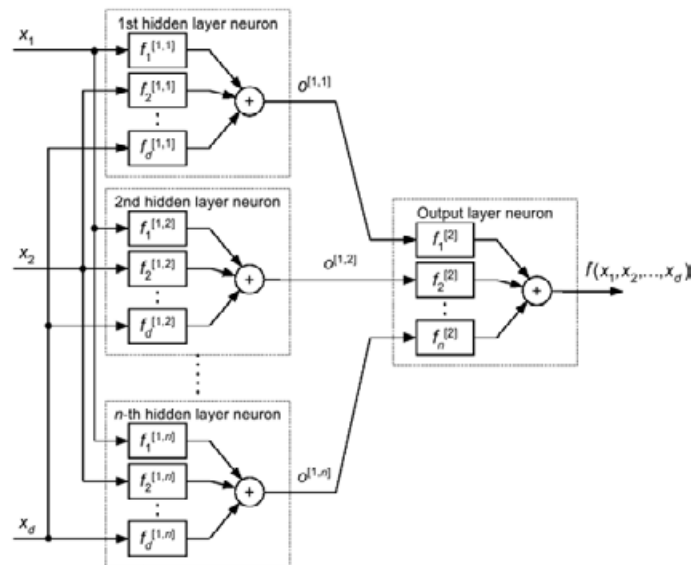


Bild 5: NFKN mit d Eingangs- und n Hidden-Layer-Neuronen (nach [11])

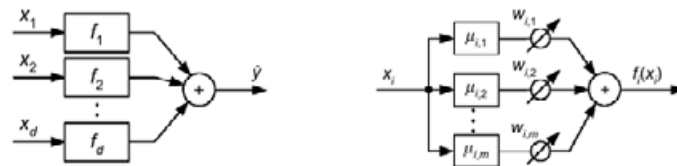


Bild 6: Neuro-Fuzzy-Neuron (links) und dazugehörige nichtlineare Synapse (rechts) (nach [11])

Gleichungen

$$f_i^{[1,l]}(x_i) = \sum_{h=1}^{m_1} \mu_{i,h}^{[1]}(x_i) w_{i,h}^{[1,l]},$$

$$f_l^{[2]}(o^{[1,l]}) = \sum_{j=1}^{m_2} \mu_{l,j}^{[2]}(o^{[1,l]}) w_{l,j}^{[2]},$$

$$l = 1, \dots, n, \quad i = 1, \dots, d$$
(18)

beschrieben, wobei m_1 und m_2 die Anzahl der Zugehörigkeitsfunktionen (ZGFs) pro Eingang eines Hidden- bzw. Ausgabeneurons ist, $\mu_{i,h}^{[1]}(x_i)$ und $\mu_{l,j}^{[2]}(o^{[1,l]})$ die ZGFs und $w_{i,h}^{[1,l]}$ und $w_{l,j}^{[2]}$ die einstellbaren Gewichte.

Als Lernverfahren finden in der Hidden-Schicht ein gradientenbasierter und in der Ausgabeschicht ein Kleinstquadratalgorithmus Anwendung. Der zweistufige Lernprozess bestimmt während der Vorwärtspropagierung die Gewichte der Ausgabeschicht. Die Einstellung der Gewichte der Hidden-Neuronen erfolgt in der Rückwärtspropagierung.

3 Ergebnisse

Ziel der durchgeführten Untersuchungen war es, die Einschnittvorhersagegenauigkeit der Vorhersagemodelle aus Abschnitt 2 zu eruieren. Als Referenzbeispiele wurden die Mackey-Glass-Zeitreihe und eine energiewirtschaftliche Bedarfszeitreihe erwählt (Abb. 8 und 9).

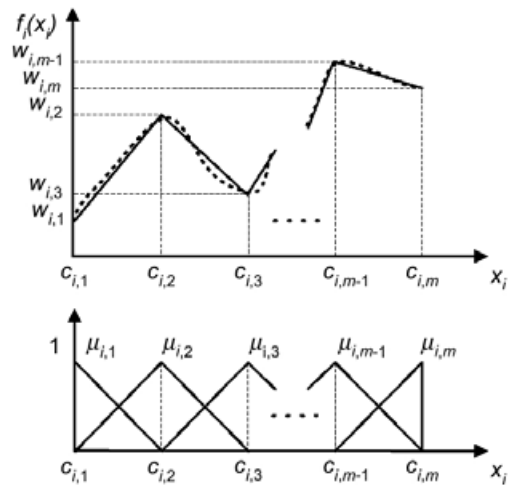


Bild 7: Approximation einer univariaten, nichtlinearen Funktion durch eine nichtlineare Synapse (nach [11])

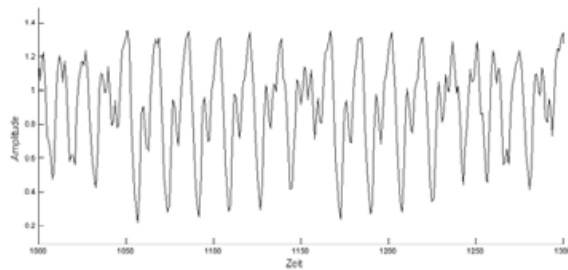


Bild 8: Mackey-Glass-Zeitreihe (Ausschnitt für $k = 1000, \dots, 1300$)

Die Mackey-Glass-Zeitreihe wurde durch die Integration der gleichnamigen nichtlinearen Differentialgleichung [12]

$$\frac{dy(t)}{dt} = \frac{ay(t - t_d)}{1 + y^c(t - t_d)} - by(t). \quad (19)$$

mit den Parametern $a = 0.2$, $b = 0.1$, $c = 10$ und $t_d = 30$ gewonnen. Die numerische Lösung wurde mittels des Runge-Kutta-Verfahren, einer Integrations-schrittweite von 0.01 und der Anfangsbedingung $y(0) = 0.9$ bestimmt.

Die zweite in den Untersuchungen verwendete Zeitreihe repräsentiert den elektrischen Lastgang eines Flächenversorgers, der zu etwa gleichen Teilen Industrie, Landwirtschaft und Haushalte beliefert. Das hierbei betrachtete Versorgungsgebiet ist durch zahlreich einspeisende Windkraftanlagen charakterisiert. Die Messung der elektrischen Last erfolgte an der Übergabestation im Viertelstundentakt. Die 15-Minutenwerte ergeben sich dabei als Mittelwert der einlaufenden Minutenwerte.

Vor der Modellvalidierung erfolgten in Voruntersuchungen die Ermittlung der optimalen Zeitverzögerung und Einbettungsdimension⁴, Tests auf Nichtlinearität sowie die Bestimmung der theoretisch möglichen linearen und nichtlinearen Vorhersagbarkeit. Diese Voruntersuchungen dienen der Zeitreihenanalyse und Parameteridentifikation für die nachfolgenden Experimente.

⁴Diese Größen werden für die Rekonstruktion der Systemdynamik aus den Messwerten einer Zeitreihe entsprechend der Einbettungstheoreme von Takens [13] und Sauer et al. [14] benötigt.

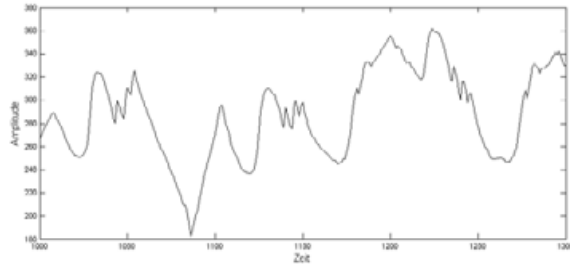


Bild 9: Energiewirtschaftliche Bedarfszeitreihe (Ausschnitt für $k = 1000 \dots 1300$)

	MG		Last	
	nRMSE	U_T	nRMSE	U_T
EKF	0.076	0.123	0.257	1.640
UKF	0.074	0.120	0.243	1.550
CDKF	0.074	0.120	0.184	1.172

Tabelle 1: Prognosefehler der Kalman-Filter

Es wird das Einschrittvorhersageverhalten der verschiedenen nichtlinearen Vorhersagemodelle untersucht. Die aus 400 Messwerten bestehenden Zeitreihen werden hierfür zu gleichen Teilen in einen Trainings- und einen Validierungsdatensatz aufgeteilt. Die Adäquatheit der Prognosemodelle wird anschließend mittels des $nRMSE$

$$nRMSE = \sqrt{(1/N) \sum (y_k - \hat{y}_k)^2 / y_k} \quad (20)$$

und des Ungleichheitskoeffizienten nach Theil U_T

$$U_T = RMSE_{Modell} / RMSE_{naiv} \quad (21)$$

quantifiziert. Der Theil'sche Ungleichheitskoeffizient wird verwendet, um die Prognosequalität des Vorhersagemodells im Vergleich zur „naiven Prognose“ zum Ausdruck zu bringen⁵.

Die Experimente werden mehrmals durchgeführt und jeweils der mittlere Prognosefehler angegeben. Bei den Bayes-Filtern wird als Prognosewert die Ausgangsvorhersage, d. h. die Zustandsschätzung vor dem Korrekturschritt verwendet. Für die fuzzybasierten Prognosestrategien werden Zugehörigkeitsfunktionen nach Zadeh benutzt. Bei allen Lernverfahren wird die Epochenanzahl auf 25 festgelegt. Die neuronalen Netze werden mit dem skalierten konjugierten Gradientenverfahren angelernet. Bei der musterbasierten Vorhersage wird die erste Hälfte der Zeitreihe für das Anlernen des hybriden Vorhersagesystems, d. h. für die Bildung der Klassenkonfiguration, verwendet und die zweite Hälfte der Zeitreihe für die Prognose. Der verwendete Situationsvektor wird dabei alleinig aus den Zeitreihenwerten gebildet. Die Ergebnisse der Untersuchungen sind nachfolgend tabellarisch abgebildet.

⁵Unter der naiven Prognose wird die Verwendung des letzten Beobachtungswertes verstanden.

	MG		Last	
	nRMSE	U_T	nRMSE	U_T
SIR-PF	0.542	0.861	0.403	2.573
SPPF	0.392	0.622	0.353	2.254
GSPF	0.356	0.565	0.298	1.903
GMSPPF	0.066	0.105	0.050	0.321

Tabelle 2: Prognosefehler der Partikel-Filter. Bei dem SPPF und dem GMSPPF wurde als Sigma Point Particle Filter das CDKF verwendet.

	MG		Last	
	nRMSE	U_T	nRMSE	U_T
Muster	0.412	0.374	0.383	0.107
KNN	0.167	0.145	0.202	0.022
Fuzzy	0.220	0.151	0.112	0.012
ANFIS	0.087	0.095	0.197	0.023
NFKN	0.130	0.126	0.163	0.018

Tabelle 3: Prognosefehler der verschiedenen nichtlinearen Vorhersagemodelle

4 Zusammenfassung und Ausblick

Der vorliegende Beitrag gab einen Überblick über existierende nichtlineare Vorhersagemodelle und verglich in einheitlichen Simulationen deren Einschrittvorhersageleistung anhand der künstlich generierten Mackey-Glass-Zeitreihe und einer der Energiewirtschaft entstammenden Bedarfszeitreihe.

Im Rahmen der Untersuchungen erzielten die Partikel-Filter, die neuronalen und Neuro-Fuzzy Netze die besten Vorhersageergebnisse. Die Genauigkeit der Funktionsapproximation weist jedoch eine starke Abhängigkeit zur Systemdynamik auf. So verschlechtern sich bei der instationären und verrauschten Lastgangzeitreihe die Prognoseergebnisse der meisten Vorhersagemodelle. Abhilfe können hier im Vorfeld angewendete Verfahren zur Rauschreduktion oder Zeitreihentransformationen leisten.

Bei dem Vergleich der Vorhersagemodelle blieben Bewertungsfaktoren wie benötigte Datenverfügbarkeit, Speicherverbrauch, Rechenzeit oder Modellierungsaufwand unberücksichtigt. Hier ergibt sich für die untersuchten Algorithmen ein differenzierteres Bild, welches bei der Wahl einer Vorhersagestrategie für eine konkrete Anwendung zu berücksichtigen ist. So sind bspw. die Partikel-Filter mit einem hohen Berechnungsaufwand verbunden, der direkt mit der erzielbaren Prognosegüte korreliert und für das Anlernen der neuronalen Netze ist eine ausreichende Datenvergangenheit zu gewährleisten.

Gegenstand zukünftiger Forschungen muss eine Ausweitung der Untersuchungen auf Mehrschrittprognosen von multivariaten, nichtlinearen Zeitreihen sein. Sie treten in vielen technischen und nichttechnischen Systemen auf und besitzen eine große praktische Relevanz. Weiterer Forschungsbedarf besteht bei der Entwicklung und Analyse neuer nichtlinearer Vorhersagemethoden wie bspw. Neuro-Fuzzy-Netzen [15]. Weitere Arbei-

ten hierzu befinden sich in der Vorbereitung.

Literatur

- [1] Wan, E. A.; Merwe, R. v. d.: *Kalman filtering and neural networks*, Kap. The Unscented Kalman Filter, S. 221–280. New York: John Wiley & Sons, Inc. 2001.
- [2] Ristic, B.; Arulampalam, S.; Gordon, N.: *Beyond the Kalman filter : particle filters for tracking applications*. Boston, London: Artech House, 1th ed. Aufl. 2004.
- [3] Bretschneider, P.: *Ein Beitrag zur Vorhersage nichtlinearer stochastischer Signale*. Dissertation, Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, Ilmenau. 2001.
- [4] Bretschneider, P.; Rauschenbach, T.: Ein Beitrag zur Vorhersage nichtlinearer stochastischer Signale. *at - Automatisierungstechnik* 50 (2002) 7, S. 334–345.
- [5] Wan, E. A.: *Time series prediction: Forecasting the future and understanding the past*, Kap. Time series prediction by using a connectionist network with internal delay lines, S. 195–217. Perseus Books Publishing. 1994.
- [6] Otto, P.: Fuzzy-basierte Zeitreihenvorhersage. *at - Automatisierungstechnik* 48 (2000) 7, S. 327–334.
- [7] Jang, J.-S. R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics* 23 (1993), S. 665–685.
- [8] Jang, J.-S. R.; Sun, C.-T.: Neuro-Fuzzy Modeling and Control. *Proceedings of the IEEE* 83 (1995), S. 378–406.
- [9] Kolodyazhniy, V.; Bodyanskiy, Y.: Fuzzy Kolmogorov's Network. *Lecture Notes in Computer Science* 3214 (2004), S. 764–771. Mircea Gh. Negoita, Robert J. Howlett, Lakhmi C. Jain (Eds.): Knowledge-Based Intelligent Information and Engineering Systems: 8th International Conference, KES 2004, Wellington, New Zealand, September 20-25, 2004, Proceedings, Part II.
- [10] Bodyanskiy, Y.; Gorshkov, Y.; Kolodyazhniy, V.; Poyedyntseva, V.: Neuro-fuzzy Kolmogorov's Network. *Lecture Notes in Computer Science* 3697 (2005), S. 1–6. Włodzisław Duch, Janusz Kacprzyk, Erkki Oja, Sławomir Zadrozny (Eds.): Artificial Neural Networks: Biological Inspirations - ICANN 2005: 15th International Conference, Warsaw, Poland, September 11-15, 2005. Proceedings, Part II.
- [11] Bodyanskiy, Y.; Kolodyazhniy, V.; Otto, P.: Neuro-Fuzzy Kolmogorov's Network for Time Series Prediction and Pattern Classification. *Lecture Notes in Computer Science* 3698 (2005), S. 191–202. Ulrich Furbach (Eds.): KI 2005: Advances in Artificial Intelligence: 28th Annual German Conference on AI, KI 2005, Koblenz, Germany, September 11-14, 2005. Proceedings.
- [12] Mackey, M. C.; Glass, L.: Oscillation and chaos in physiological control systems. *Science* (1977), S. 287–289.

- [13] Takens, F.: Detecting strange attractors in turbulence. *Lecture Notes in Mathematics* 898 (1981), S. 366–381.
- [14] Sauer, T.; Yorke, J. A.; Casdagli, M.: Embedology. *Journal of Statistical Physics* 65 (1991) 3–4, S. 579 – 616.
- [15] Abraham, A.: *Fuzzy System Engineering: Theory and Practice*, Kap. Adaptation of Fuzzy Inference System Using Neural Learning, S. 53–83. *Studies in Fuzziness and Soft Computing*. Berlin: Springer Verlag. 2005.

Methoden der Sensordiagnose zur Bewertung von Verfügbarkeit, Qualität und Sicherheit

Andreas Gommlich, Thomas Alber

Endress+Hauser Conducta GmbH+Co. KG

Dieselstraße 24, 70839 Gerlingen

Tel.: (07156) 209-{283,186}

Fax: (07156) 209-152

E-Mail: {andreas.gommlich, thomas.alber}@conducta.endress.com

1 Einleitung

Die Prozessmesstechnik in sicherheitskritischen und/oder kostenintensiven Anwendungen, wie sie zum Beispiel in den Bereichen Chemie, Pharma und Food vorkommt, muss sehr hohen Ansprüchen gerecht werden. Besonders stehen dabei die Verfügbarkeit, Qualität und Sicherheit der eingesetzten Systeme und Technologien im Vordergrund. In [1] wird deshalb die Bewertung der Sensorzustände Ausfall F, Funktionskontrolle C, außerhalb der Spezifikation S und Wartungsbedarf M empfohlen.

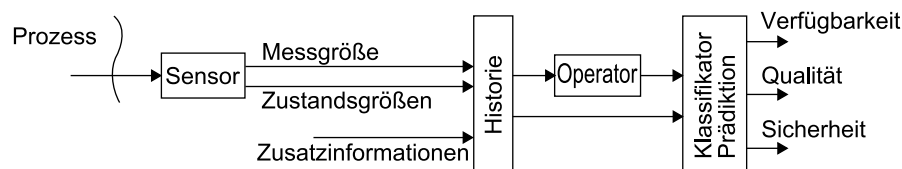


Bild 1: Sensor mit erweiterter Diagnose

In Bild 1 ist der schematische Aufbau der erweiterten Diagnose eines Sensors dargestellt. Neben der primären Messgröße werden außerdem Zustandsgrößen des Sensors wie z.B. Kennlinienparameter und externe, prozessbeschreibende Informationen für die Sensordiagnose bereitgestellt. Über einen Operator werden weitere abstrakte Größen wie die Varianz, Ableitungen oder Verknüpfungen mehrerer Größen berechnet. Alle Größen zusammengenommen bilden den vollständigen Merkmalsvektor $\mathbf{x}_i = (x_1, x_2, \dots, x_m)$ eines Sensors. Im Klassifikator erfolgt dann die Zuordnung von \mathbf{x}_i zu einer entsprechenden Klasse j in einer Diagnosekategorie i : $\mathcal{C}_{i,j}$.

Als weitere Funktion der erweiterten Diagnose setzt die Prädiktion zukünftiger Sensorzustände auf der Klassifikation auf. Basis dieser Funktion ist die zuverlässige Klassifikation und das analysierte Verhalten in der Vergangenheit. Mittels einer Extrapolation können somit die Zeitpunkte bestimmter Klassenzugehörigkeiten wie z.B. kritische und fehlerhafte Zustände vorherbestimmt werden.

Moderne Sensoren beinhalten bereits Diagnosefunktionen die nach dem Schema in Bild 1 aufgebaut sind und Sekundärgrößen des Sensors auswerten. Dabei handelt es sich meist um ad-hoc signalbasierte Verfahren bei denen Einzelmerkmale unmittelbar bzgl. eines gültigen Wertebereichs ausgewertet werden, z.B. über eine Schwellwertbetrachtung (siehe [2]).

Die Bewertung der Sensoren hinsichtlich der oben genannten Sensorzustände F, C, S und

M macht jedoch die Anwendung erweiterter signalbasierter Verfahren notwendig. Ein erster Ansatz zur Anwendung dieser Verfahren zur erweiterten Sensordiagnose soll hier vorgestellt und hinsichtlich seiner Möglichkeiten und Schwierigkeiten diskutiert werden.

In Abschnitt 2 wird der Ansatz diskutiert, wie die multivariate Analyse für die Zustandsüberwachung von Sensoren eingesetzt werden kann. Mittels der Hauptkomponentenanalyse erfolgt eine Reduktion der Daten auf die relevanten Informationen. Die Bewertung anhand der reduzierten Daten macht eine schnelle und übersichtliche Zuordnung/Klassifikation der Sensoren zu bestimmten Klassen möglich. Abschnitt 3 widmet sich der Möglichkeit interessierende Sensorzustände über einfache mathematische Modelle zu präzisieren. Dazu ist die Betrachtung der Vergangenheit der Sensormerkmale notwendig worauf die Extrapolation auf zukünftige Zeitpunkte aufsetzt. In Abschnitt 4 wird abschließend eine Zusammenfassung und ein Ausblick auf folgende Entwicklungsarbeiten geliefert.

2 Zustandsüberwachung von Sensoren

Grundsätzlich können die signalbasierten Methoden, mit denen ein Sensor gemäß seinem Zustand bewertet werden kann, nach Bild 2 in univariate und multivariate Methoden gegliedert werden.

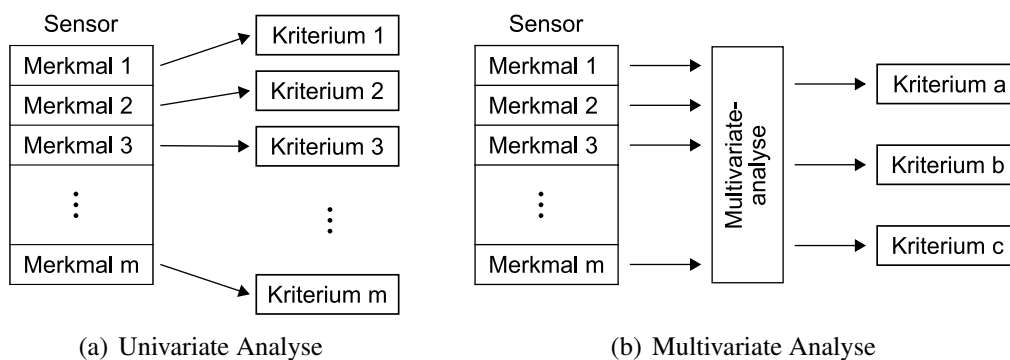


Bild 2: Signalbasierte Analysemethoden

Während bei der univariaten Analyse (UVA) jedes Sensormerkmal $x_{i,1}, \dots, x_{i,m}$ separat bewertet wird (Bild 2(a)), erfolgt bei der multivariaten Analyse (MVA) die Bewertung des gesamten Merkmalsvektors x_i hinsichtlich abstrakter, wie z.B. den oben genannten Kategorien Verfügbarkeit, Qualität und Sicherheit (Bild 2(b)).

2.1 Vorverarbeitung

Der Rohwert eines Messwertes wird im Allgemeinen stets einer Reihe vorverarbeitender Schritte wie Normierung, Filterung, Korrektur oder Transformation unterzogen (siehe [3]). Durch diese Datenvorverarbeitung werden zusätzliche Informationen, die über den reinen Rohwert hinaus gehen, bereitgestellt.

Außerdem enthalten neben den Primärgrößen sog. sekundäre Größen wesentliche Informationen, die zur Diagnose genutzt werden können. Sensorbeschreibende Zustandsgrößen können beispielsweise nach der Anordnung aus Bild 3 abgeleitet werden.

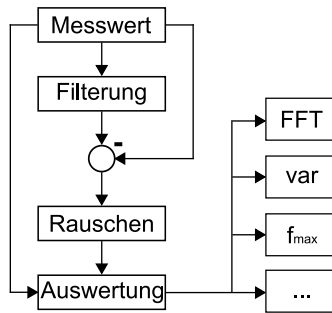


Bild 3: Berechnung sekundärer Messgrößen

Auf diese Weise stehen Größen wie z.B. Rauschfrequenzen, Zeitkonstanten bei Messwertänderungen und andere statistische Größen wie die Varianz, der Minimal- oder der Maximalwert sowie Kenngrößen die aus Transformationen abgeleitet werden, der Sensordiagnose zur Verfügung. Alle Informationen eines Sensors zusammen bilden den vollständigen Merkmalsvektor \mathbf{x}_i .

2.2 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (PCA¹) zählt zu den multivariaten Analysemethoden. Sie ermöglicht die gleichzeitige Betrachtung mehrerer Sensormerkmale unter der Berücksichtigung latenter Zusammenhänge. Redundante Anteile der Daten, die keine oder wenig Information beinhalten wie Rauschen und stark korrelierende Größen, werden bei dieser Methode nicht beachtet, wodurch die reine Datenmenge u.U. erheblich reduziert wird. Daraus ergibt sich ein reduzierter Merkmalsvektor $\tilde{\mathbf{x}}_i$.

Aus der PCA der vollständigen Merkmalsvektoren $\mathbf{x}_i \in \mathbb{R}^m$ resultiert der Unterraum \mathbb{R}^n mit $m \geq n$, der von den Hauptkomponenten (PC²) aufgespannt wird. Die wesentliche Information der Datenmatrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_k)^T$, die sich aus mehreren Merkmalsvektoren zusammensetzt, wird nach (1) durch die beiden Matrizen \mathbf{T} (Faktorwerte, Scores) und \mathbf{L} (Faktorladungen, Loadings) beschrieben. Die Loadings bilden die Orthonormalbasis des Unterraums \mathbb{R}^n . In den Scores ist der Anteil jeder PC auf den jeweiligen Datenpunkt beschrieben. Die verbleibende Information, die nicht durch die berechneten PC beschrieben wird, ist in \mathbf{E} enthalten.

$$\mathbf{X} = \mathbf{T} * \mathbf{L}' + \mathbf{E} \quad (1)$$

Für das PCA-Modell ist ein Datensatz mit repräsentativen Merkmalsvektoren \mathbf{x}_i notwendig. Merkmalsvektoren \mathbf{x}_i , die nicht diesem Datensatz zugehören, können nach (2) auf die berechneten Faktoren projiziert werden (siehe [4]).

$$\hat{\mathbf{x}} = \mathbf{L}(\mathbf{L}'\mathbf{L})^{-1}\mathbf{L}'\mathbf{x} \quad (2)$$

In Bild 4 ist ein theoretisches Beispiel der PCA nach dem NIPALS³-Algorithmus veranschaulicht.

¹Principle Component Analysis

²Principle Component

³Nonlinear Iterative Partial Least Squares, siehe [5]

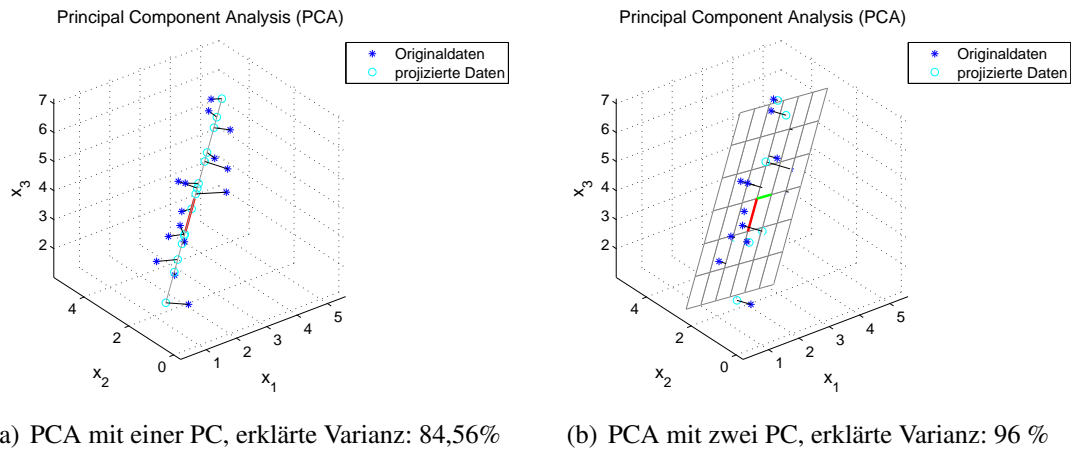


Bild 4: theoretisches Beispiel einer Hauptkomponentenanalyse

In Bild 4(a) wird eine Punktwolke im \mathbb{R}^3 durch eine PC angenähert. Damit werden bereits 85% der enthaltenen Varianz (Information) beschrieben. Mit der Berechnung einer weiteren PC steigt der erklärte Informationsgehalt auf 96% (Bild 4(b)).

In den Bildern 5 und 6 ist ein praktisches Beispiel der PCA dargestellt. Hierfür wurde das Zeitsignal mehrerer Sensoren aus der Klasse $\mathcal{C}_{\text{defekt}}$ und aus der Klasse $\mathcal{C}_{\text{intakt}}$ einer Fouriertransformation unterzogen. Die resultierten Frequenzspektren setzen sich aus 2500 Einzelwerten zusammen. Um zu erkennen, welche Frequenzen den Einfluss der durchgeführten Messreihe wiedergeben, wurde die PCA angewandt.

Das Bild 5 stellt das Resultat der PCA mit allen Frequenzanteilen des Spektrums dar. Die Projektion der Merkmalsvektoren aus dem \mathbb{R}^{2500} in den \mathbb{R}^2 ist in Bild 5(a) zu sehen. Die Trennung der beiden Klassen $\mathcal{C}_{\text{intakt}}$ (*) und $\mathcal{C}_{\text{defekt}}$ (+) ist in diesem Raum klar erkennbar. Der Loadings-Plot (Bild 5(b)) zeigt, dass die wesentliche Information in den wenigen, mit ”+” markierten Frequenzen enthalten ist.

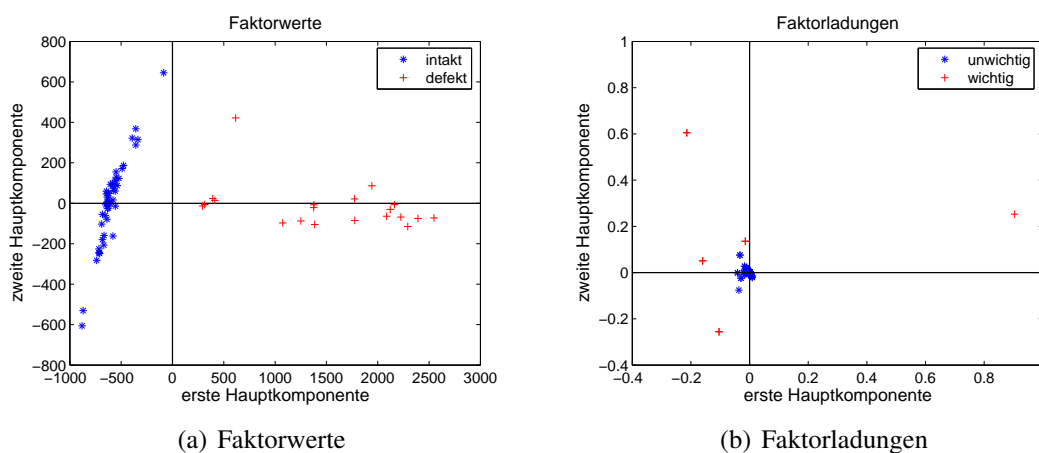


Bild 5: praktisches Beispiel einer Hauptkomponentenanalyse mit vollständigem Merkmalsvektor

Im Bild 6 ist das Resultat nach der Reduzierung um die als ”unwichtig” erkannten Frequenzen dargestellt. Nach der Datenreduktion werden im Scores-Plot (Bild 6(a)) mit den

ersten beiden PC nahezu 100% der Information erklärt. Dabei enthält die erste PC bereits 98% der Information. Die Trennbarkeit der beiden Datengruppen bleibt trotz der starken Reduktion des Datensatzes erhalten.

Im Bild 6(b) (Faktorladungen) fehlen folglich die "unwichtigen" (*) Frequenzen und nur die fünf wichtigsten (+) Frequenzen sind zu sehen.

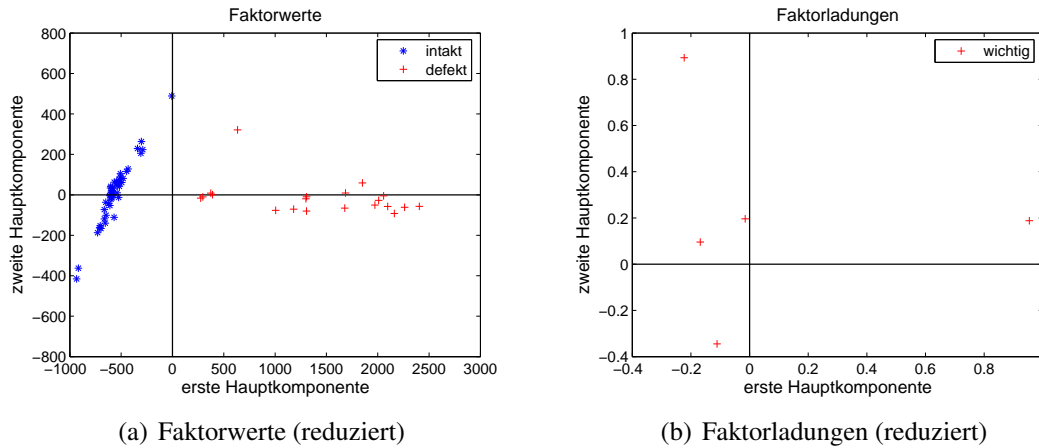


Bild 6: praktisches Beispiel einer Hauptkomponentenanalyse mit reduziertem Merkmalsvektor

3 Prädiktive Sensordiagnose

Erfolgt die Änderung der Klassenzugehörigkeit eines Sensors kontinuierlich, z.B. durch Drift und nicht sprunghaft kann neben einer physikalisch-chemischen Modellierung auch ein mathematischer Zusammenhang basierend auf einer PCA mehrerer Einzelmerkmale durchgeführt werden. Je nach Belastungsart und -dauer ist dann im \mathbb{R}^n eine andere Klassenzugehörigkeit zu erkennen.

Im Bild 7 ist als einfaches Beispiel einer Prädiktion eine lineare Extrapolation über der Zeit dargestellt, die über zwei Warn- und zwei Alarm-Schwellen verfügt.

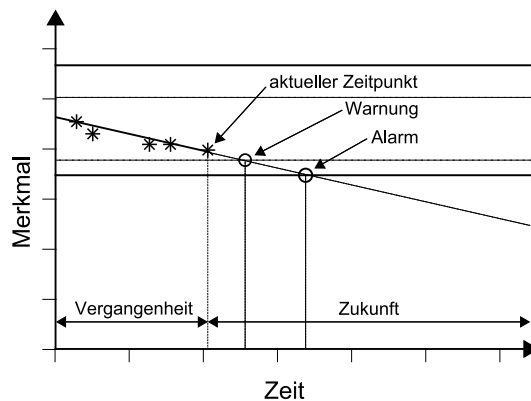


Bild 7: Prädiktive Sensordiagnose

Somit kann basierend auf dem aktuellen Sensorzustand, der Sensorhistorie, dem aktuellen Prozesszustand, der Prozesshistorie und entsprechendem Expertenwissen mit einem

geeigneten Modell eine prädiktive Diagnoseaussage getroffen werden. Von besonderem Interesse sind hierbei die verbleibende Zeit bis zum Erreichen einer Warn- bzw. einer Alarmschwelle. Damit lassen sich alle in der Literatur [6, 7] abgeleiteten Vorteile der prädiktiven Wartung wie Kostenminimierung der Wartungsarbeiten und Maximierung der Auslastung nutzen.

4 Zusammenfassung und Ausblick

Auf Grund der hohen Anforderungen an technische Systeme wurde in Abschnitt 1 das Schema der erweiterten Diagnose bei Sensoren vorgestellt. Als Grundvoraussetzung für die Umsetzung dieses Schemas muss der Sensor über ein Mindestmaß an elektronischer Datenverarbeitung verfügen. Für die Aufgabe des Klassifikators/Prädiktors in Bild 1 wurde in Abschnitt 2 die Methode der Hauptkomponentenanalyse (PCA) diskutiert. Mit dieser Art von Transformation ist die Beschreibung von abstrakten Zuständen, die von vielen Parametern beeinflusst werden, mit wenigen Hauptkomponenten möglich. Der Entwurf eines Klassifikators zur Unterscheidung der Sensorzustände liefert, basierend auf der PCA, bereits im ein- oder zweidimensionalen Raum anwendbare Resultate. In Abschnitt 3 wird die Möglichkeit vorgestellt die klassifizierten Sensorzustände über der Zeit zu betrachten und mittels einfachen mathematischen Modellen eine prädierte Diagnoseaussagen zu treffen.

Bei zukünftigen Untersuchungen werden sowohl weitere Sensoren einbezogen und mit den bereits entwickelten Methoden untersucht, als auch weitere Methoden an den bereits durchgeführten Analysen getestet. So lassen sich diese Analyseschritte neben der PCA auch mit den Methoden aus dem Bereich der Fuzzy-Logik oder der künstlichen neuronalen Netze (KNN) umsetzen. Wie in [8] beschrieben, können auch bei der Sensordiagnose komplexe Klassifizierungsaufgaben mit Hilfe der KNN bewältigt werden.

Literatur

- [1] NE 107: Selbstüberwachung und Diagnose von Feldgeräten. NAMUR. 2006.
- [2] Feid, M.; Pandit, M.: Systematischer Entwurf eines signalbasierten Fehlerdiagnosesystems, Teil 1. *at – Automatisierungstechnik* 54 (2006) 1, S. 23 – 28.
- [3] Runkler, T.: *Information Mining*. Braunschweig, Wiesbaden: Vieweg. 2000.
- [4] Martens, H.; Næs, T.: *Multivariate Calibration*. Chichester: John Wiley & Sons. 1989.
- [5] Otto, M.: *Chemometrie*. Weinheim: VCH. 1997.
- [6] Mobley, R.: *An Introduction to Predictive Maintenance*. Butterworth Heinemann, 2. Aufl. 2002.
- [7] Ruan, D.; Fantoni, P.: *Power Plant Surveillance and Diagnostics – Applied Research with Artificial Intelligence*. Berlin Heidelberg New York: Springer. 2002.
- [8] Babel, W.: *Einsatzmöglichkeiten neuronaler Netze in der Industrie*. Renningen-Malmsheim: Expert. 1997.

Optimierungsplattform für verteilte Computational Intelligence Controller

Alexander Grau, Johannes Brenig, Jörg Krone, Ulrich Lehmann

Fachhochschule Südwestfalen, Abteilung Iserlohn
Computational Intelligence and Control Laboratory
Prof. Dipl.-Ing. Dipl.-Ing. Ulrich Lehmann
D-58644 Iserlohn, Germany
Tel. (+49)-(0)2371/566-180
Fax (+49)-(0)2371/566-209
E-Mail: lehmann@fh-swf.de

1 Abstract

Kompaktregler mit "Direct Digital Control" (DDC), also digital basierte Regler bieten heute einen Funktionsumfang, der längst über den eines klassischen analogen Reglers hinausgeht. Viele Regelungsprozesse der Industrieautomatisierung mit eher kleineren, jedoch nicht weniger anspruchsvollen Anlagen, können durch Kompaktregler durchgeführt werden, vorausgesetzt diese sind "intelligent" genug.

In früheren Projekten [1] wurde bereits die Idee des CI (Computational Intelligence)-Controllers (Reglers) umgesetzt. Dabei soll sich der Regler ohne Expertenwissen an eine Regelstrecke adaptieren und auch bei nichtlinearen und zeitinvarianten Regelstrecken ständig auf optimale Regelgüte einstellen.

Im Rahmen des gleichnamigen Projektes wurde die Idee erneut aufgenommen und ein Softwaresystem entwickelt, mit der die Architektur des CI-Controllers (CI-Reglers) mit kommerziellen Tools in einem Client-/Server-System angewendet werden kann.

Ausgehend von einem Kompaktregler (Front-End) [2], auf dem die CI-Regler-Architektur übertragen wurde (virtuelles Front-End), konnten die Teile der Client-/Server-Systems prototypisch entwickelt werden.

Das virtuelle Front-End verfügt über fertige Komponenten zur Reglerentwicklung (Industrie-PID-Regler, Komponenten zur Identifikation von Regelstrecken, KNN-(Künstlich Neuronales Netz) und Fuzzy-Komponenten) und eignet sich damit insbesondere für die Kombination aus Fuzzy-Logik, Neuronalen Netzen und evolutionären Algorithmen für die Regelung. Die Echtzeitfunktionen liegen im Kompaktregler - Abtastzeiten im Millisekundenbereich sind damit durchaus erreichbar.

Kommerzielle Tools zur Reglerentwicklung (WinFACT¹) und Optimierung (MATLAB², WinFACT, SNNS/JNNS³) können eingesetzt werden, um den CI-Regler

¹ WinFACT vom Ingenieurbüro Dr. Kahlert ist ein Programmsystem zur Analyse, Synthese und Simulation konventioneller Regelungs- sowie Fuzzy-Systeme

² MATLAB von der Firma MathWorks ist eine Sprache zur technischen Berechnung und vereint Berechnung, Visualisierung und Programmierung in einer Entwicklungsumgebung

³ Der "Stuttgart Neuronale Netze Simulator" ist eine Software zur Simulation Künstlich Neuronaler Netze und wurde am "Institute for Parallel and Distributed High Performance Systems (IPVR)" an der Universität Stuttgart entwickelt

sowohl in der Simulationsumgebung als auch in der Zielentwicklung betreiben zu können.

Das Ergebnis des Projektes ist ein funktionsfähiger Prototyp bestehend aus einem virtuellen Front-End, einem Server zur Überwachung, Visualisierung und Sammlung von Wissen über die Regelstrecke (Trainingsdaten) und Funktionsbibliotheken zur Optimierung des Reglers mit kommerziellen Tools. Alle Komponenten laufen auf einem herkömmlichen PC und sind damit kostengünstig realisierbar.

2 Einleitung

Ziel dieses Projektes war es, eine Plattform zur Optimierung von "intelligenten" Reglern zu entwickeln. Dabei sollte eine Software entwickelt werden, mit der ein bestehender Kompaktregler (sog. Front-End) basierend auf einem Mikrocontrollersystem um die Möglichkeit der Optimierung erweitert wird. Des Weiteren sollten dessen Prozessdaten visualisiert und die Grundlagen geschaffen, diese Optimierung in verschiedenen Entwicklungsumgebungen durchführen zu können.

Das Projekt baut auf vorangegangenen Projekten auf, bei denen das erwähnte Mikrocontrollersystem entwickelt wurde, welches bereits eine HTTP-Schnittstelle zu einem adaptiven PID-Regler enthält. An dieser Stelle setzt dieses Projekt an, indem einerseits ein virtuelles Front-End um Komponenten für ein Künstlich Neuronales Netz (KNN) und Fuzzy-System erweitert, und andererseits eine Softwareschnittstelle entwickelt wird, mit der dieses Front-End-System mit einem über ein Netzwerk verbundenen PC optimiert werden kann.

3 Design und Realisierung

3.1 Anforderungen

Wie in der Vergangenheit in verschiedenen Arbeiten dargestellt wurde [3], kann die Regelung eines Prozesses, dessen Streckenparameter sich im Laufe der Zeit (z.B. durch Verschleiß und Alterung) verändert haben, durch eine Optimierung derart verbessert werden, dass beispielsweise ein Überschwingen minimiert wird. Hierzu gibt es unterschiedliche Ansätze und Modelle.

Das folgende Optimierungsbeispiel eines Neuro-PID-Reglers soll exemplarisch verdeutlichen, wie die Optimierung erfolgen könnte und welche Anforderungen sich hieraus ergeben. Bei diesem Ansatz soll ein Künstlich Neuronales Netz (KNN) aus den ermittelten Güteparametern der Regelung (K_s , d_{xm} , T_{AN} , T_{AUS}) die Reglerparameter eines PID-Reglers ermitteln. Damit die Güteparameter im eingeschwungenen Zustand ermittelt werden können (Online-Betrieb), wird hierbei für kurze Zeit ein kleines, deterministisches Störsignal aufgeschaltet, anhand dessen die Regelstrecke identifiziert werden kann.

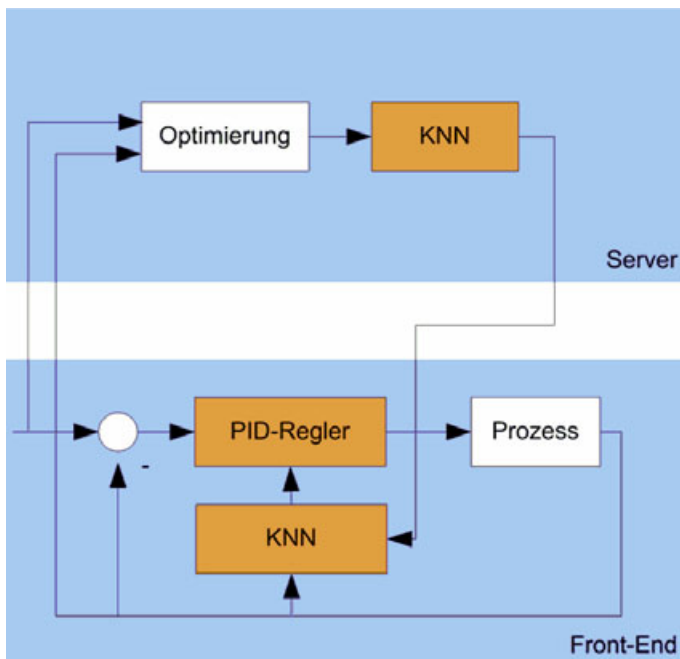


Abbildung 1: Modell zur Optimierung eines Neuro-PID-Reglers

Sobald die Identifikation eine Abweichung festgestellt hat, findet auf einem über Netzwerk verbundenen Server die Optimierung statt. Hierzu wird ein KNN mit der Historie der Güteparameter am Eingang und den dazugehörigen PID-Parametern (welche beispielsweise zuvor mit einem evolutionärem Algorithmus ermittelt wurden) am Ausgang trainiert. Das trainierte KNN wird sodann in das Front-End übertragen, so dass bei einem trainierten Arbeitspunkt die Reglerparameter angewendet werden können. Verlässt der Regler irgendwann wieder den zugelassenen Toleranzbereich, wiederholt sich der beschriebene Vorgang, - das KNN liefert also mit der Zeit zu immer mehr Arbeitspunkten passende Reglerparameter.

Bei der Analyse für die benötigte Funktionalität zur Optimierung haben sich folgende Funktionen für das Optimierungs-Protokoll herausgestellt:

- Abfrage von Prozessdaten
- Abfrage von Güte-Logging
- Transport von Parametern zur Konfiguration des Reglers
- Transport von KNN- und Normierungsdaten
- Transport von Fuzzy-Systemen

Für das zu entwickelnde Protokoll musste ferner berücksichtigt werden, dass dieses auf dem HTTP-Protokoll basieren soll und damit ein zustandloses Protokoll ist (Front-End bzw. Server also keine Information darüber besitzen, von wem sie eine Anfrage erhalten haben). Daher muss eine Anfrage "ohne Zusatzwissen" eindeutig beantwortet werden können, wenn z.B. eine Anwendung fortlaufend Prozessdaten der letzten halben Stunde anfordert.

Die folgende Abbildung zeigt die erstellte Software und dessen Interaktion mit den beteiligten Hard- und Softwarekomponenten.

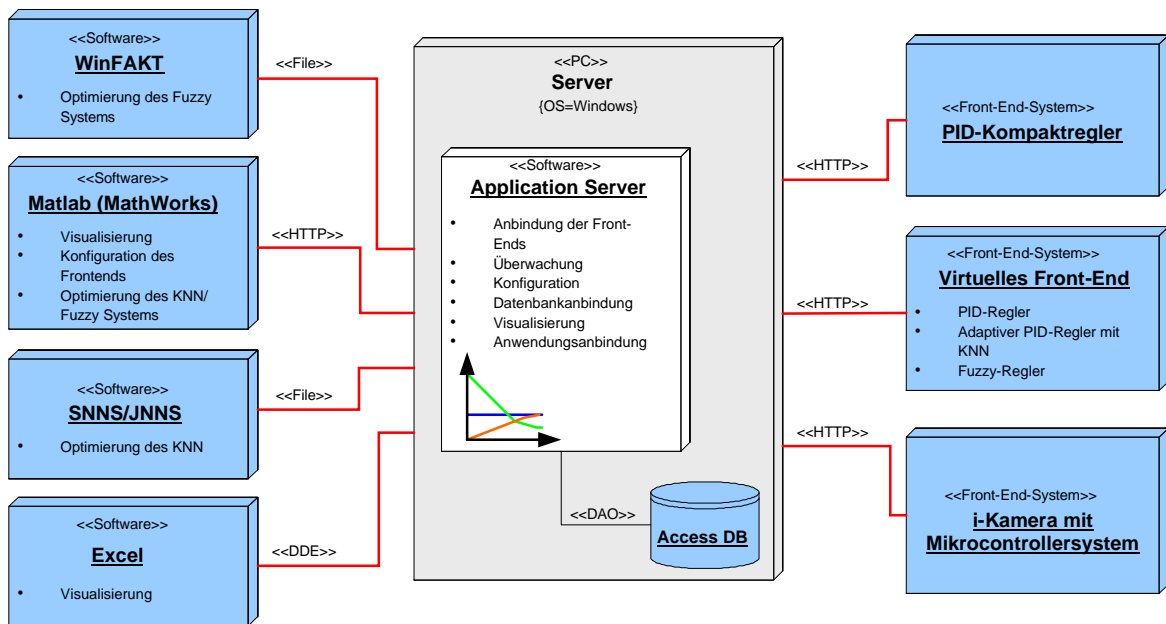


Abbildung 2: Architektur des Gesamtsystems

Das virtuelle Front-End kommuniziert (wie auch die anderen Front-Ends) über das zu entwickelnde FES-Protokoll mit dem Application Server. Es dient der Abfrage von Prozessdaten und dem Transport von Parametern zur Konfiguration sowie dem Transport des Künstlich Neuronales Netzes und des Fuzzy Systems.

Das Prozess- und Güte-Logging wird in regelmäßigen Zeitabständen (beispielsweise 10 Sekunden) vom Application Server angefordert und in einer Access-Datenbank abgelegt. Der Application Server seinerseits stellt eine Art "Proxy-Server" dar – über die von ihm angebotene HTTP-Schnittstelle kommunizieren die Anwendungen zur Optimierung über das gleiche Protokoll wie die des Front-Ends. Es werden also die meisten Anfragen (bis auf Anfragen, die die Datenbank betreffen) zum Front-End "durchgereicht", und die Antwort des Front-Ends wieder zur Anwendung zurückgereicht. Durch diesen Ansatz ist es möglich, die Anwendung zu Testzwecken auch direkt mit einem Front-End zu betreiben. In diesem Fall muss jedoch für die Logging-Abtastzeit in der Anwendung

$$t_{ab,Application} < t_{ab,Frontend}$$

eingehalten werden, damit kein Buffer-Overflow am Front-End eintritt und Daten verloren gehen.

3.2 Virtuelles Front-End System

Folgende Abbildung gibt die generelle Funktionsweise des virtuellen Front-End Systems anhand eines der implementierten Reglermodelle, einem adaptiven PID-Regler (ADAPID) wieder. Der anhand der Reglerparameter und der Regeldifferenz berechnete Reglerausgangswert (Reglerstellwert) wird über einen nachgebildeten Digital-Analog-

Wandler einer PT-4 Regelstrecke zugeführt. Der Ausgang der Regelstrecke wird über einen Analog-Digital-Wandler wieder dem Regler zugeführt.

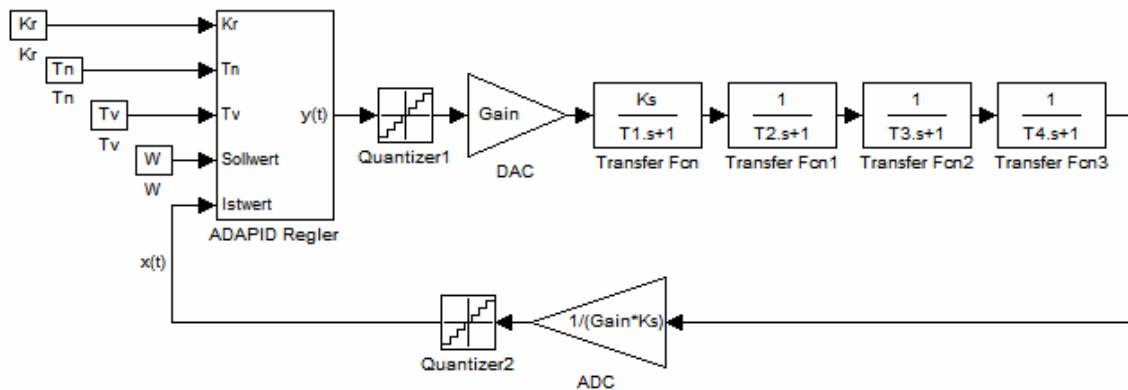


Abbildung 3: Blockdiagramm virtuelles Front-End

Das Front-End System wurde in C++ für das Betriebssystem Windows entwickelt. Über einer grafischen Benutzeroberfläche können regler-spezifische Parameter wie beispielsweise Regelverstärkung K_R , Nachstellzeit T_N , Vorhaltezeit T_V , Verzögerungszeit T_{VZ} , Abtastzeit T_{ab} usw. eingestellt werden.

Der zu simulierende Prozess wird durch eine PT-4 Regelstrecke mit konfigurierbarer Übertragungsfunktion eingestellt; dem Streckenausgangssignal kann ein Störsprung aufaddiert werden und die Strecke dynamisch verändert werden (zeitliche Veränderung der Streckenverstärkung K_S und größten Zeitkonstante).

Das Front-End enthält einen HTTP-Server, der das entwickelte FES-Protokoll implementiert. Dieses Protokoll ist zustandslos, d.h. der Web-Server behandelt alle Anfragen unabhängig voneinander, es werden keine Informationen über zuvor bereits durchgeführte Anfragen gespeichert.

Ein weiterer Bestandteil dieses Front-Ends ist das automatische Güte-Logging: Verlässt die Regelung ein fest eingestelltes Toleranzband, wird ein Testsprung ausgeführt und aus den ermittelten Daten ein Güte-Logging-Eintrag erzeugt.

In zwei weiteren Modulen wurde die Funktionalität für einen Neuro-PID- und einen Fuzzy-PID-Regler implementiert. Hervorzuheben ist, dass die so entwickelten Reglermodelle, welche später aus der Ferne optimiert werden sollen, nicht mit herkömmlichen Programmiersprachen entwickelt werden müssen, sondern mit einem blockorientiertem Simulationssystem (WinFACT BORIS) entworfen werden können. Hierbei können das im Front-End implementierte KNN und Fuzzy-System beliebig angewendet werden, d.h. es existiert eine Schnittstelle für die genannte Simulationsumgebung.

Die folgende Ausführung zeigt wie eines der Reglermodelle umgesetzt wurde. Mit den eingestellten Reglerparametern wird zunächst ein PID-Regler betrieben. Verlässt die Regelung ein fest eingestelltes Toleranzband, wird ein Testsprung ausgeführt und aus den ermittelten Daten ein Güte-Logging-Eintrag erzeugt. Anhand dieses Güte-Logging Eintrags kann dann über das FES-Protokoll von einer Optimierungsanwendung (z.B. MATLAB) eine Optimierung des Künstlich Neuronales Netzes durchgeführt werden. Das hier eingesetzte Künstlich Neuronales Netz (Feed-Forward Netz mit 2 Hidden-Layer mit 6 und 5 Neuronen) wurde jeweils mit den Trainingsdaten mit Hilfe von SNNS und MATLAB trainiert und dann über das FES-Protokoll ins Front-End geladen (Download). Es besitzt die folgenden Eingänge:

Eingang	Symbol	Bezeichnung
1	K_s	Streckenverstärkung
2	K_r	Reglerverstärkung
3	T_n	Nachstellzeit (s)
4	T_v	Vorhaltezeit (s)
5	Δx_m	Überschwingweite (%)
6	T_{an}	Anregelzeit (s)
7	T_{aus}	Ausregelzeit (s)

Die Ausgänge des KNN sind:

Ausgang	Symbol	Bezeichnung
1	ΔK_r	Delta Reglerverstärkung
2	ΔT_n	Delta Nachstellzeit (s)
3	ΔT_v	Delta Vorhaltezeit (s)

Jedes mal, wenn das Toleranzband verlassen wird, wird versucht, die Strecke zu identifizieren, um die Reglerparameter entsprechend zu korrigieren. Dazu berechnet das KNN anhand der Güteparameter die Differenz zu den neuen Reglerparametern. Diese werden mit den aktuellen Parametern verrechnet und schließlich neu gesetzt.

Ein analoges Modell wurde für einen Fuzzy-PID-Regler entwickelt.

3.3 Application Server

Der Application Server wurde ebenfalls in C++ entwickelt und umfasst folgende Funktionalität:

- Überwachung einzelner Front-Ends
- Visualisierung der Prozessdaten eines Front-Ends
- Serverdienst zum Abfragen der Prozess- und Gütedaten von externen Anwendungen
- Funktionen zur manuellen Änderung von Reglerparametern und zum manuellen Upload von Künstlich Neuronalen Netzen und Fuzzy Systemen
- Proxy-Server zur Weiterleitung von Anfragen von externen Anwendungen an ein Front-End und Weiterleitung der Antwort des Front-Ends an die Anwendung

Prozess- und Güte logging des Front-Ends werden im Application Server in einer MS Access-Datenbank abgelegt.

Der Application Server stellt für die FES Client Anwendungen eine Art Proxy-Server dar. Alle Anfragen, die nicht direkt vom Application Server behandelt werden können, werden zum Front-End weitergeleitet. Dessen Antwort wird sodann wieder zum Client zurückgeschickt.

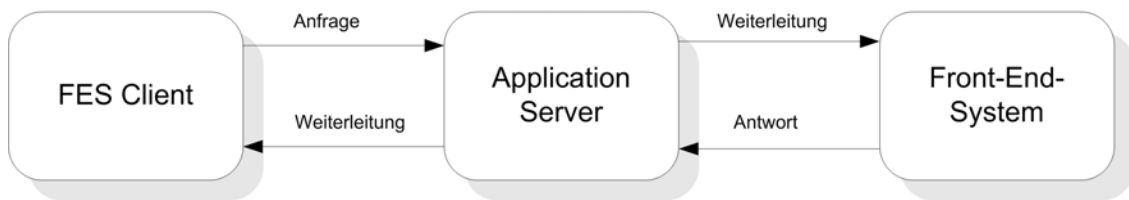


Abbildung 4: Application Server als Proxy

In einer Benutzeroberfläche können die zu optimierenden Front-Ends angezeigt und konfiguriert werden.

Für die Simulationsumgebung WinFACT und den Stuttgarter Neuronale Netze Simulator (SNNS) existiert im Application Server die Möglichkeit, ein KNN und Fuzzy-System über Dateien in ein Front-End zu übertragen (Upload), da diese keine oder unzureichende Möglichkeiten zur automatisierten Optimierung bieten.

Des Weiteren gibt es eine DDE-Schnittstelle, mit der Prozess- und Güte-Logging in andere DDE-fähige Anwendungen (wie beispielsweise Excel) exportiert werden können.

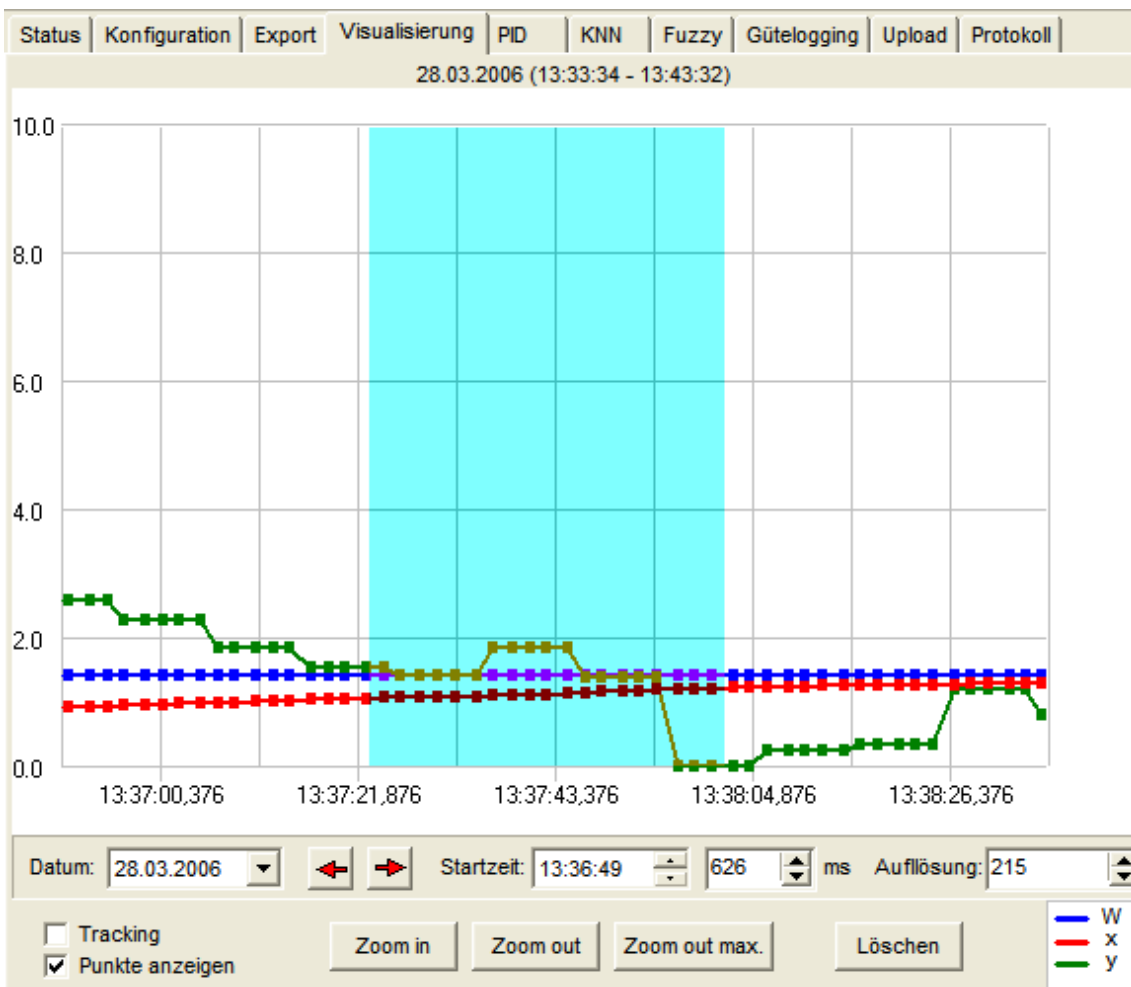


Abbildung 5: Visualisierung des Regelungsverlaufs im Application Server

3.4 Realisierte Schnittstellen

Damit die Optimierungsplattform so universell wie möglich anwendbar ist, wurden eine Reihe von Schnittstellen geschaffen, welche folgende Abbildung wiedergibt:

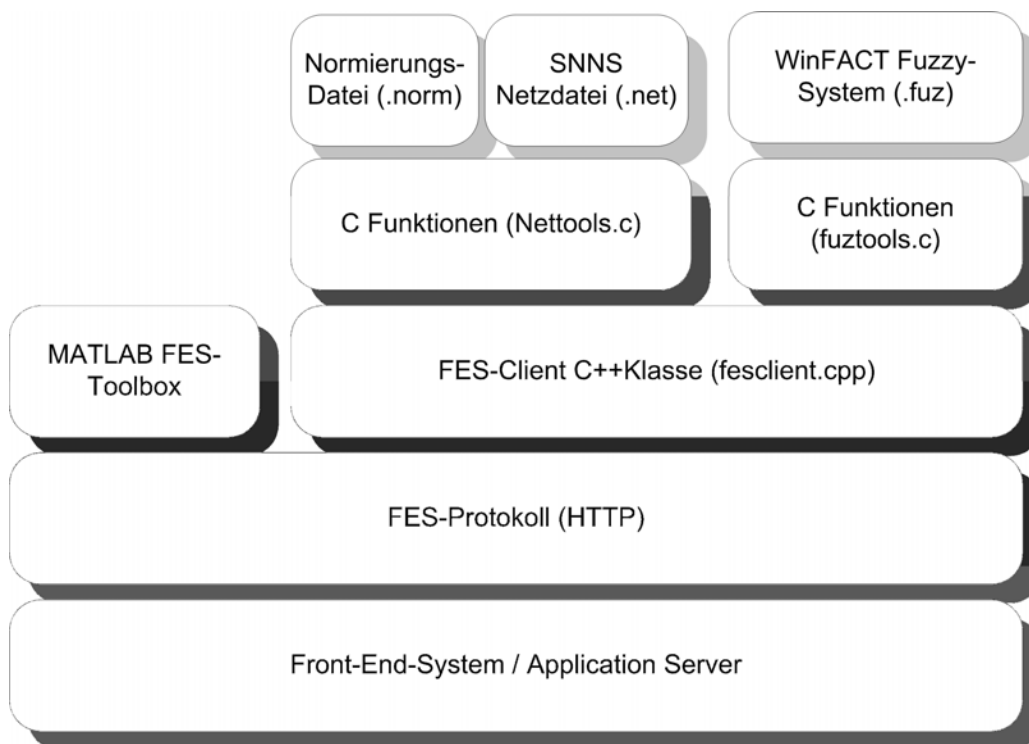


Abbildung 6: Zusammenfassung der Schnittstellen zur Optimierung

Einige der Schnittstellen wurden bereits genannt. Ziel der Optimierungsplattform war es, ebenfalls Schnittstellen für kommerzielle Tools anzubieten.

4 Anwendung, Bewertung

Um die Integration des FES-Protokolls mit den MATLAB-Bibliotheken "Neural Network Toolbox" und "Fuzzy Logic Toolbox" so nahtlos wie möglich zu schaffen, wurde eine "FES-Toolbox" entwickelt.

Hauptbestandteil der Funktionen zum Übertragen des KNN oder Fuzzy-Systems ist eine Transformation der MATLAB-spezifischen Datenstrukturen in das FES-Protokoll-Format.

Beispiele für die Anwendung der vorgestellten Optimierungsplattform:

- Neue Reglermodelle können mit einer Simulationsumgebung (WinFACT) erstellt und mit der vorgestellten Plattform eingesetzt und optimiert werden.
- Alle KNN und Fuzzy-Systeme, welche in einem Reglermodell eingesetzt werden, können über die Plattform mit externer kommerzieller Software optimiert werden (hier: MATLAB und SNNS)
- Optimierung von KNN und Fuzzy-Systemen aus einer C++ Anwendung

In diesem Projekt wurde ein herkömmlicher digitaler PID-Regler mit KNN- und Fuzzy-Komponenten zu einem CI-Regler erweitert. Ein Application Server wurde entwickelt, um diesen CI-Regler von der Optimierungsanwendung zu entkoppeln, dessen Logging-Daten aufzuzeichnen, und diesen überwachen zu können. Es wurden Schnittstellen entwickelt, die es ermöglichen, diesen CI-Regler über ein Netzwerk mit ausgesuchten Entwicklungsumgebungen optimieren zu können.

Der für dieses Projekt entworfene Prototyp zeigt, dass mit den zuvor genannten Entwicklungsumgebungen nun eine Optimierung prinzipiell möglich ist. Allerdings hat die Lösung – wie bei jeder Software – noch Punkte, die man verbessern kann (siehe Ausblick).

5 Ausblick

Das entwickelte System zur Optimierung öffnet ein neues Feld für Optimierungs-Komponenten und stellt mit Sicherheit die Grundlage für weitere Diplom- und Projektarbeiten dar. Insbesondere folgende Themen schließen sich nahtlos daran an:

- Den entwickelten Code des virtuellen Front-Ends für einen produktiv einsetzbaren Kompaktregler anpassen
- OPC-Unterstützung zur Prozessautomatisierung mit anderen OPC-fähigen Geräten hinzufügen und dadurch um einen wichtigen industriellen Automatisierungsstandard erweitern
- Entwicklung einer optimierbaren WinFACT-Fuzzy-Komponente (.DLL) mit Integration des FES-Protokolls und damit die Optimierung innerhalb einer Simulationsumgebung ermöglichen
- Den SNNS- bzw. JNNS-Kernel mit einem C++ bzw. Java-Programm ansprechen, mit Integration des FES-Protokolls (zur Online-Optimierung, vergleichbar mit der bisher umgesetzten MATLAB-Implementierung)

6 Danksagung

Die Arbeiten wurden im Rahmen der Kompetenzplattform Computer Vision based on Computational Intelligence vom Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen gefördert.

7 Literatur

- [1] Ulrich Lehmann, Jörg Krone, Johannes Brenig, Tobias Mund, Udo Reitz, Roman Schamne: *Computational Intelligence-Regler (CI-Controller)*. VDI GMA und GI Workshop Fuzzy-Control, Dortmund/Witten, 19. bis 21. November 2003.
- [2] Christian Beckert: *Front-End System für verteilte Computational Intelligence Anwendungen*. Diplomarbeit, Iserlohn, Juli 2002
- [3] Sascha Beuchel, Frank Brüggemann: *Neuro-PID-Temperaturregelung für den Extruder einer Kunststoffverarbeitungsmaschine*, Diplomarbeit, Iserlohn, Juli 1999

European Monetary Union Revisited: How Good was the Fuzzy Cognitive Map Prediction?

Christina Katz¹, Thomas A. Runkler¹, Jürgen Hollatz²

¹ Siemens Corporate Technology Information & Communications
CT IC 4
81730 München, Germany

² Siemens Professional Education
SPE CA 8
Günther–Scharowsky–Straße 2
91058 Erlangen, Germany

Abstract

At the 1999 workshop Fuzzy–Neuro–Systems in Leipzig we presented a fuzzy cognitive map to predict the effects of the European Monetary Union. Now, seven years later, we reconsider this work and compare our predictions from 1999 with the real economic development based on statistics from the German Federal Statistical Office. The comparison shows that the changes in prices, investments, foreign trade, economic competition, and taxes were predicted correctly, while the predictions for the debts and employment were wrong. A closer analysis shows that the incorrect prediction of the debts resulted from an over–estimation of the influence of the European Central Bank on the national debts and the budget discipline. The predictions for the employment were only correct on a time horizon of two years. On a longer time horizon our prediction of the employment was not correct, which indicates the existence of additional influences that had not been included in the fuzzy cognitive map. Since most of the predictions have turned out to be correct, fuzzy cognitive maps have proven to be attractive methods for the knowledge based modeling and prediction of the impact of unique incidents (such as the European Monetary Union) on economic systems.

1 Introduction

In January 1999 the *Euro* was introduced to world financial markets in the following 11 European countries: Austria, Belgium, Finland, France, Germany, Ireland, Italy, Luxembourg, Netherlands, Portugal, and Spain [12]. From January 1999 to December 2001 the Euro existed only as a deposit currency, but launched as a currency in January 2002. In 2001 Greece also joined the Euro countries. Besides these 12 European countries, the following six countries outside of the EU launched the Euro as a legal currency: Andorra, Kosovo, Monaco, Montenegro, San Marino, and Vatican City.

Before its introduction many economists and politicians made predictions which effects the Euro could have on the European and the German economy and society. All these predictions explicitly based on the knowledge and experiences of the economists. In our approach expert knowledge was only implicitly used. Experts were interviewed and from the results of these interviews a fuzzy cognitive map (FCM) was built that was then used to predict the effects of the introduction of the Euro to the German economy and society. Another difference to a usually economic prediction is, that here the experts provided a

general model of the economy instead of only predicting the reaction on the introduction of the Euro.

Our paper is structured as follows: Sections 2 and 3 give a short introduction to fuzzy cognitive maps and briefly review the fuzzy cognitive map that we designed in 1999 to predict the effects of the introduction of the Euro. Section 4 specifies the data base that was used for our analysis, and Sections 5–8 present the detailed analysis for prices and gross domestic product, foreign trade, debts and taxes, and employment. Our conclusions are finally given in Section 9.

2 Fuzzy Cognitive Maps

Fuzzy cognitive maps (FCM) [7, 11] are directed graphs whose nodes represent variables, and whose (directed) edges represent causal relationships between pairs of variables. The edge weights $w_{ij} \in [-1, 1]$ quantify the strengths of the relationship, where positive edge weights represent excitatory causalities, and negative edge weights represent inhibitory causalities. For example, a directed edge from a to b with weight 1 is associated with the following semantics: If a increases then b will increase, and if a decreases then b will decrease. The same edge with weight -1 has the following semantics: If a increases then b will decrease, and if a decreases then b will increase. Positive edges with lower weight and negative edges with higher weight indicate the same causal relation, but to a lesser extent. Edges with zero weight in particular are equivalent to non-existing edges and indicate the absence of a causal relation between the corresponding pair of variables. The dynamics of these causalities are not explicitly modeled in a fuzzy cognitive map. The causal relations are assumed to become effective immediately, i.e. in the subsequent simulation step.

Applications of fuzzy cognitive maps include stock market modeling [8], clinical treatment modeling [3], decision support in geographic information systems [10], decision support in urban design [13], strategic planning simulation [9], and relationship management in airline service [5].

3 The 1999 Fuzzy Cognitive Map

In 1999 we designed a fuzzy cognitive map for the prediction of the effects of the introduction of the Euro [4]. The basis for this fuzzy cognitive map were several interviews with experts in politics and economy from the University of the Federal Armed Forces in Munich. The interviews and the implementation of the fuzzy cognitive map were done by the internship students Markus Knoll and Klaus Lichtenfeld [6] based on a JAVA tool written by Stephen Moor.

The structure of the FCM model is shown in Fig. 1. First we designed a socio-economic model that represents the situation *before* the introduction of the Euro. This socio-economic model consists of a model of the German economy, a model of the German society, and four nodes that are strongly relevant for both economy and society and are therefore used to connect both models: the *attractivity* of Germany to foreign investors and workers, the relative *employment*, the *stability* involving economic requirements (full employment, balance on merchandise trade, price stability, economic growth) as well as

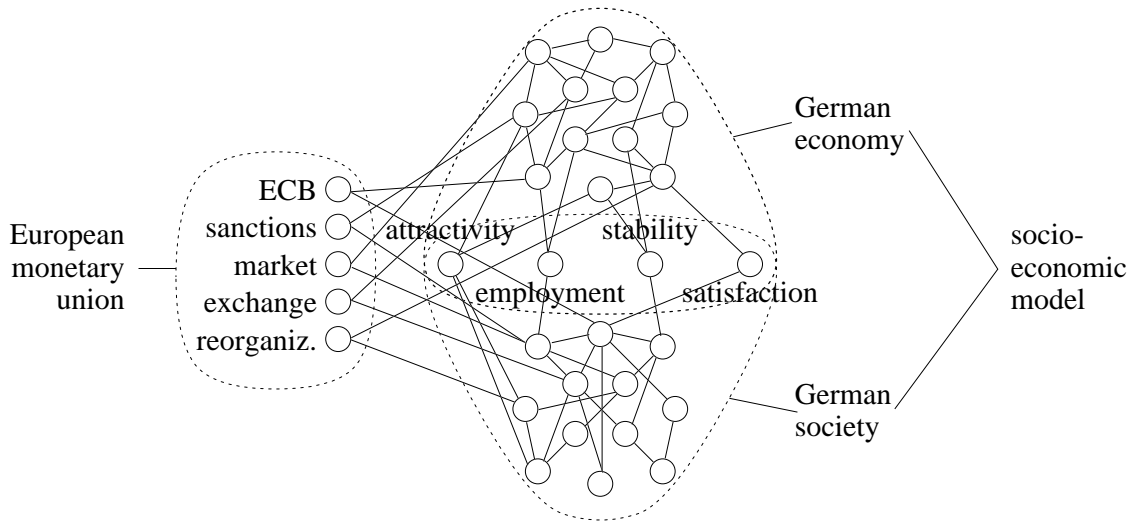


Figure 1: Architecture of the 1999 FCM model (taken from [4])

social stability, and the subjective *satisfaction* of the German citizens. We were mainly interested to investigate the influence of the introduction of the Euro on five key indicators that were used as nodes inside the economic model: the economic *competition*, the *debts*, the *dollar dependence* of the economy, the *profit* of the companies, and the *prices*. In the next step we added five nodes that represent the changes caused by the introduction of the Euro and connected these nodes to those nodes in the socio-economic model that are most influenced by these five Euro nodes: the *European Central Bank (ECB)* which controls the financial policies of the EMU members, the possible *sanctions* against EMU members violating the financial rules of the Maastricht contract, the *common European market*, the drop of *exchange costs* for foreign transactions, and the *reorganization effort*, e.g. for financial computer systems. These five nodes are designed to be simply switched on at some time step during simulation in order to be able to easily simulate the transition phase from the pre-Euro to the post-Euro model.

After the design of the fuzzy cognitive map we first switched off the five Euro nodes and simulated the model until an equilibrium was reached. Then we switched on the five Euro nodes and continued the simulation, for three single steps first and then again until an equilibrium was reached. This simulation yielded the values presented in the following table (taken from [4]):

	pre-Euro	step 1	step 2	step 3	post-Euro
attractivity	0.54	0.54	0.53	0.54	0.55
employment	0.43	0.43	0.48	0.48	0.47
stability	0.7	0.7	0.71	0.73	0.73
satisfaction	0.49	0.55	0.51	0.51	0.52
competition	0.71	0.83	0.83	0.84	0.84
debts	0.42	0.23	0.26	0.30	0.30
dollar dependence	0.48	0.33	0.33	0.33	0.33
profit	0.54	0.69	0.62	0.62	0.62
prices	0.64	0.64	0.63	0.61	0.62

The changes caused by the Euro are very small, which indicates economic and social stability. Without further analysis we can say that this general prediction was definitively

correct. If we take a closer look at the individual variables, we observe that following our model the attractivity, the employment, the stability, the satisfaction, the competition, and the profit increase after the introduction of the Euro, and that the debts, the dollar dependence, and the prices decrease. The purpose of the paper at hand is to verify if this prediction was correct.

4 Data base for the analysis of the 1999 prediction

To get solid results we base our analysis on official statistics from the Federal Statistical Office Germany: “Statistisches Jahrbuch 2002” [1] and “Statistisches Jahrbuch 2005” [2]. Up to date statistics can be obtained at www.destatis.de. We deal with the statistics from the internet, because there all values have the same index, but in the literature there can be smooth differences between the editions. For example, in “Statistisches Jahrbuch 2002” 1995 is the index year for the determination of the price index whereas in “Statistisches Jahrbuch 2005” it is 2000. As a reasonable period for our analysis we chose the years 1995 to 2004, so we can follow the development of the economy from five years before the Euro until five years after.

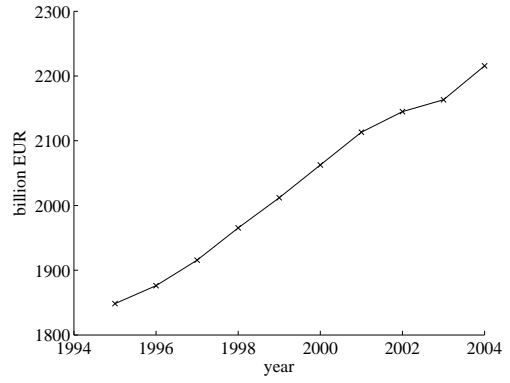
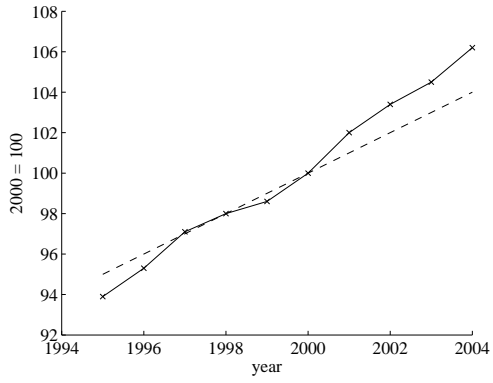
Our analysis is structured in four sections, each of which describes one part of the economy: section 5: prices and usage of gross domestic product, section 6: foreign trade, section 7: debts and taxes, and section 8: employment.

5 Prices and use of gross domestic product

The main indicator for the price development in Germany is the consumer price index, which represents changes in prices of all goods and services purchased for consumption by urban households. Following our model the prices stayed almost constant, because of an equilibrium between the positive influences, which are reorganization effort, inflation, production costs and demand, and the negative influences, which are competition and drop of exchange costs. Figure 2 (a) displays the development of the consumer price index of the years 1995 to 2004 in contrast by the curve which shows an increase of one percent per year. The comparison shows that the increase of the consume price index lies at about one percent per year.

The stability of the prices is a basic requirement for a sustainable economic increase. Because of that the ECB defined the “Harmonized Index of Consumer Prices” (HICP), which approximately reflects the consumer price index. The demand of the ECB to the Euro countries is that the HICP must not increase about more than two percent per year. If we had a null-inflation this would de facto mean a deflation, so an increase of the consumer price index of one percent per year roughly leads to constant prices. Another fact which has to be considered is that the mean rate of the price increase in the considered six years of the Euro is clearly lower than the one of the D-Mark in the 50 years of its existence.

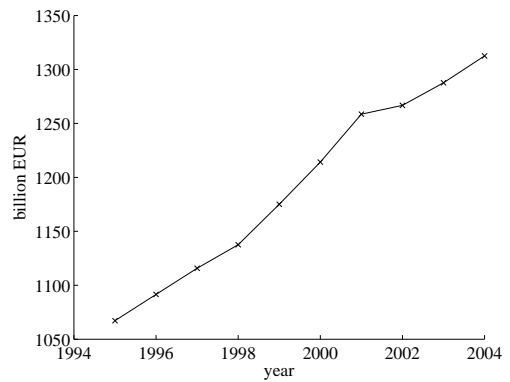
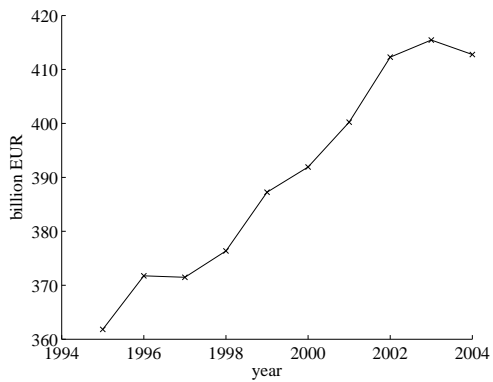
Figure 2 (b) shows the development of the gross domestic product (GDP) in Germany for the considered period. Figure 3 displays the use of the GDP, divided in (a) the consumer spending of the state and (b) the private consumer spending. Our FCM predicted a growth of the investments, which is clearly supported by both figures.



(a) Consumer price index

(b) Gross domestic product

Figure 2: Prices and gross domestic product



(a) Consumer spending state

(b) Private consumer spending

Figure 3: Use of gross domestic product

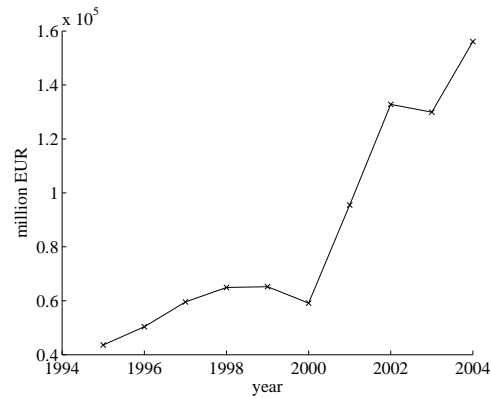
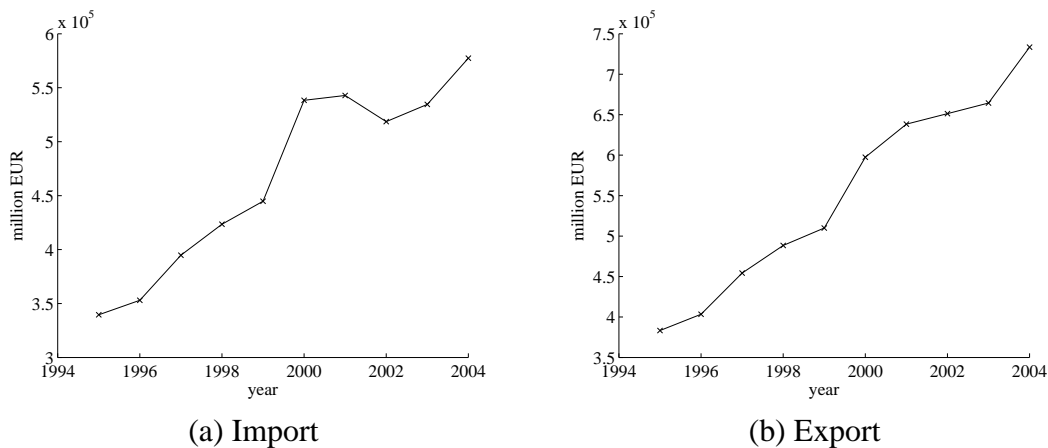


Figure 4: Foreign trade balance



(a) Import

(b) Export

Figure 5: Foreign trade

6 Foreign trade

The main indicators of foreign trade are import, export, and the foreign trade balance, which is the difference between import and export. Our model predicted a significant increase of the economic competition and the foreign trade balance.

Figure 4 shows the chart of the foreign trade balance. From 1995 to 2000 there is only a slight increase, in 1999 even a slight decrease, but from 2000 to 2004 the increase of the foreign trade balance is evident.

Figure 5 shows that (a) the imports and (b) the exports in the considered 10 years are (almost) continuously increasing. The economic competition can be regarded as the combination of the three mentioned factors of the foreign trade. So we see a clear increase of the economic competition since the introduction of the Euro.

Beside these measurable values there are also aspects like transparency or dollar dependence, which cannot be numerical quantified. For the drop of exchange costs there is no official data available. The increase of transparency and the drop of exchange costs in the European market happened as expected, because now most of the countries use the same currency, so there is less organization effort, and the prices abroad are clear to the customers. Five years after the introduction of the Euro it became a nearly equitable con-

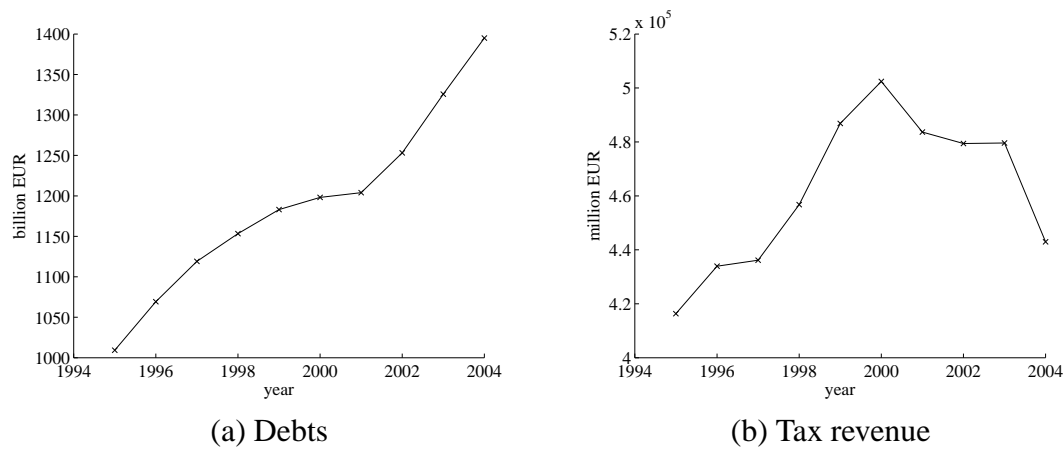


Figure 6: Debts and taxes

current for the world currency Dollar, as the Euro has a money circulation of 41%, the Dollar of 43%.

7 Debts and taxes

Following our model the introduction of the Euro should have caused a reduction of the national debts because of the ECB and its possible sanctions. However, the real economic development shows a constant increase of the national debts, see figure 6 (a). This mismatch is caused by a modeling error in the FCM. In our model the ECB, the sanctions, and the budget discipline influences the national debts in an excitatory, the inventions in an inhibitory way. A closer analysis of the fuzzy cognitive map shows that the incorrect prediction of the debts resulted from an over-estimation of the influence of the ECB on the national debts and the budget discipline. As the ECB is a fundamental factor in the European Monetary Union, naturally this failure causes several effects.

The tax revenues were predicted to decrease because of the Euro. From 1995 until the introduction of the Euro the tax revenue increased significantly. This trend continued in the next two years until 2000, but then it really decreased for the rest of the period. This effect can be seen in figure 6 (b).

8 Employment

In our model the development of the employment belongs to the common key factors of the evaluation. The value for the employment during the iterations first rose from 0.43 to 0.48, but in the last step to the post-Euro value it decreased slightly to 0.47, so there is no significant reduction of the unemployment rate. Figure 7 shows that in the first two years after the introduction of the Euro the economically active population increases, but from 2002 it decreases again. So here the prediction does not agree with the long term development of the employment, but strongly with the short dated, which indicates the existence of additional influences that had not been included in the fuzzy cognitive map.

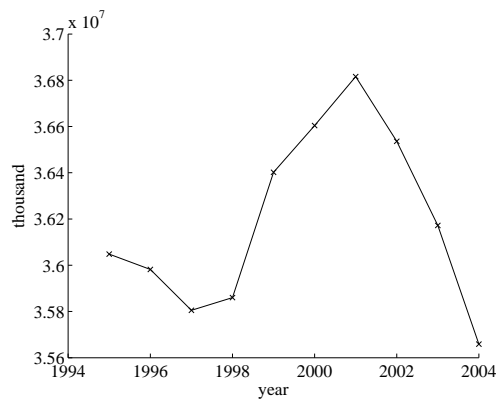


Figure 7: Economically active population

9 Conclusions

We used fuzzy cognitive maps, feeded with expert knowledge, to make predictions about the effects of the Euro to the German economy and society.

Our model predicted increases of investments, foreign trade, economic competition, employment, and decreases of taxes and debts, while the prices should remain constant. Except the aspects employment and debts, all of our predictions have been correct which shows that fuzzy cognitive maps have proven to be attractive methods for the knowledge based modeling and prediction of the impact of unique incidents (such as the European Monetary Union) on economical systems.

At the end of our 1999 paper we wrote: “all the results presented here are just predictions” and “we will have to check our results again to see whether or not the predictions of our fuzzy cognitive map have become reality”. With the paper at hand we have carried out our duty and showed that most of the predictions have turned out to be correct. The fuzzy cognitive map has proven to be a suitable tool for simulating and predicting the socio-economic effects of the introduction of the Euro. Our results indicate in general that fuzzy cognitive maps are attractive methods for the knowledge based modeling and prediction of the impact of external incidents to complex systems.

References

- [1] *Statistisches Jahrbuch 2002 für die Bundesrepublik Deutschland*. Statistisches Bundesamt, 2002.
- [2] *Statistisches Jahrbuch 2005 für die Bundesrepublik Deutschland*. Statistisches Bundesamt, 2005.
- [3] C. M. Helgason and T. H. Jobe. Necessary and sufficient causal ground and effect is measured by fuzzy cardinality and may represent natural edge strength connections in a clinical fuzzy cognitive map. In *North American Fuzzy Information Processing Society Conference*, pages 117–123, New Orleans, June 2002.

- [4] J. Hollatz and T. A. Runkler. Predicting the effects of the European Monetary Union (EMU) using fuzzy cognitive maps. In G. Brewka, R. Der, S. Gottwald, and A. Schierwagen, editors, *Fuzzy-Neuro-Systems '99, Leipzig*, pages 105–111. Leipziger Universitätsverlag, March 1999.
- [5] I. Kang, S. Lee, and J. Choi. Using fuzzy cognitive map for the relationship management in airline service. *Expert Systems with Applications*, 26(4):545–555, 2004.
- [6] M. Knoll and K. Lichtenfeld. Die Einführung des Euro und seine Auswirkungen auf die Bundesrepublik Deutschland. Universität der Bundeswehr München / Siemens AG München, 1998. Praktikumsbericht.
- [7] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice Hall, Englewood Cliffs, 1992.
- [8] D. E. Koulouriotis, I. E. Diakoulakis, and D. M. Emiris. A fuzzy cognitive map-based stock market model: synthesis, analysis and experimental results. In *IEEE International Conference on Fuzzy Systems*, pages 465–468, Melbourne, December 2001.
- [9] K. C. Lee, W. J. Lee, O. B. Kwon, J. H. Han, and P. I. Yu. Strategic planning simulation based on fuzzy cognitive map knowledge and differential game. *Simulation*, 71(5):316–327, 1998.
- [10] Z. Q. Liu and R. Satur. Contextual fuzzy cognitive map for decision support in geographic information systems. *IEEE Transactions on Fuzzy Systems*, 7(5):495–507, 1999.
- [11] C. D. Stylios, V. C. Georgopoulos, and P. P. Groumpos. The use of fuzzy cognitive maps in modeling systems. In *The 5th IEEE Mediterranean Conference on Control and Systems*, 1997.
- [12] P. Welfens. *European Monetary Union*. Springer, 2001.
- [13] G. Xirogiannis, J. Stefanou, and M. Glykas. A fuzzy cognitive map approach to support urban design. *Expert Systems with Applications*, 26(2):257–268, 2004.

Die MATLAB-Toolbox Gait-CAD

Ralf Mikut¹, Ole Burmeister¹, Markus Reischl¹, Tobias Loose²

¹Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik,
D-76021 Karlsruhe, Postfach 3640, Telefon: (07247) 82-5731, Fax: (07247) 82-5702,
E-Mail: ralf.mikut@iai.fzk.de

² 97816 Lohr am Main, Am Rechtenbach 3, Telefon: (09352) 181309,
Fax: (09352) 181293, E-Mail: tobias.loose@boschrexroth.de

1 Einleitung und Historie

Seit 1998 werden am Forschungszentrum Karlsruhe Algorithmen zur Klassifikation entwickelt und in einer Vielzahl von medizinischen und technischen Anwendungen eingesetzt. Während dieser Zeit entstand die MATLAB-Toolbox KAFKA (Karlsruher Fuzzy-Modellbildungs-, Klassifikations- und datengestützte Analyse-Toolbox) für die interne Nutzung. Es handelte sich von der Zielrichtung her in erster Linie um eine offene Entwicklungsplattform für methodische Weiterentwicklungen von Data-Mining-Verfahren und deren Spezifikation für verschiedene Anwendungsfelder. Eine Kommerzialisierung der Toolbox war wegen des erheblichen Aufwands zum Erreichen einer hohen Softwarequalität, bei der Wartung, dem Support und der Dokumentation nie geplant.

Etwa ab 2001 verschob sich im Rahmen neuer Projekte (insbesondere durch das DFG-geförderte Projekt "Diagnoseunterstützung in der Ganganalyse") der Fokus zunehmend in Richtung Klassifikation aus Zeitreihen [1]. Hierbei entstand die MATLAB-Toolbox Gait-CAD (Gait = Englisch für Gang, CAD: Computer Aided Diagnosis), die zunächst auf die Visualisierung und Analyse von Ganganalyse-Zeitreihen zielte [2,3]. Diese Toolbox nutzt wegen der unterschiedlichen Problemstellungen und den inzwischen verfügbaren erweiterten Möglichkeiten in MATLAB teilweise abweichende Datenstrukturen im Vergleich zu KAFKA. Etwa ab 2004 wurde Gait-CAD zu einer universellen Toolbox zur Analyse von Zeitreihen umgestaltet und der überwiegende Teil der KAFKA-Funktionalität wurde in Gait-CAD integriert. Das Ziel bestand darin, in MATLAB die Auswertung und Visualisierung hochdimensionaler Datensätze komfortabel zu ermöglichen und so Einblicke in die strukturellen Eigenschaften von komplizierten Datensätzen zu erhalten. Hierbei wurde besonderer Wert auf offene Benutzerschnittstellen, die Anpassbarkeit an neue Anwendungsgebiete und eine komfortable Erweiterung der Funktionalität gelegt. Inzwischen wird Gait-CAD u.a. zum Entwurf von Brain-Machine-Interfaces [4], zur Auswertung von Nervensignalen [5] und von Handkraftmessungen [6] sowie für industrielle Vorstudien in der thermischen Abfallbehandlung eingesetzt.

Gait-CAD setzt MATLAB (getestet für Version 5.3 und 7.0) voraus und erfordert in vielen Fällen keine zusätzlichen MATLAB-Toolboxen. Einzelne Befehle greifen jedoch auf Standard-Toolboxen wie die Signal-Toolbox, die Statistik-Toolbox und die Wavelet-Toolbox zu. Die Entscheidung für eine MATLAB-basierte Lösung fiel, um die umfangreiche mathematische Funktionalität dieses Programmpakets der Fa. The Mathworks Inc. zu nutzen.

Gait-CAD ist eine freie Software (open source) und wird in der deutschsprachigen Version inkl. einer umfangreicheren Beschreibung [7] ab Ende November 2006 im Inter-

net unter <http://www.iai.fzk.de/biosignal> → Download zur Verfügung gestellt. Eine englische Version und zusätzliche Download-Varianten sind für 2007 geplant. Sie kann unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifiziert werden (siehe <http://www.fsf.org/>).

In dieser Arbeit wird zunächst Gait-CAD beschrieben (Abschnitt 2). Anschließend wird auf einige Besonderheiten bei der Programmierung und Dokumentation eingegangen (Abschnitt 3) und anhand einiger Anwendungsbeispiele die Breite der Einsatzmöglichkeiten diskutiert (Abschnitt 4).

2 Programmbeschreibung und Leistungsumfang

2.1 Arbeitsweise und Bedienphilosophie

Gait-CAD verfügt über eine grafische Oberfläche (GUI - Graphical User Interface) mit Menüeinträgen sowie Bedienelementen wie Auswahllisten, Checkboxen und Editierfeldern (Bild 1). Hierzu wurden die MATLAB-Standardfunktionen `uicontrol` und `uimenu` verwendet, die z.T. durch eigene Funktionen mit zusätzlicher Funktionalität gekapselt aufgerufen werden. Diese Elemente kommunizieren über Callback-Funktionen mit den Variablen im Workspace und rufen GUI-unabhängige Funktionen auf. Eine solche Arbeitsweise reduziert den Einarbeitungsaufwand für neue Nutzer und erleichtert bei Verwendung der später vorgestellten Makros die Automatisierung und Standardisierung von Auswertungsabläufen. Prinzipiell ist aber auch eine MATLAB-typische Arbeit mit dem Kommandofenster und den Variablen möglich.

2.2 Datensätze und Problemstellungen

Ausgangspunkt ist ein Lerndatensatz mit $n = 1, \dots, N$ Datentupeln, die jeweils

- s_z Zeitreihen (Matrizen $\mathbf{X}_{ZR}[n]$ mit $x_{ZR,r}[k,n]$, $r = 1, \dots, s_z$, $k = 1, \dots, K$ Abtastzeitpunkte),
- s (Einzel-) Merkmale (Vektoren $\mathbf{x}[n]$ mit $x_l[n]$, $l = 1, \dots, s$) und
- s_y Ausgangsgrößen (Vektor $\mathbf{y}[n]$ mit $y_j[n]$, $j = 1, \dots, s_y$)

enthalten. Eine Visualisierung zeigt Bild 2.

Hierbei sind ordinale, intervallskalierte und rationalskalierte Merkmale [8] zulässig. Ordinale Merkmale sind wertediskret und unterliegen einer Ordnungsrelation (z.B. Größe mit Werten wie sehr klein, klein, mittel und groß). Die Rangnummern solche Werte sagen aber noch nichts über die Semantik der Abstände aus. Wenn alle Abstände zwischen aufeinanderfolgenden Werten gleich sind, handelt es sich um intervallskalierte Merkmale (z.B. Temperatur in [°C]). Bei rationalskalierten Werten gibt es noch zusätzlich einen natürlichen Nullpunkt (z.B. Größe in [m], Temperatur in [K]).

Neben diesen Daten können weitere Informationen wie z.B. A-priori-Präferenzen für Merkmale, Kostenmatrizen für Fehlentscheidungen und Merkmale (z.B. für Kosten von Messungen) verarbeitet werden.

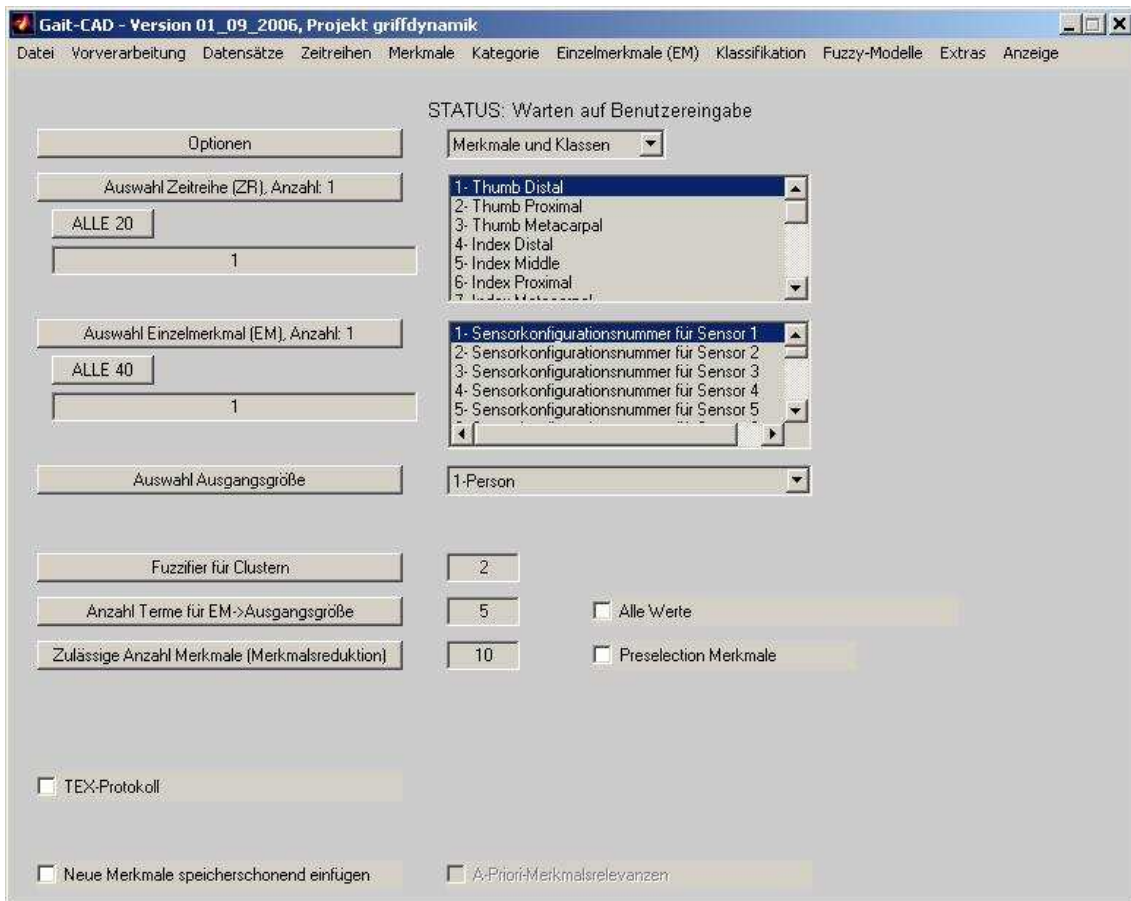


Bild 1: Bedienoberfläche von Gait-CAD

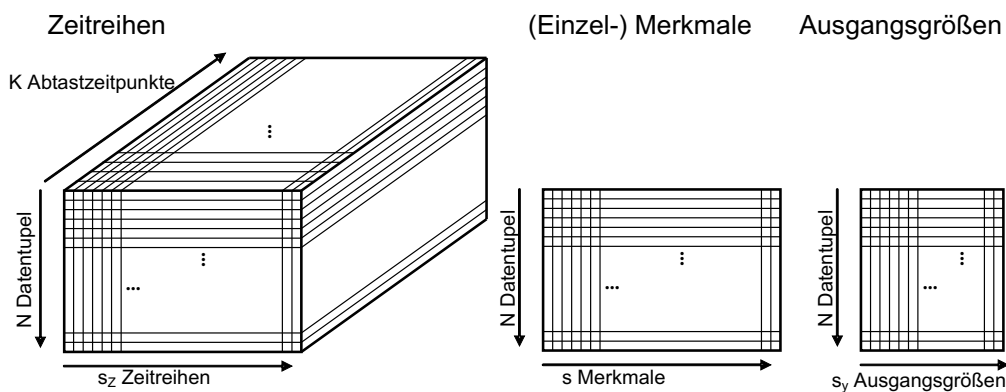


Bild 2: Veranschaulichung der Datenstrukturen in einem Lerndatensatz (nach [2])

Die Zielstellung besteht hauptsächlich in der Generierung von statischen bzw. quasi-statischen Schätzungen

$$\hat{y}_j[n + 1] = f(\mathbf{x}[n + 1]) \text{ bzw.} \quad (1)$$

$$\hat{y}_j[n + 1] = f(\mathbf{x}[n + 1](\mathbf{X}_{ZR}[n + 1])) \quad (2)$$

für ein Datentupel $n + 1$ mit unbekanntenen Ausgangsgrößen sowie in der Ermittlung von Zwischenergebnissen wie Listen von relevanten Merkmalen für bestimmte Problemstellungen usw. Die parallele Verwaltung mehrerer Ausgangsgrößen (z.B. Diagnosen bezüglich verschiedener Erkrankungen, Therapieentscheidungen, qualitative Bewertung von

Therapieerfolgen, Untersuchungszeitpunkt vor/nach einer Behandlung, Geschlecht, Altersgruppen usw. in der medizinischen Entscheidungsunterstützung) für ein Datentupel ermöglicht eine flexible Auswahl verschiedener Klassifikationsprobleme. Zudem können je nach Problemstellung Ein- und Ausgangsgrößen zumindest teilweise vertauscht werden.

Der Lerndatensatz wird durch eine binäre MATLAB-Projektdatei dargestellt, die Matrizen mit standardisierten Namen enthält (z.B. `d_orgs` für Zeitreihen, `d_org` für Einzelmerkmale und `code_all` für Ausgangsgrößen). Hinzu kommen optionale Matrizen bzw. Strukturen mit textuellen Bezeichnern und weiteren Informationen. Fehlende Informationen werden soweit möglich durch Standardwerte und -bezeichner ergänzt. Gait-CAD-Projektdateien können hierbei auf einem Computer mit 1 GB Hauptspeicher Größen von etwa 500 MB erreichen, ohne dass es zu Speicherproblemen kommt. Noch größere Projektdateien sind derzeit wegen der gewählten Datenstrukturen (Arrays statt Cell-Arrays) wegen MATLAB-internen Unterschieden in der Speicherverwaltung problematisch.

2.3 Leistungsumfang

Für ein verbal formuliertes Data-Mining-Problem muss zunächst eine geeignete Formalisierung gefunden werden. Diese Formalisierung beeinflusst dann sowohl die Zusammenstellung des Lerndatensatzes aus einer (externen) Datenbank (z.B. über spezielle Importwerkzeuge wie HeiDATAProViT im Bereich der Ganganalyse [9]) als auch die auszuwählenden Bewertungsmaße in Gait-CAD (Bild 3).

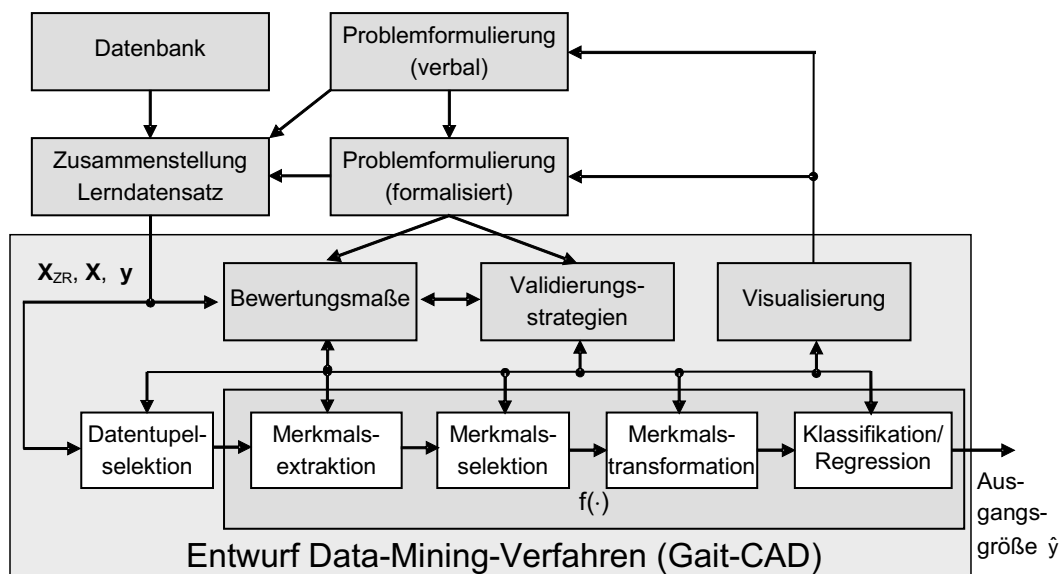


Bild 3: Einbindung von Gait-CAD in den Entwurfsprozess von Data-Mining-Verfahren

Zur Lösung von Data-Mining-Problemen ist eine Vielzahl bekannter Algorithmen zur

- Datentupelselektion (z.B. Ausreißerdetektion, Löschen unvollständiger Datentupel bzw. Merkmale, Auswählen von Teildatensätzen),
- Merkmalsextraktion (z.B. Spektrogramme, FFT-Analysen, Korrelationsanalysen, lineare Filter, Ermittlung von Extrema, Mittelwerten, Fuzzifizierung usw.),

- Merkmalsbewertung und -selektion (z.B. Multivariate Varianzanalyse, *t*-Test, informationstheoretische Maße, Regression),
- Merkmalstransformation (z.B. Diskriminanzanalyse, Hauptkomponentenanalyse, ICA - Independent Component Analysis),
- überwachten bzw. unüberwachten Klassifikation (z.B. Entscheidungsbäume, Cluster-Verfahren, Bayes-Klassifikatoren, Neuronale Netze, Nearest-Neighbour-Verfahren, Support-Vektor-Maschinen) sowie
- Validierung (z.B. Crossvalidierung, Bootstrap)

in Gait-CAD eingebunden. Dazu werden je nach Verfügbarkeit Matlab-Standardfunktionen, freie Matlab-Toolboxen (FastICA¹, SVM and Kernel Methods Matlab Toolbox [10]², SOM Toolbox [11]³, Ip_solve⁴) oder eigene Funktionen aufgerufen. Hierbei sind auch selbst entwickelte Algorithmen integriert:

- Entwurf von Fuzzy-Einzelregeln und -Regelbasen inkl. der Erzeugung natürlicher Erklärungs-texte [12],
- entscheidungstheoretische Ansätze zum Entwurf von Fuzzy-Regeln [13, 14],
- zeitreihenspezifische Algorithmen zur Merkmalsextraktion, zum Fuzzy-Clustering [2, 15] und zur zeitvarianten Klassifikation [16],
- A-priori-Relevanzen zur Berücksichtigung der Interpretierbarkeit [12] und
- eine Merkmalsaggregation und -auswahl mit einem modifizierten Wrapper-Ansatz [17].

Hinzu kommen umfangreiche Möglichkeiten zur komfortablen Auswahl von Datentupeln (z.B. über ihre Werte der Ausgangsgrößen), zur Visualisierung, zum Import und Export von Daten, zum Protokollieren von Ergebnissen in Text- und \LaTeX -Dateien, zur Codegenerierung für ausgewählte Klassifikatoren, zum Umbenennen usw.

Außerdem existieren ergänzende Funktionen, die zwar nicht direkt zur Toolbox gehören, die aber deren Erstellung, Wartung (z.B. durch Abhängigkeitsanalysen für Funktionen und Variablen) und Dokumentation wesentlich erleichtern (siehe auch Abschnitt 3).

2.4 Arbeiten mit Makros

Gait-CAD ermöglicht die Aufzeichnung von Bedienabläufen bei der Betätigung von Menü- und Bedienelementen als Makros. Diese Makros sind Textdateien und können so durch weitere MATLAB-Funktionen manuell ergänzt werden. Der Hauptvorteil besteht darin, auch komplexe Verarbeitungsketten verschiedener Operationen schnell und fehlerlos ausführen zu können. Diese Vorgehensweise ersetzt zumindest teilweise grafische Editoren zum Festlegen der Verarbeitungsreihenfolge (z.B. Clementine der Fa. SPSS⁵, InformationMiner [18]), die deutlich aufwändiger zu implementieren sind.

Über die Arbeit mit Makros ist es insbesondere möglich, universelle Funktionen für die Crossvalidierung und Bootstrap zu implementieren. Hier werden zwei verschiedene Makros für das Lernen und Testen angegeben. Die Validierungsfunktion übernimmt lediglich noch die Auswahl der Datentupel entsprechend dem aktuellen Lern- bzw. Testdatensatz.

¹<http://www.cis.hut.fi/projects/ica/fastica/>

²<http://asi.insa-rouen.fr/~arakotom/toolbox/index.html>

³<http://www.cis.hut.fi/projects/somtoolbox/>

⁴<http://lpsolve.sourceforge.net/5.5/>

⁵<http://www.spss.com/clementine/>

Makros können über Skripte auch auf mehrere Projektdateien automatisiert angewendet werden, was ein effektives Arbeiten für Probleme mit einem hohen Rechenaufwand ermöglicht. Beispiele hierfür sind Projekte mit vielen Datentupeln und Merkmalen, bei denen eine Validierung oft einige Stunden Rechenaufwand erfordert.

2.5 Arbeiten mit Plugins

Die Merkmalsextraktion aus Zeitreihen wird in Gait-CAD mit Hilfe von Plugins umgesetzt. Plugins sind speziell konfigurierbare und strukturierte Funktionen, die aus ganzen Zeitreihen oder bestimmten Abschnitten neue Zeitreihen oder Einzelmerkmale (z.B. Extrema usw.) extrahieren.

Bei der Extraktion von neuen Einzelmerkmalen aus Zeitreihen existiert beispielsweise immer ein Auswahlschema von

- einer ausgewählten Zeitreihe aus dem aktuellen Projekt,
- einem u.U. projektspezifischem Einzugsbereich (Datei *.einzug) in Abtastzeitpunkten für ein Segment der Zeitreihe $[k_{min}, \dots, k_{max}]$ sowie
- dem eigentlichen Algorithmus der Merkmalsextraktion (enthalten in Datei plugin_*.m) aus dem jeweiligen Segment.

Zur Merkmalsextraktion existieren dabei vielfältige vorgefertigte Plugins, das Erzeugen nutzerspezifischer Plugins ist aber ebenfalls möglich. Ein Beispiel für ein vorgefertigtes Plugin ist das Berechnen der Mittelwerte von Zeitreihensegmenten mit dem Plugin plugin_mean_em.m und dem Einzug $[k_{min} = 10, \dots, k_{max} = 100]$ gemäß

$$x_{Mean_{[10..100]}_r}[n] = \frac{1}{91} \sum_{k=10}^{100} x_{ZR,r}[k, n] \quad (3)$$

Die Auswahl mehrerer Zeitreihen, Einzugsbereiche und Merkmalsextraktoren bewirkt ein sequentielles Abarbeiten aller möglichen Kombinationen. Somit ist ein schnelles Erzeugen einer Vielzahl von Einzelmerkmalen möglich.

2.6 Installationsroutinen

Zur Installation wurde der frei verfügbare NSIS-Compiler⁶ eingesetzt, der zunächst alle benötigten Dateien in eine ausführbare Datei packt. Eine Installation erfordert die Zustimmung zu den Lizenzbedingungen und ermöglicht eine Definition der gewünschten Pfade auf dem Zielrechner. Anschließend wird die Toolbox in die gewünschten Pfade kopiert und die entsprechenden Suchpfade werden beim Start von Gait-CAD mit in den aktuellen MATLAB-Suchpfad eingetragen.

3 Strategie-Elemente bei der Softwareentwicklung

3.1 Überblick

Im Unterschied zu kommerziellen Projekten ist die Softwareentwicklung an Hochschulen und Forschungseinrichtungen oft durch viele wechselnde Entwickler mit unterschied-

⁶<http://nsis.sourceforge.net>

licher Programmiererfahrung, einen geringen gleichzeitigen Personaleinsatz, geringe finanzielle Mittel (kein Outsourcing) und über die Jahre wechselnde Zielstellungen (z.B. durch neue Projekte, Methoden und Anwendungsgebiete) verbunden. Daraus resultieren lange Entwicklungszyklen inkl. aller damit verbundenen Probleme (z.B. durch veränderte Programm-Versionen, historische und nahezu irreversible Entscheidungen über Daten- und Programmstrukturen). Diese Situation erschwert eine konsistente Vorgehensweise beim Software-Engineering. Wichtige Arbeitsschritte wie Problemanalysen inkl. der Erstellung eines Lastenhefts, eine formelle Systemspezifikation und -planung mit der Erstellung eines Pflichtenhefts sowie Systemtests (siehe z.B. [19]) werden gar nicht oder nur teilweise durchgeführt. Das führt unvermeidlich zu einer eingeschränkten Softwarequalität. Diese äußert sich u.a. in historisch gewachsenen und suboptimalen Datenstrukturen, einer mangelnden Bedienerführung, Dokumentation und Robustheit, nur teilweise beseitigten Fehlern, teilweise redundanten und widersprüchlichen Funktionen unter gleichem Namen usw.

Diese Probleme traten auch bei der Entwicklung von KAFKA und Gait-CAD auf. Allerdings konnte durch die konsequente Verwendung des unter GNU-GPL lizenzierten Versionierungstools WinCVS⁷ eine verbesserte Qualität erreicht werden. Hiermit war es möglich,

- immer eine aktuelle Fassungen auf einem Server zu führen (wichtig für eine komfortable Fehlerkorrektur, den Know-How-Erhalt und konsistente Installationen),
- alte Versionen für einzelne Dateien und die ganze Toolbox wieder zu rekonstruieren und
- Unterschiede zwischen Versionen anzuzeigen.

Eine parallele Überarbeitung durch mehrere Entwickler ist in den meisten Fällen problemlos möglich und wird durch ein automatisches Zusammenführen ("Mergen") der erzeugten Versionen realisiert. Eventuelle Konflikte (z.B. durch Bearbeiten der gleichen Programmzeile) werden automatisch erkannt und müssen manuell aufgelöst werden.

Weitere Maßnahmen für eine verbesserte Qualität ermöglichen u.a. das Erzeugen fremdsprachiger Versionen (Abschnitt 3.2) und eine modulare Dokumentation (Abschnitt 3.3).

3.2 Fremdsprachige Versionen

Da die Toolbox ursprünglich komplett in deutscher Sprache konzipiert war, wurden alle Sprachelemente in der GUI sowie Textausgaben im MATLAB-Kommandofenster und in Protokolldateien in Deutsch implementiert. Eine nachträgliche Umstellung auf eine variablenbasierte Lösung zur Realisierung beliebiger Sprachversionen war aus Aufwandsgründen nicht realistisch.

Deshalb wurde eine Übersetzungsroutine konzipiert, deren erste Fassung in [20] beschrieben ist und die seitdem als Basis für Weiterentwicklungen diente. Dabei wurde der Weg gewählt, vollautomatisch alle Strings aus allen Quelldateien zu extrahieren, in ein Wörterbuch (L^AT_EX-Datei) zu schreiben und dann jeden String entweder einmalig manuell zu übersetzen oder als nicht zu übersetzen zu kennzeichnen. Die Entscheidung für vollständige Strings fiel, um Grammatikprobleme durch eine wortweise Übersetzung zu vermeiden. Ein Beispiel lautet:

⁷<http://www.wincvs.org>

```
wort(354).en='%d Output variables will be deleted\n';
% wort(354).de='%d Ausgangsgrößen werden gelöscht\n'
```

Typische Beispiele für nicht zu übersetzte Strings sind stringförmige MATLAB-Konstrukte (z.B. "LineWidth"), standardisierte Dateinamen, eigene Funktions- und Variablennamen in eval-Konstrukten oder Quelltext-Bestandteile zum Export von C++-Dateien. Generell werden mehrfach auftretende Strings gleich behandelt. Im Beispiel finden sich dann 2405 zu übersetzende und 2513 zu ignorierende Strings (Stand 8.9.2006). Diese Extraktion muss immer dann wiederholt werden, wenn neue Strings hinzukommen. Neu hinzukommende Strings werden dann automatisch ergänzt und zunächst nicht übersetzt.

Nach dem Übersetzen mit dem Wörterbuch entsteht im Ergebnis eine Kopie der aktuellen deutschsprachigen Version, in der alle zu übersetzenden Strings ausgetauscht werden (vgl. Bild 4). Eine Umschaltung zur Laufzeit ist so aber nicht möglich. Strings in Projekten (z.B. Merkmalsbezeichner) werden ebenfalls nicht verändert. Bei der Implementierung ist lediglich darauf zu achten, dass unterschiedliche Stringlängen unproblematisch sind (z.B. bei Fenstergrößen oder Tests auf bestimmte Stringlängen) und keine Grammatikprobleme durch einen kleinteiligen Zusammenbau von Strings zu Sätzen entstehen. Beide Anforderungen erforderten aber nur geringfügige Anpassungsarbeiten.

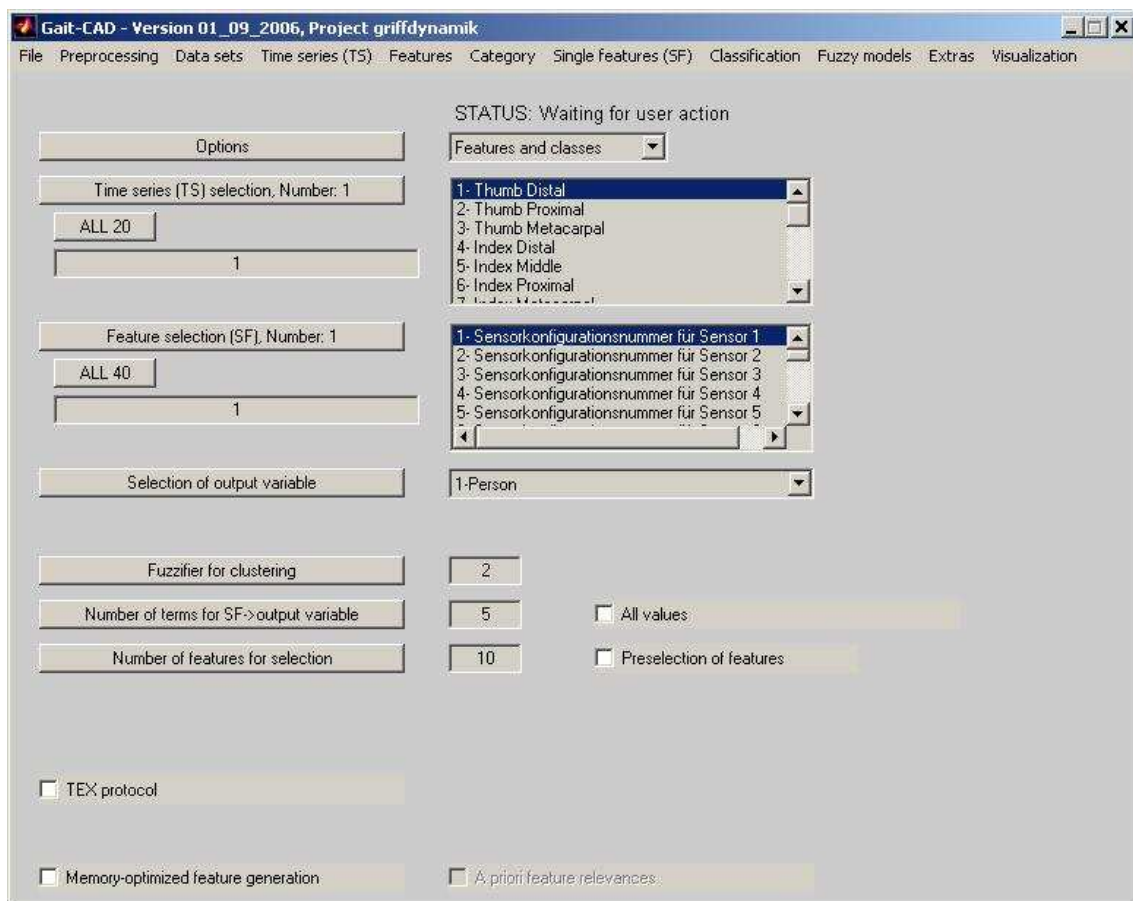


Bild 4: Englische Bedienoberfläche von Gait-CAD

Dieser Ablauf hat den Vorteil, dass eine Weiterentwicklung in Deutsch relativ problemlos möglich ist. Aus jedem beliebigen Entwicklungsstand kann vollautomatisch eine komp-

tible fremdsprachige Version erzeugt werden. Beim Erweitern sind lediglich neu eingeführte Strings einmalig zu bearbeiten.

Bislang wurde eine Übersetzung Deutsch → Englisch realisiert, weitere Sprachversionen des Programms sind prinzipiell mit einem Arbeitsaufwand einiger Personenwochen erzeugbar. Natürlich kommen noch entsprechende Aufwendungen für die Dokumentation hinzu, die sich aber auf das gleiche modulare Konzept stützt.

3.3 Modulare Dokumentation

Auf Grund der laufenden Weiterentwicklungen, der möglichen Änderung von Funktionen in Menüs und Eingabefeldern sowie der möglichen Definition projektspezifischer Funktionen in Gait-CAD besteht die Notwendigkeit, nur die bereits implementierten Teile einer Version in die Dokumentation einzubeziehen. Der manuelle Aufwand und das Risiko von Inkonsistenzen bei einer versionsspezifischen Anpassung sind allerdings sehr hoch. Deshalb wurde eine Variante gewählt, einen möglichst großen Teil der Dokumentation an die Menüeinträge und Bedienelemente zu koppeln und nur aktive Teile einzubinden [21]. Das Erkennen der aktiven Einträge und Elemente erfolgt anhand der jeweils gestarteten Gait-CAD-Version. Die zugehörigen Dokumentationen werden aus einer \LaTeX -Datei ausgelesen. Für eventuell noch fehlende Dokumentationsteile werden wiederum Einträge in dieser vorbereitet und entsprechende Warnungen generiert.

Außerdem werden die gegenseitigen Abhängigkeiten der Funktionen aus dem Quellcode extrahiert, um eine Liste der rufenden und gerufenen Funktionen zu generieren.

3.4 Modulare Erweiterbarkeit

Für das Einbinden neuer Funktionalitäten existieren zwei verschiedene Wege: Der einfachste Weg besteht darin, eigene Funktionen über Makros und Plugins zu integrieren. Leistungsfähiger ist eine Vollintegration in Gait-CAD, bei der eigene Funktionen durch zusätzliche Menüeinträge, die Erweiterung bestehender Bedienelemente um zusätzliche Optionen oder durch die Definition neuer Bedienelemente erfolgt. Durch die offen gelegten Quelltexte ist diese Vollintegration mit einem überschaubaren Aufwand möglich und kann entweder durch die Gait-CAD-Anwender selbst oder durch eine Kooperation von Gait-CAD-Anwendern und -Entwicklern erfolgen.

4 Anwendungserfahrungen

Inzwischen wird Gait-CAD u.a. zur Auswertung von Bewegungsanalysen [22], Nervensignalen [5] und von Handkraftmessungen [6] sowie zum Entwurf von Handprothesensteuerungen [17] und von Brain-Machine-Interfaces [4] eingesetzt. Außerdem wurde es für die Auswertung von Kfz-Benchmark-Daten [23] und für mehrere nicht publizierte industriellen Vorstudien auf dem Gebiet der thermischen Abfallbehandlung und der Auswertung verschiedener Sensordaten verwendet.

In allen bisher bearbeiteten Projekten erleichtert Gait-CAD insbesondere durch seine starke Orientierung auf die Visualisierung von Rohdaten, Zwischen- und Endergebnissen eine

interaktive Arbeitsweise. Somit gelingt oftmals ein Einblick in die strukturellen Eigenheiten der jeweiligen Aufgabenstellung, um Ursachen für unbefriedigende Ergebnisse zu verstehen und die Auswertung gezielt zu verbessern. Hervorzuheben sind Möglichkeiten zum Generieren und Bewerten neuer Merkmale, zum Auswählen einzelner Klassen, zum Vergleich verschiedener Klassifikatoren, zur Validierung usw. Typische Probleme wie z.B.

- zeitvariante Änderungen von Merkmalen,
- das Vorhandensein von Ausreißern,
- gravierende Abweichungen zu getroffenen Verteilungsannahmen (z.B. das Vorhandensein heterogener Subklassen innerhalb einer Klasse),
- die grundlegende Nichteignung bestimmter Algorithmen für die jeweilige Aufgabenklasse und
- die Nicht-Unterscheidbarkeit einzelner Klassen,

können so insbesondere in der Phase des Algorithmenentwurfs für eine Aufgabenklasse komfortabel analysiert werden. Deren Behandlung erfolgt dann über eine manuelle oder teilautomatische Selektion von Datentupeln und Merkmalen. Das Arbeiten mit Makros und Plugins erleichtert die Automatisierung von Abläufen und das einfache Einbinden anwendungsspezifischer Funktionalitäten.

5 Zusammenfassung und Ausblick

Dieser Beitrag beschreibt die freie MATLAB-Toolbox Gait-CAD und erläutert einige Strategieelemente bei der Entstehung sowie bisherige Anwendungserfahrungen. Das Ziel der Toolbox besteht darin, eine Plattform für die Data-Mining-Anwendungen und den umfangreichen Vergleich verschiedener Methoden bereitzustellen. Die Toolbox ist so konzipiert, das zukünftig weitere Algorithmen vergleichsweise einfach eingebunden werden können.

Danksagung:

Die Autoren danken an dieser Stelle vielen fleißigen Programmierern, Algorithmen-Entwicklern und Programm-Testern, insbesondere Sebastian Beck, Jens Jäkel, Lutz Gröll, Matthias Schablowski, Sebastian Braun, Mihai Lipovei, Sebastian Gollmer, Joachim Dieterle, Sebastian Wolf, Andreas Gommlich und Dimitrios Patikas. Die Förderung durch die Deutsche Forschungsgemeinschaft (DFG) im Rahmen des Projektes 'Diagnoseunterstützung in der Ganganalyse' trug wesentlich dazu bei, den ganganalytischen Teil aufzubauen und somit eine Basis für die Weiterentwicklung der Toolbox zu schaffen.

Literatur

- [1] Mikut, R.; Peter, N.; Malberg, H.; Jäkel, J.; Gröll, L.; Bretthauer, G.; Abel, R.; Döderlein, L.; Rupp, R.; Schablowski, M.; Gerner, H.: *Diagnoseunterstützung für die instrumentelle Ganganalyse (Projekt GANDI)*. Forschungszentrum Karlsruhe (FZKA 6613). 2001.
- [2] Loose, T.: *Konzept für eine modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse*. Dissertation, Universität Karlsruhe, Universitätsverlag Karlsruhe. 2004.
- [3] Loose, T.; Mikut, R.; Rupp, R.; Schablowski, M.: Gait-CAD - A Matlab Toolbox for Application of Data Mining Methods in Gait Analysis. *Gait & Posture* 18(E2) (2003), S. 114–115.
- [4] Burmeister, O.; Reischl, M.; Mikut, R.: Zeitvariante Klassifikatoren für Brain Machine Interfaces und Neuroprothesen. In: *Proc., 15. Workshop Computational Intelligence*, S. 129–143. Universitätsverlag Karlsruhe. 2005.

- [5] Krüger, T.; Reischl, M.; Lago, N.; Burmeister, O.; Mikut, R.; Ruff, R.; Hoffmann, K.-P.; Navarro, X.; Stieglitz, T.: Analysis of Microelectrode-Signals in the Peripheral Nervous System, In-Vivo and Post-Processing. In: *Proc., Mikrosystemtechnik Kongress Deutschland*, S. 69–72. Freiburg: VDE-Verlag. 2005.
- [6] Pylatiuk, C.; Reischl, M.; Mikut, R.; Kargov, A.; Bretthauer, G.: Determination of Stability in Multi-Contact Grasping. *Biomedizinische Technik* (2006). Akzeptierter Beitrag.
- [7] Mikut, R.; Loose, T.; Burmeister, O.; Braun, S.; Reischl, M.: Dokumentation der der MATLAB-Toolbox Gait-CAD. Techn. Ber., Forschungszentrum Karlsruhe GmbH. 2006.
- [8] Backhaus, K.; Erichson, B.; Plinke, W.; Weiber, R.: *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. Berlin u.a.: Springer. 2000.
- [9] Schablowski, M.; Schweidler, J.; Rupp, R.: HeiDATAProVIT-Heidelberg Data Archiving, Tag Assembling, Processing and Visualization Tool. *Computer Methods and Programs in Biomedicine* 73 (2004), S. 61–70.
- [10] Canu, S.; Grandvalet, Y.; Rakotomamonjy, A.: SVM and Kernel Methods Matlab Toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France. 2003.
- [11] Vesanto, J.; Himberg, J.; Alhoniemi, E.; Parhankangas, J.: SOM Toolbox for MATLAB. Techn. Ber., Helsinki University of Technology. 2000.
- [12] Mikut, R.; Jäkel, J.; Gröll, L.: Interpretability Issues in Data-Based Learning of Fuzzy Systems. *Fuzzy Sets and Systems* 150(2) (2005), S. 179–197.
- [13] Beck, S.; Mikut, R.; Jäkel, J.: A Cost-Sensitive Learning Algorithm for Fuzzy Rule-Based Classifiers. *Mathware and Soft Computing* 11(2-3) (2004), S. 175–195.
- [14] Beck, S.: *Ein Beitrag zum automatischen Entwurf von Fuzzy-Entscheidungs-systemen bei unvollständiger Information*. Dissertation, Universität Karlsruhe, Universitätsverlag Karlsruhe. 2005.
- [15] Loose, T.; Dieterle, J.; Mikut, R.; Rupp, R.; Abel, R.; Schablowski, M.; Bretthauer, G.; Gerner, H. J.: Automatisierte Interpretation von Zeitreihen am Beispiel von klinischen Bewegungsanalysen. *at - Automatisierungstechnik* 52 (2004), S. 359–369.
- [16] Burmeister, O.; Reischl, M.; Gröll, L.; Mikut, R.: Zeitvariante Klassifikatoren zur Steuerung von Brain Machine Interfaces und Neuroprothesen. *at - Automatisierungstechnik* 54(11) (2006). Im Druck.
- [17] Reischl, M.: *Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen*. Dissertation, Universität Karlsruhe, Universitätsverlag Karlsruhe. 2006.
- [18] Rügheimer, F.; Kruse, R.: Datenanalyse-Plattform InformationMiner. In: *Proc., 15. Workshop Computational Intelligence*, S. 117–128. Universitätsverlag Karlsruhe. 2005.
- [19] Pomberger, G.; Pree, W.: *Software Engineering. Architektur-Design und Prozessorientierung*. Hanser. 2004.
- [20] Gollmer, S.: Softwareentwicklung für medizinische Anwendungen am Beispiel myoelektrischer Handprothesen. Techn. Ber., Bericht zum Praxissemester, Forschungszentrum Karlsruhe und FH Koblenz, Standort Remagen. 2005.
- [21] Braun, S.: Automatisierte Dokumentation der Toolboxen KAFKA und Gait-CAD für Matlab. Techn. Ber., Forschungszentrum Karlsruhe. 2006.
- [22] Wolf, S.; Loose, T.; Schablowski, M.; Döderlein, L.; Rupp, R.; Gerner, H. J.; Bretthauer, G.; Mikut, R.: Automated Feature Assessment in Instrumented Gait Analysis. *Gait & Posture* 23(3) (2006), S. 331–338.
- [23] Mikut, R.: Fuzzy-Modellbildung für den Benchmarkdatensatz Kfz-Aggregate. In: *Proc., 12. Workshop Fuzzy Systeme*, S. 127–134. Forschungszentrum Karlsruhe (FZKA 6767). 2002.

Vergleich von klassischem und temporalem Fuzzy-Regler beschrieben in Fuzzy Control Language mit PID-Reglern

Thorsten W. SCHMIDT, Dominik HENRICH

Universität Bayreuth
Fakultät für Mathematik, Physik und Informatik
Lehrstuhl Angewandte Informatik III
D-95440 Bayreuth
E-Mail: {thorsten.w.schmidt, dominik.henrich}@uni-bayreuth.de
[Http://ai3.inf.uni-bayreuth.de](http://ai3.inf.uni-bayreuth.de)

1. Einleitung

1.1. Motivation

Seit vielen Jahren beschäftigt sich die Forschung mit Fuzzy-Logik und ihrer Anwendung in Fuzzy-Reglern. Seit einigen Jahren gibt es auch industriell eingesetzte Fuzzy-Regler. Diese werden zum Beispiel in Waschmaschinen oder anderen Geräten des häuslichen Gebrauchs verwendet. Die Vorteile der Fuzzy-Logik sind, dass vorhandenes Wissen über ein Regelungsprozess sehr leicht zur Modellierung eines Reglers verwendet werden kann und im weiteren Verlauf der Entwicklung und Verbesserung des Reglers dieses Wissen immer transparent bleibt und somit nicht verloren geht. Durch die klare Lesbarkeit der Fuzzy-Logik bleibt das Wissen an sich wartbar. Jedoch können diese Fuzzy-Regler nicht als Wartungssystem eingesetzt werden, da sie nicht in der Lage sind zeitliche Abhängigkeiten von Ereignissen untereinander oder überhaupt Zeit zu modellieren, wie dies beispielsweise beim Modell Checking möglich ist, welches Temporal-Logik verwendet [Karjoth87]. Aus diesem Grund ist eine Erweiterung der Fuzzy-Logik um zeitliche Aspekte nötig. In [Schmidt04] wird eine solche Erweiterung an den Prädikaten vorgenommen. Die Prädikate werden so erweitert, dass zum einen die so genannte temporale Fuzzy-Logik genauso mächtig ist wie die Temporal-Logik und zum anderen eine solide mathematische Basis geschaffen wird, so dass die Bedingungen, welche an Fuzzy-Prädikate und Zugehörigkeitsfunktionen gestellt werden auch erfüllt sind. Die Vereinigung der Temporal-Logik und Fuzzy-Logik zur *temporalen Fuzzy-Logik* mit einer Beschreibung der Zeit als Fuzzy-Zeit-Terme ist in [Schmidt05] gezeigt. Die Fuzzy-Zeit-Terme sind dabei nicht mit den Fuzzy-Zeit-Objekten von [Bovenkamp97] zu verwechseln.

Hier soll nun eine Untersuchung eines temporalen Fuzzy-Reglers im Vergleich mit PID-Reglern und klassischen Fuzzy-Reglern die Vorteile der temporalen Fuzzy-Logik zeigen. In [Giron02] wird eine Untersuchung der Stabilität von Fuzzy-Reglern auf hohem Abstraktionslevel durchgeführt. Dabei wird eine Literaturliste von annähernd 200 Veröffentlichungen angeführt, aber konkrete Ergebnisse lassen sich daraus nicht ableiten. Da wir aber unseren eigens entwickelten temporalen Fuzzy-Regler mit anderen Reglern vergleichen, führen wir diese Untersuchung selbst am Beispiel eines inversen Pendels durch.

1.2. Abgrenzung

Als Endziel, welches über die temporalen Fuzzy-Prädikate und Fuzzy-Zeit-Terme hinaus geht, möchten wir ein Wartungssystem, wie oben beschrieben entwickeln. Das Wartungssystem generiert Wartungsaufgaben für einen zu wartenden Prozess. Die Wartungsaufgaben werden dabei durch zeiterweiterte Fuzzy-Logik Regeln beschrieben. Natürlich kann das Wartungssystem auch als Überwachungssystem oder Regelungssystem verwendet werden. Das Hauptaugenmerk liegt jedoch auf dem Wartungssystem, welches Wartungsaufträge generiert. Die automatisch generierten Wartungsaufträge sind zeitlich so geplant, dass eine möglichst ökonomische Abarbeitung ohne größere Standzeiten des Prozesses möglich ist. Hierzu vergleichen

wir einen temporalen Fuzzy-Regler mit einem klassischen Fuzzy-Regler und verschiedenen PID-Reglern.

2. Temporaler Fuzzy-Regler

Der temporale Fuzzy-Regler nutzt zur Darstellung seiner Regeln und Daten eine Erweiterung der industriell eingesetzten Fuzzy-Regler-Sprache FCL (= Fuzzy Control Language) um temporale Aspekte sprachlich zu modellieren. Durch diese Erweiterung ist es möglich weitaus komplexere Regler zu schreiben, wodurch diese Regler ein breiteres Anwendungsgebiet haben als gewöhnliche Fuzzy-Regler.

2.1. Einordnung des temporalen Fuzzy-Reglers

Ein temporaler Fuzzy-Regler unterscheidet sich grundsätzlich von einem normalen Fuzzy-Regler. Zum einen in seinem Verhalten und dann auch in seinem Aufbau. Gehen wir davon aus, dass beide Regler zeitliche Aspekte behandeln sollen, dann benötigt der klassische Fuzzy-Regler die Zeit als Eingabevariable. Diese benötigt man aber bei einem temporalen Fuzzy-Regler nicht, denn dieser kennt die aktuelle Zeit und weiß somit, zu welchem Zeitpunkt ein-treffende Daten aufgenommen wurden.

Die Schnittstellen des temporalen Fuzzy-Reglers kann man in zwei Klassen unterteilen. Zum einen die direkten Schnittstellen. Diese beinhalten den Input beziehungsweise Output, welche in das beziehungsweise aus dem System fließen. Diese Schnittstellen gibt es bei klassischen und temporalen Fuzzy-Reglern. Nur enthält der Input bei letzterem keine Zeit. Zum anderen gibt es die indirekten Schnittstellen. Dies sind die Schnittstellen zu anderen Komponenten, zu welchen ein der Prozess keine direkte Verbindung oder genauer gesagt keinen direkten Einfluss hat. Dies sind die Datenbank und das Orakel, welche in einem klassischen Fuzzy Regler nicht vorhanden sind (siehe Abbildung 1).

Das Informationspaar von Datenwert und Aufnahmezeitpunkt speichert der temporale Fuzzy-Regler in der Datenbank ab. Diese Datenbank bildet eine Historie auf welche immer zugegriffen werden kann. Im optimalen Fall treffen die Daten in äquidistanten Zeitabständen ein, so dass Zeitreihen mit gleichen Zeitabständen von benachbarten Daten vorliegen hat. Sind die Zeitabstände nicht äquidistant, so können die Daten immer noch interpoliert werden. Dies ist ein struktureller Unterschied zu einem klassischen Fuzzy Regler, der diese Datenbank nicht besitzt und auch nur auf aktuell in das System eingespeiste Daten zugreifen kann: „*Temperatur IS low*“. Der temporale Fuzzy-Regler dagegen kann zum Beispiel durch den Ausdruck „*Temperatur IS_{TIME} gestern low*“ auf vergangene Werte zugreifen.

Ein weiterer struktureller unterschied ist das in Abbildung 1 dargestellte Orakel. Es soll verdeutlichen, dass es nicht einfach ist in die Zukunft zu sehen. Hier werden zur Vorhersage der zukünftigen Signalverläufe zum Beispiel lineare Vorhersagen oder andere einfache mathematische Extrapolationsmethoden eingesetzt. Sofern ein Modell des zu regelnden Prozesses bekannt ist, können auch die Modellannahmen beziehungsweise die Gleichungen die den Prozess beschreiben eingesetzt werden, um eine höhere Vorhersagegenauigkeit zu erreichen.

Ein Vergleich eines temporalen Fuzzy-Reglers mit einem klassischen Fuzzy-Regler gibt es in Kapitel 3. Außerdem werden die beiden Fuzzy-Regler mit optimierten PID-Regler verglichen.

2.2. Abkürzungen und Definitionen

Die Erweiterte Backus-Naur-Form (EBNF), ist eine Erweiterung der Backus-Naur-Form (BNF). Ursprünglich wurde sie von Niklaus Wirth zur Darstellung der Syntax der Programmiersprache Pascal eingeführt. Sie ist eine formale Metasprache, welche benutzt wird um kontextfreie Grammatiken darzustellen. Definiert ist sie durch die Norm ISO/IEC 14977 (final draft version SC22/N2249) in [IEC96].

Um eine einfachere und übersichtlichere Schreibweise zu erhalten definieren wir, dass Wörter, welche komplett in Großbuchstaben geschrieben sind immer Terminal-Symbole sind.

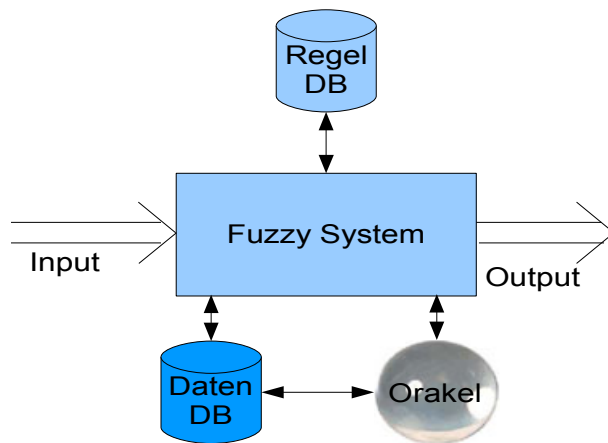


Abbildung 1: Temporaler Fuzzy Regler mit direkten (Input, Output) und indirekten Schnittstellen (Datenbank, Orakel).

Bei den meisten Erweiterungen handelt es sich um zielgerichtete Erweiterungen, welche nötig sind um zeitliche Aspekte zu beschreiben. Manche Änderungen werden jedoch nur vorgenommen, um die Arbeit mit der Regelungssprache zu vereinfachen.

2.3. Fuzzy Control Language

Die Fuzzy Control Language definiert und standardisiert von der International Technical Electrical Commission (IEC) in [IEC97] beschreibt einen Fuzzy-Regler hinsichtlich seiner Eingabe und Ausgabe mittels Fuzzy-Termen und seinem Verhalten mittels Regeln. Dieses Kapitel beschreibt kurz die Fuzzy Control Language so, wie sie von der IEC standardisiert ist.

Die Sprache wird zur Beschreibung von programmierbaren Reglern verwendet. Diese Regler werden programmiert, indem sie die Fuzzy-Terme zur Fuzzifizierung und Defuzzifizierung mitgeteilt bekommen. Ebenso werden die Regeln im Speicher des Reglers abgelegt. Alle Schritte von der Fuzzifizierung bis Defuzzifizierung können vom Regler selbst in Echtzeit mit einer Reaktionszeit von wenigen Millisekunden ausgeführt werden. Beschränkungen gibt es nur in der Anzahl der Regeln und Terme die maximal definiert werden können.

Die Fuzzy Control Language wird mit der Syntaxbeschreibungssprache Enhanced Backus-Naur Form (ENBF) definiert. Die Syntaxbeschreibungssprache EBNF ist in [IEC96] beschrieben. Die wichtigsten und im Folgenden verwendeten Definitionen zum EBNF Syntax sind kurz in Tabelle 1 beschrieben.

EBNF-Term	Beschreibung
$\{A\}$	Der Ausdruck A kann beliebig oft wiederholt oder ausgelassen werden.
$\{A\}$ -	Der Ausdruck A kann beliebig oft wiederholt werden. Mindestens jedoch ein mal.
$[A]$	Optionale Auswahl von A . Es kann gewählt oder weggelassen werden.
$A B$	Eine Auswahl von entweder A oder B . Es muss genau eine Möglichkeit ausgewählt werden.
GROSSSCHREIBWEISE	Terminalsymbole, welche durch EBNF-Regeln nicht weiter ersetzt werden.
GemischteSchreibweise	Nicht-Terminalsymbole, welche durch EBNF-Regeln solange weiter ersetzt werden, bis nur noch Terminalsymbole vorhanden sind.

Tabelle 1: Beschreibung der wichtigsten EBNF Regeln.

Im Folgenden dargestellt ist der Aufbau eines FCL Programmes wie im Standard von [IEC97] definiert. Dies kann als Programmgerüst angesehen werden, welches noch um Fuzzy-Terme, Variablen und Regeln gefüllt werden muss.

Dagegen sind alle anderen Wörter Nicht-Terminalsymbole. Für diese Wörter existiert eine Ersetzungsregel, durch welche eine neue Folge von Terminal und/oder Nicht-Terminalsymbolen entsteht.

Wird ein Ausdruck, ein Symbol oder ähnliches grau dargestellt, so ist dieser Ausdruck oder dieses Symbol zwar in dem betreffenden Standard definiert, von uns aber nicht verwendet. Es handelt sich dabei um Funktionalität, welche für unsere Zwecke unnötig ist.

Werden Ausdrücke oder Symbole bei der Beschreibung eines Standards fett dargestellt, so handelt es sich um Neuerungen, welche wir einführen um diesen Standard zu erweitern.

```

1  FUNCTION_BLOCK                20      {TERM FuzzyTermName := Points;}-
2                                21      [METHOD: DefuzzificationMethod;]
3  {STRUCT StructName           22      [DEFAULT := Real | NC;]
4  {StructDefinition;}-         23      [RANGE := (Min .. Max)]
5  END_STRUCT}                 24      END_DEFUZZIFY}
6                                25
7  {VAR_INPUT                   26      {RULEBLOCK RuleBlockName
8  {InputVariableName: DataType;}- 27      [AND:AndAlgorithm;]
9  END_VAR}                    28      [OR:OrAlgorithm;]
10                               29      [ACCU:AccumulationMethod;]
11  {VAR_OUTPUT                 30      [ACT:ActivationMethod;]
12  {OutputVariableName: REAL;}-   31      {RULE Integer: IF Condition THEN Conclu-
13  END_VAR}                    32      sion [WITH WeightingFactor;]-
14                               33      END_RULEBLOCK}
15  {FUZZIFY InputVariableName    34      {OPTION
16  {TERM FuzzyTermName := Points ;}- 35      UserDefinedOptions
17  END_FUZZIFY}                36      END_OPTION}
18                               37
19  {DEFUZZIFY OutputVariableName 38      END_FUNCTION_BLOCK

```

Das Programmgerüst enthält noch Nicht-Terminalsymbole. Diese werden solange durch die im Folgenden dargestellten Regeln ersetzt, bis nur noch Terminalsymbole vorhanden sind. Dann liegt ein syntaktischgültiges und korrektes FCL Programm vor. Besondere Nicht-Terminalsymbole sind *String*, *Integer* und *Real*, welche für eine Zeichenkette, eine Ganzzahl und eine Gleitkommazahl stehen. Diese sind nicht in Tabelle 2 angegeben.

Nicht-Terminalsymbol	EBNF Ersetzungsregel	Beschreibung des Nicht-Terminalsymbols
<i>StructName</i>	<code>::= String</code>	Gibt den Namen eines neuen Datentypes an. Dies wird nicht von uns benutzt.
<i>StructDefinition</i>	<code>::= String</code>	Definiert neue Datentypen. Da diese Möglichkeit von uns nicht Benutzt wird, beschreiben wir sie auch nicht genauer.
<i>DataType</i>	<code>::= REAL INT Struct-Name</code>	Drei Verschiedene Datentypen sind möglich. REAL für Gleitkommazahlen, INT für Ganzzahlen und selbstdefinierte Structs.
<i>InputVariableName</i>	<code>::= String</code>	Inputvariablen für das Fuzzysystem deren Wert fuzzyfiziert wird.
<i>OutputVariableName</i>	<code>::= String</code>	Der Wert von Outputvariablen wird durch Regelaktivierungen gesetzt.
<i>Min</i>	<code>::= Real</code>	Untere Intervallgrenze von RANGE
<i>Max</i>	<code>::= Real</code>	Obere Intervallgrenze von RANGE
<i>FuzzyTermName</i>	<code>::= String</code>	Name eines Fuzzy Terms
<i>Points</i>	<code>::= Real {(Real, Real)}</code>	Stützpunkte einer Zugehörigkeitsfunktionen. Jeder Fuzzy Term wird durch eine Zugehörigkeitsfunktion beschrieben, welche entweder ein Singleton (einzelner Punkt) oder ein Polygonzug (mindestens zwei Punkte) ist.
<i>DefuzzificationMethod</i>	<code>::= COG COGS COA LM RM</code>	Die Art, wie aus unscharfen Fuzzy-Werten scharfe Werte berechnet werden sollen. Am gebräuchlichsten ist die Center of Gravity Methode (COG=Schwerpunktmethode) oder COGS (COG für Singletons). COA (Center of Area) ist nur ein Synonym für COG. LM beziehungsweise LR stehen für die left beziehungsweise right max Methode.
<i>RuleBlockName</i>	<code>::= String</code>	Eindeutiger Name eines Regelblockes. Es können beliebig viele Regelblöcke existieren. In einem Regelungsschritt können die aktuell zu aktivierenden Regelblöcke über ihren Namen angegeben werden.
<i>AndAlgorithm</i>	<code>::= MIN PROD BDIF</code>	Am gebräuchlichsten ist die Min-Methode für AND Verknüpfungen. Weitere Methoden sind das Produkt PROD oder die beschränkte Differenz BDIF. MIN: $\min(\mu_a(x), \mu_b(x))$ PROD: $\mu_a(x)\mu_b(x)$ BDIF: $\max(0, \mu_a(x) + \mu_b(x) - 1)$
<i>OrAlgorithm</i>	<code>::= MAX ASUM BSUM</code>	Am gebräuchlichsten ist die Max-Methode für OR Verknüpfungen. Weitere Methoden sind die algebraische Summe ASUM und die beschränkte Summe BSUM. MAX: $\max(\mu_a(x), \mu_b(x))$ ASUM: $\mu_a(x) + \mu_b(x) - \mu_a(x)\mu_b(x)$ BSUM: $\min(1, \mu_a(x) + \mu_b(x))$

<i>AccumulationMethod</i>	:= MAX BSUM NSUM	Am gebräuchlichsten ist die Max-Methode für die Akkumulierung. Weitere Methoden sind die beschränkte BSUM und die normalisierte Summe NSUM. MAX: $\max(\mu_a(x), \mu_b(x))$ BSUM: $\min(1, \mu_a(x) + \mu_b(x))$ NSUM: $\frac{\mu_a(x) + \mu_b(x)}{\max(1, \max_{x'}(\mu_a(x') + \mu_b(x'))))}$
<i>ActivationMethod</i>	:= MIN PROD	Am gebräuchlichsten ist die Min-Methode für die Aktivierung. Eine weitere Methode ist das Produkt PROD. MIN: $\min(\mu_{Rule}, \mu_f(x))$ PROD: $\mu_{Rule} \mu_f(x)$
<i>Condition</i>	:= <[NOT] AtomIn> <[NOT] SystemCheck> <[NOT] (Condition {<AND OR> Condition}-)>	Eine beliebige Verschachtelungstiefe ist für die Bedingungen einer Regel möglich.
<i>Conclusion</i>	:= (AtomOut) < Conclusion, Conclusion >	Eine beliebige Anzahl von Folgerungen sind pro Regel möglich.
<i>AtomIn</i>	:= (InputVariableName IS FuzzyTermName)	Atom in einer Bedingung ist eine Fuzzy-Bedingung.
<i>AtomOut</i>	:= (OutputVariableName IS FuzzyTermName)	Atom in einer Folgerung ist eine Fuzzy-Bedingung.
<i>WeightingFactor</i>	:= Real	Gewichtungsfaktor einer Regel. Wird kein Gewicht angegeben, so wird das Gewicht als 1.0 angenommen.
<i>UserDefinedOptions</i>	:= String	Hier können beliebige vom Benutzer zu definierende Optionen in freiem Format angegeben werden. Diese müssen auch von Benutzer selbst ausgewertet werden. Diese Möglichkeit wird nicht von uns genutzt.

Tabelle 2: Erläuterungen zu Nicht-Terminalsymbolen der EBNF-Beschreibung der Fuzzy Control Language.

2.4. Temporal Fuzzy Control Language

Dieses Kapitel beschäftigt sich mit der zeitlichen Erweiterung der Fuzzy Control Language, der so genannten Temporal Fuzzy Control Language. Diese Erweiterung ist nötig, da sich, wie in Kapitel 2.2 beschrieben, die zugrunde liegende Fuzzy-Logik Sprache geändert hat. Die temporale Fuzzy-Logik hat mehr Prädikate und mehr Formulierungsmöglichkeiten (zum Beispiel die Angabe von Zeit in Bedingungen oder die Deklaration von Fuzzy-Zeit-Termen) wie die klassische Fuzzy-Logik. Diese Erweiterungen werden im Folgenden genauer vorgestellt.

Wie schon in Kapitel 2.3 eingeführt, kann man auch die Temporal Fuzzy Control Language in EBNF beschreiben. Schon im Programmgerüst gibt es Unterschiede zur Fuzzy Control Language. Diese sind, wie in folgender Legende erklärt, hervorgehoben:

- **Grau:** In FCL definiert, aber von TFCL nicht unterstützt. Dies betrifft nur die Definition von eigenen Datenstrukturen (Structs), Ganzzahlen als Datentypen und die Angabe von eigenen Optionen. Mit anderen Worten, in FTCL gibt es nur REAL als möglichen Datentyp.
- **Normal:** In FCL und TFCL gleichermaßen definiert.
- **Fett:** In FCL nicht vorgesehen, aber in TFCL definiert.

Gegeben ist nun das Programmgerüst, wie es für die Temporal Fuzzy Control Language verwendet wird.

```

1  FUNCTION_BLOCK                13  END_VAR}
2                                14
3  {STRUCT StructName           15  {VAR_OUTPUT
4  {InputVariableName: DataType}- 16  {OutputVariableName [actuator]: REAL;}-
5  END_STRUCT}                 17  END_VAR}
6                                18
7  {VAR_SYSTEM                   19  {VAR_EVENT
8  {SystemVariableName [actuator]: 20  {EventName;}-
9  DataType;}-                 21  END_VAR_EVENT}
10  END_VAR}                    22
11  {VAR_INPUT                   23  {FUZZIFY InputVariableNameStar
12  {InputVariableName: DataType;}- 24  {TERM FuzzyTermName := Points ;}-
25  [RANGE := (Min .. Max)]

```

```

26   END_FUZZIFY}
27
28   {DEFUZZIFY OutputVariableName
29     {TERM FuzzyTermName := Points;}-
30     [METHOD: DefuzzificationMethod;]
31     [DEFAULT := Real | NC;]
32     [RANGE := (Min .. Max)]
33   END_DEFUZZIFY}
34
35   FUZZY_TIME_TERM FuzzyTimeTermName
36     {FACT Integer := Poin;}-
37     {TIME Integer := Poin;}-
38   END_FUZZY_TIME_TERM
39
40   {EVENT EventName
41     {TASK TaskName := TaskNumber;}-
42   END_DEFUZZIFY}
43
44   {RULEBLOCK RuleBlockName
45     [AND:AndAlgorithm;]
46     [OR:OrAlgorithm;]
47     [ACCU:AccumulationMethod;]
48     [ACT:ActivationMethod;]
49     [PREDICTION:PredictionMethod;]
50     {RULE Integer: IF Condition THEN Conclu-
51       sion [WITH WeightingFactor;]}-
52   END_RULEBLOCK}
53
54   {OPTION
55     UserDefinedOptions
56   END_OPTION}
57
58   END_FUNCTION_BLOCK

```

Bei den Erläuterungen zu den Nicht-Terminalsymbolen in Tabelle 3 werden nur Änderungen und Neuerungen zu Tabelle 2 gezeigt. Außerdem verfügen nun auch Eingabevariablen über die Angabe eines Gültigkeitsbereiches RANGE für die eingehenden Daten. Diese Angabe wird benötigt, um den Bereich zur Anzeige von Daten festzulegen.

<i>Datenfeld</i>	<i>EBNF Notation</i>	<i>Beschreibung</i>
<i>StructName</i>	<i>:= String</i>	Nicht unterstützt
<i>DataType</i>	<i>:= REAL INT StructName</i>	Nur Realvariablen (Doublegenauigkeit) werden unterstützt.
<i>EventName</i>	<i>:= String</i>	Name eines Ereignisses, welches beim Auftreten an einen Benutzer gegeben werden kann.
<i>SystemVariableName</i>	<i>:= String</i>	Systemvariablen sind Realvariablen, welche nur innerhalb des Fuzzysystems verwendet werden. Sie werden nicht fuzzifiziert und auch nicht defuzzifiziert. Durch Angabe des Schlüsselwortes actuator wird die Variable zur Steuerung verwendet. Ansonsten ist es nur eine interne Variable.
<i>InputVarName</i>	<i>:= String InputVarNameStar</i>	Inputvariablen für das Fuzzysystem deren Wert fuzzifiziert wird.
<i>InputVariableNameStar</i>	<i>:= String InputVarName</i>	Der Name der Variablen darf mit dem Zeichen "*" enden, wenn der Stern durch eine Zeichenkette ersetzt werden kann, so dass der resultierende Name einer unter VAR_INPUT definierten Variablen entspricht. Die Definition der TERME betrifft alle Variablen auf, welche <i>InputVariableNameStar</i> expandiert werden kann.
<i>OutputVariableName</i>	<i>:= String</i>	Der Wert von Outputvariablen wird durch Regelaktivierungen gesetzt. Durch Angabe des Schlüsselwortes actuator wird die Variable zur Steuerung verwendet. Ansonsten werden sie nicht nach Außen gegeben
<i>FuzzyTimeTermName</i>	<i>:= String</i>	Name eines Fuzzy-Zeit-Terms. Der Term gibt an, aus welchem unscharfen Zeitbereich ein Prädikat Daten verarbeitet.
<i>DefuzzificationMethod</i>	<i>:= COG COGS COA <MCOG := Real> MM LM RM</i>	Die Art, wie aus unscharfen Fuzzy Werten scharfe Werte berechnet werden sollen. Am gebräuchlichsten ist die Center of Gravity Methode (COG=Schwerpunktmethode) oder COGS (COG für Singletons)
<i>PredictionMethod</i>	<i>:= TAYLOR LINEARITY SPLINE FIT</i>	Art der Vorhersage von Daten. Entweder durch annähern eines Taylorpolynomes, gewichtete Linearität, extrapolation durch Splineinterpolation oder durch Fit mit Downhill-Simplex.
<i>Conclusion</i>	<i>:= (AtomOut) (SystemModify) < Conclusion, Conclusion GOTO RuleBlockName ></i>	Folgerung einer Regel. Neu ist hier <i>SystemModify</i> um Systemvariablen zu verändern und GOTO um den aktuellen Regelblock zu wechseln.
<i>AtomIn</i>	<i>:= (InputVariableName < IS IsTime IsExists GREATER LESS > FuzzyTermName)</i>	Atom in einer Bedingung
<i>AtomOut</i>	<i>:= (OutputVariableName < IS TimeIs TimeExists > FuzzyTermName) (EventName IS TaskName)</i>	Atom in einer Folgerung
<i>SystemCheck</i>	<i>:= (SystemVariableName < < > = > <REAL SystemVariableName >)</i>	Vergleicht die Werte von Systemvariablen, ob die kleiner, größer oder gleich einem gegebenen Wert oder dem Wert einer anderen Variablen ist.

<i>SystemModify</i>	 ::= <i>SystemVariableName</i> < ::= REAL ::= <i>SystemVariableName</i> ++ -- >	Systemvariablen können einen festen Wert zugewiesen bekommen, genauso wie sie um die Regelaktivierung erhöht beziehungsweise erniedrigt werden können. Sie können aber auch den Wert einer anderen Systemvariablen erhalten
<i>IsTime</i>	 ::= IS_TIME [<i>FuzzyTimeTermName</i>]	IS_TIME betrachtet, ob alle Daten in ihrer Gesamtheit in dem gegebenen FuzzyTerm der Bedingung liegen. Ein Fuzzy Zeit Termes bestimmt den Zeitbereich, aus aus welchen Daten betrachten werden. Siehe dazu [Schmidt04] und [Schmidt05].
<i>IsExists</i>	 ::= IS_EXISTS[<i>FuzzyTimeTermName</i>]	IS_EXISTS betrachtet, ob es ein Datum gibt, welches in dem gegebenen FuzzyTerm der Bedingung liegt. Ein Fuzzy Zeit Termes bestimmt den Zeitbereich, aus aus welchen Daten betrachten werden. Siehe dazu [Schmidt04] und [Schmidt05].

Tabelle 3: Erläuterungen zu Terminalsymbolen der EBNF-Beschreibung der Temporal Fuzzy Control Language. Änderungen zu Tabelle 2 sind dabei Fett hervorgehoben.

2.5. Auswerten von TFCL-Beschreibungsdateien

Hier beschreiben wir wie eine TFCL-Datei ausgewertet wird. Eine solche Datei beinhaltet zuallererst Eingabe- und Ausgabevariablen. Für diese Variablen sind ein oder mehrere Fuzzy-Terme gegeben. Jeder Fuzzy-Term, auch die Fuzzy-Zeit-Terme, wird durch einen Polygonzug beschrieben, welcher die Zugehörigkeitsfunktion $\mu(x)$ angibt. Wenn diese Informationen eingelesen sind, dann sind die Schnittstellen zum Regler definiert. Anschließend müssen nur noch die Regelblöcke eingelesen werden.

Jeder Regelblock hat einen eindeutigen Namen, durch welchen er identifiziert wird. Dies ermöglicht es, unterschiedliche Regelblöcke zu unterschiedlichen Zeiten oder unterschiedlich oft auszuwerten. Soll ein Regelblock ausgewertet werden, so werden alle Regeln in diesem einzeln und nacheinander ausgewertet. Regeln die zuerst in einem Regelblock stehen werden auch zuerst ausgewertet. Dies ist aber nur von Belang, wenn Systemvariablen und GOTO-Anweisungen verwendet werden. Bei Systemvariablen werden die Änderungen sofort nach der Regelauswertung übernommen. Deshalb hat eine unterschiedliche Auswertungsreihenfolge auch unterschiedliche Auswirkungen. Auch GOTO-Anweisungen werden sofort ausgeführt, so dass eine Auswertung der weiteren Regeln im Regelblock nicht mehr stattfindet. Die Defuzzifizierung wird sehr wohl noch ausgeführt, aber alle folgenden Regeln haben darauf keinen Einfluss mehr.

Die Regeln der Regelblöcke werden beim Einlesen in zwei Teile aufgeteilt. Die Regel-Bedingung und die Regel-Folgerung. Dadurch, dass die Daten in einer eigenen und effizienten Datenstruktur vorliegen, benötigen wir keinen Interpreter, der immer wieder Zeile für Zeile interpretiert. Dies Beschleunigt die Auswertung der TFCL-Dateien.

Die Regel-Bedingung besteht aus einzelnen Atomen, welche durch AND beziehungsweise OR verknüpft sind. Intern wird eine Regel-Bedingung als Datenstruktur durch einen Baum dargestellt. Außerdem kann die Reihenfolge der Auswertung durch Klammerung der Ausdrücke festgelegt werden. Bei jeder Klammerung steigt man eine Ebene tiefer in den Baum. Ausdrücke die innerhalb einer Klammerungsebene stehen befinden sich auch auf der gleichen Ebene im Baum. Die Entscheidung einen Baum anstatt die herkömmlich verwendete Regel-Matrix zu verwenden ist darin begründet, dass eine Matrix sehr viel Speicherplatz belegt und man in dieser keine OR-Verknüpfung und noch weniger verschachtelte Ausdrücke darstellen kann.

Die Regel-Folgerung dagegen ist sehr viel einfacher aufgebaut. Sie besteht aus einer Liste von Folgerungen, welche gelten, sollte die Regel-Bedingung gültig sein. Zur Auswertung wird jedes Element dieser Liste nacheinander von Links nach Rechts ausgewertet. Elemente die zuerst in der Liste stehen werden auch zuerst ausgewertet. Einen Einfluss auf die Auswertung hat noch der Gewichtungsfaktor einer Regel, welcher die Aktivierung einer Regel verstärken oder abschwächen kann. Der Einfluss hiervon geht direkt in die Folgerungen ein, da diese immer unter Beachtung der gewichteten Regelaktivierung ausgewertet werden. Zum Beispiel die am meisten vorkommende Folgerung „*x* IS hoch“. Die Ausgabevariable *x* ist hoch zu einem bestimmten Grad. Der Grad ist genau die gewichteten Regelaktivierung.

Nachdem die Daten eingelesen wurden. Können mit neuen Eingabewerten neue Ausgabewerte bestimmt werden. Diese Aufgabe unterteilt sich in vier Schritte.

Der erste Schritt der Auswertung ist die Aggregation (auch Fuzzifizierung genannt). Hierfür wird die Liste aller Regeln des aktiven Regelblockes sequentiell durchgegangen. Bei jeder Regel werden deren atomaren Bedingungen durch die vorliegenden Eingaben aktiviert. Die atomaren Fuzzy-Bedingungen werden durch traversieren des Baumes der Regel-Bedingungen gefunden.

Danach erfolgt die Aktivierung. Hierzu wird die Aktivierung aller atomaren Bedingungen, welche keine Kinder und jeweils den gleichen Vater im Baum haben, mit der Standard Fuzzy-Aktivierung berechnet. Das Ergebnis ist der Aktivierungsgrad des Vaters, dessen Kinder nun aus dem Baum gelöscht werden. Dieser Berechnungsschritt wird solange ausgeführt, bis der Baum nur noch aus einer Wurzel besteht. Die Aktivierung der Wurzel entspricht dann der Aktivierung der Regel. Bei der Berechnung hat eine AND-Verknüpfung eine höhere Priorität wie eine OR-Verknüpfung. Nach diesem Schritt ist für jede Regel bekannt, wie stark diese feuert.

Anschließend wird die Akkumulation durchgeführt. Hierzu setzt man die Aktivierung von jedem Fuzzy-Term in jeder Ausgabevariablen auf Null. Danach setzt man für jede Regelfolgerung die Aktivierung der Regel nach der MAX- oder PROD-Methode in den angegebenen Fuzzy-Term, der durch die Folgerung gegeben ist. Das Ergebnis von diesem Schritt sind Ausgabevariablen mit aktivierten Fuzzy-Termen. Außerdem gibt es noch Folgerungen welche Variablen (zum Beispiel Systemvariablen) direkt ändern (zum Beispiel durch den Operator ++) oder den Programmablauf beeinflussen (zum Beispiel durch den Operator GOTO).

Der letzte Schritte ist die Defuzzifizierung. Mit den aktivierten Fuzzy-Termen in den Ausgabevariablen bestimmt man nun durch die gegebene Defuzzifizierungsmethode für jede Ausgabevariable deren scharfen Ausgabewert.

Damit ist die Auswertung der TFCL-Datei abgeschlossen. Gelangen neue Daten in das Fuzzy-System, so werden nur noch die letzten vier Schritte wiederholt. Das Einlesen der Beschreibungsdatei und das Aufbauen der Bäume zur Beschreibung der Regel-Bedingungen entfällt, da sich diese beim Programmablauf nicht ändern.

3. Vergleich vom Fuzzy- und PID-Reglern am Beispiel eines simulierten Stabwagens

In diesem Kapitel vergleichen wir qualitativ und quantitativ verschiedene PID- und Fuzzy-Regler miteinander. Dies geschieht am Beispiel eines simulierten Stabwagens mit einem inversen Pendel. Ein wichtiges Augenmerk liegt bei der Stabilitätsuntersuchung und der Zeit zum Entwickeln und Implementieren eines Reglers.

Zuerst zum Experiment selbst. Beim inversen Pendel handelt sich um ein Beispielproblem, das Balancieren eines Metallstabes im Schwerfeld durch Bewegung seines Fußpunkts (Abbildung 2). Das Beispiel ist an das inverse Pendel aus [Bothe95] angelehnt, bei welchem das Pendel zwar vorgestellt wird, aber keine Untersuchung des Verhaltens oder eine Auswertung der Regeln berechnet wird. Vergleiche dieser Art können in der Literatur bei [Butkiewicz00] gefunden werden. Hier wird ebenfalls ein Fuzzy-Logik-Regler mit einem PID-Regler verglichen, aber es wird nur die Zeit und das Schwingverhalten untersucht. Außerdem wurde kein temporaler Fuzzy-Regler bei den Experimenten verwendet. Dies und die wirkenden Kräfte untersuchen wir hier.

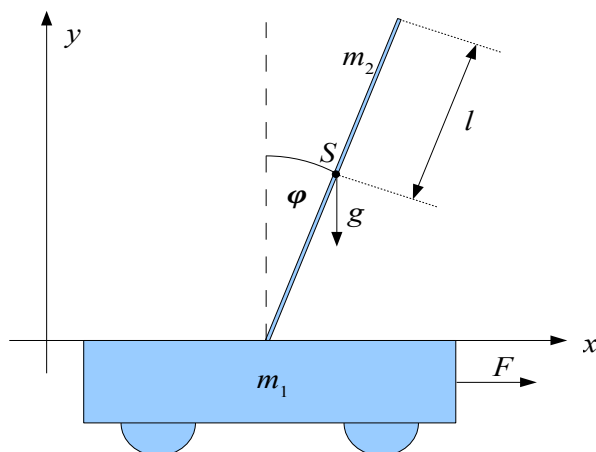


Abbildung 2: Inverses Pendel auf Stabwagen.

Die Untersuchung in [Chang97] geht noch ein Stück weiter. Hier werden PID- und Hybrid-Regler (= P-Fuzzy + ID-Regler) in Kombination mit einer manuellen Steuerung einer Verbrennungsanlage verglichen. Die Überwachten Kriterien sind Dampfdruck und -fluss und Sauerstoff- und Kohleverbrauch. Das Ergebnis der Untersuchungen ist, dass der Hybrid-Regler dem PID-Regler überlegen ist.

Bei unserem Experiment sei zur Vereinfachung im Folgenden angenommen, dass die Masse im Metallstab homogen verteilt ist. Die Massen m_1 und m_2 sind durch ein Drehgelenk reibungsfrei miteinander Verbunden. Hierbei erlaubt das Drehgelenk nur eine Bewegung in einer Ebene, der Stab hat also nur einen Freiheitsgrad. Der Winkel zwischen dem Stab und der y -Achse beträgt φ . Mit der Kraft F kann die untere Masse m_1 längs der x -Achse beschleunigt und dadurch der Stab verschoben beziehungsweise balanciert werden. Als Werte setzen wir folgende Konstanten ein: $m_1 = 1\text{kg}$, $l = 2\text{m}$, $m_2 = m_1$, $g = 9.81\text{m/s}^2$.

Betrachtet man die Richtungen, in welche die Kräfte F und g in Bezug des Massenschwerpunktes S wirken, dann erhält man für die Beschleunigungen der Auslenkung $\ddot{\varphi}$ und der Beschleunigung des Wagens \ddot{x} nach [Schneider00] folgende Gleichungen:

$$\ddot{\varphi} = \frac{g \cdot \sin(\varphi) + \cos(\varphi) \frac{F + m_2 l \dot{\varphi}^2 \sin(\varphi)}{m_1 + m_2}}{l \left(\frac{4}{3} - \frac{m_2 \cos^2(\varphi)}{m_1 + m_2} \right)} \quad \text{mit} \quad \begin{aligned} \ddot{\varphi} &= \frac{\partial \dot{\varphi}}{\partial t} = \frac{\partial^2 \varphi}{\partial t^2} \\ \ddot{x} &= \frac{\partial \dot{x}}{\partial t} = \frac{\partial^2 x}{\partial t^2} \end{aligned}$$

$$\ddot{x} = \frac{F + m_2 l (\dot{\varphi}^2 \sin(\varphi) - \ddot{\varphi} \cos(\varphi))}{m_1 + m_2}$$

Um vergleichbare und wiederholbare Ergebnisse zu erhalten, betrachten wir den Stabwagen nur in einer simulierten Umgebung mit obigen Formeln. In der Simulation bringen 5 verschiedene Regler (3 PID und 2 Fuzzy) den Stab in eine Ruhelage und halten ihn dort. Der Startwinkel ist immer $\varphi_0 = +1,0$ [rad] $\approx 57,3^\circ$. Die Simulationszeit beträgt 2 Sekunden. Innerhalb dieser Zeitspanne ist es allen Reglern möglich die Auslenkung auf Null zu reduzieren und den Stab in dieser Lage zu halten.

Gewünscht ist eine möglichst kurze Zeit zum Stabilisieren des Stabes. Stabil bedeutet, dass sich der Stab nur noch ganz wenig bis gar nicht bewegt. Wir definieren die stabile Lage als eine Auslenkung von weniger als $\Delta\varphi = 0,005$ ($<0,29^\circ$), was bei einer Stablänge l von 2 Metern einer Schwankung an der Spitze von weniger als plus minus einem Zentimeter entspricht. Für einen Zeitpunkt t_0 , ab welchem der Stab stabil ist, gilt folgende Bedingung:

$$\min_{t_0} (\exists t_0 \forall t \geq t_0 : \varphi(t) \leq 0,005 < 0,29^\circ)$$

Alle fünf Regelkreise sind im folgenden in PID- und Fuzzy-Regler unterteilt und im jeweiligen Kapitel 3.1 für PID-Regler und 3.2 für Fuzzy-Regler genauer beschrieben. In Kapitel 3.3 werden die Ergebnisse der einzelnen Regler im Experiment dargestellt und miteinander verglichen.

3.1. Der PID-Regler

Der PID-Regler unterteilt sich in drei unabhängige Regler. Den Proportional-, Integral- und Differential-Regler. Jeder Regler hat als Eingang eine Regeldifferenz $e(t)$. Die Regeldifferenz gibt an, um wie viel bei einem zu regelnden Prozess ein Istwert von einem Sollwert abweicht. Das Ziel des Reglers ist die Regeldifferenz klein und wenn möglich auf Null zu halten. Um dies zu leisten, kann der Regler durch einen Stellwert $u(t)$ direkt oder indirekt Einfluss auf den Prozess nehmen.

Der einfachste Regler ist der P-Regler. Er verstärkt durch den Proportionalitätsfaktor K_P die Regeldifferenz $e(t)$ und reagiert unmittelbar auf eine Veränderung in der Regeldifferenz. Problematisch ist es, wenn ein großes K_P verwendet wird, da in diesem Fall der Prozess dazu

neigt, sich aufzuschwingen und so die Regeldifferenz immer größer wird. Der P-Regler sieht wie folgt aus: $u(t) = K_P \cdot e(t)$

Der I-Regler reagiert langsamer auf Abweichungen in der Regeldifferenz als der P-Regler, da sich die Abweichungen erst über ein Integral über vergangene Werte aufsummieren müssen. Durch diese Trägheit ist der I-Regler unempfindlicher gegenüber kleinen Störungen. Es ist zu beachten, dass der Zeitraum über den integriert wird hinreichend lang sein muss, denn ansonsten würde der I-Regler dem P-Regler entsprechen. Der Faktor K_I gibt an, wie stark der Integralanteil auf den Stellwert wirkt. Der I-Regler sieht wie folgt aus: $u(t) = K_I \cdot \int e(t) dt$

Der D-Regler reagiert nur auf Änderungen in der Regeldifferenz. Ist die Regeldifferenz konstant, so bleibt er inaktiv. Deshalb kann der D-Regler auch keine konstanten Regeldifferenzen ausgleichen. Dies macht ihn ohne einen anderen Regler unbrauchbar. Der Vorteil des D-Reglers ist seine schnelle Reaktion auf Störungen. Der Faktor K_D gibt an, wie stark der Differenzialanteil auf den Stellwert wirkt. Der D-Regler sieht wie folgt aus: $u(t) = K_D \cdot \frac{\partial e(t)}{\partial t}$

Verschiedene Kombinationen von P-, I- und D-Reglern sind möglich. Wir beschränken uns auf die gebräuchlicheren Kombinationen von P-, PI- und PID-Regler.

Alle Regler liefern ein Steuersignal $u(t)$, aber zum Regeln des Prozesses muss bekannt sein, wie das Steuersignal $u(t)$ Einfluss auf dem Prozess nimmt. Bei dem Beispiel des Stabwagens gibt das Steuersignal die gewünschte Drehgeschwindigkeit $\dot{\varphi}$ des Stabes an. Diese kann aber nicht direkt eingestellt werden. Nur die Kraft F , mit welcher der Stabwagen beschleunigt wird, kann geändert werden. Demnach benötigt man noch eine Umrechnung der benötigten

Kraft F , um die gewünschte Geschwindigkeitsänderung $\Delta \frac{\partial \dot{\varphi}}{\partial t} = \ddot{\varphi}$ des Stabes zu erreichen.

Stellt man die Gleichung für die Winkelbeschleunigung aus dem vorherigen Kapitel nach der Kraft F um, so erhält man folgende Gleichung:

$$F = \frac{m_1 + m_2}{\cos(\varphi)} \left(l \left(\frac{4}{3} - \frac{m_2 \cos^2(\varphi)}{m_1 + m_2} \right) \cdot \ddot{\varphi} - g \cdot \sin(\varphi) \right) - m_2 l \dot{\varphi}^2 \sin(\varphi)$$

Die Kraft F , welche nötig ist, um das System durch einwirken der Kraft F für Δt Sekunden so zu beschleunigen, dass die neue Drehgeschwindigkeit erreicht wird, wird mit einer vereinfachten Formel bestimmt. Wir gehen davon aus, dass die aktuelle Auslenkung des Stabes nahezu Null ist. Dadurch kann man den $\sin(\varphi)$ beziehungsweise $\cos(\varphi)$ mit 0 beziehungsweise 1 annähern. Diese Berechnung der Kraft F ist nur als Näherung für kleine Auslenkungen φ korrekt. Ist die Auslenkung φ des Stabes sehr klein, so gilt: $F \approx \frac{4m_1 + m_2}{3} \cdot l \cdot \ddot{\varphi}$

Um eine möglichst optimale Regelung zu finden, müssen optimale Parameter K_i gefunden werden. Da das Pendel nur simuliert wird, kann man es sich erlauben einfach alle Parameter mit einer kleinen Diskretisierung (Schrittweite 0.05) in der Simulation zu testen. Getestet wird die Dauer zum Stabilisieren des Stabes. Liefern mehrere Parameter beziehungsweise Parameterpaare die kürzeste Stabilisierungszeit, so entscheiden wir uns für den Median der Parameter. Genauer gesagt zuerst für den Median von K_P , dann von K_I und dann von K_D . Die Parameter, bei welchen der Stab die geringste Zeit zum Stabilisieren benötigt sind in

Regler	k_P	k_I	k_D
P	-13,00	0	0
PI	-7,20	-0,90	0
PID	-7,5	-1,65	-0,35

Tabelle 4: Optimale Regler Parameter beim hier vorgestellten inversen Pendel.

Tabelle 4 gegeben.

Der PID-Regler lässt sich in Pseudocode wie im Folgenden angegeben beschreiben. Sind die Parameter K_I und K_D gleich Null beziehungsweise der Parameter K_D gleich Null so beschreibt der Pseudocode einen P- beziehungsweise einen PI-Regler.

```

1 // Ältesten Wert werfen
2 History.shift ();
3
4 // Aktuelle Auslenkung merken
5 History.add ( $\varphi$ );
6
7 // P-, I- und D-Anteile berechnen
8  $\dot{\varphi}' = \text{Auslenkung} * K_P;$ 
9  $\dot{\varphi}' += \text{History.integral} (5 * \Delta t) * K_I;$ 
10  $\dot{\varphi}' += \text{History.derivate} (\Delta t) * K_D;$ 
11
12 // Kraft zum Ändern der Drehgeschwindigkeit
13  $F = (\dot{\varphi} - \dot{\varphi}') / \Delta t * l * (4 * m_1 + m_2) / 3;$ 

```

3.2. Der Fuzzy-Regler

Den Fuzzy-Regler stellen wir in zwei Varianten vor. Einmal den klassischen Fuzzy-Regler und dann den zeitlich erweiterten temporalen Fuzzy-Regler. Beide Fuzzy-Regler erhalten die gleichen Eingaben (Auslenkung φ und Drehgeschwindigkeit $\dot{\varphi}$) und regeln beide die Kraft F des Stabwagens. Auch sind die Zugehörigkeitsfunktionen der Fuzzy-Terme identisch, so dass der einzige Unterschied in den verwendeten Prädikaten liegt. Der klassische Fuzzy-Regler verwendet nur das Prädikat **IS**, der temporale Fuzzy-Regler nur das Prädikat **IS_{EXISTED}**.

Eine Optimierung der Regler findet ausschließlich über eine Veränderung der Zugehörigkeitsfunktionen statt. Die Regelbasis wird dabei nicht verändert. Es hat sich gezeigt, dass es genügt die Zugehörigkeitsfunktion des Fuzzy-Terms für die Kraft F anzupassen. Es wird dabei ausgegangen, dass die Kraft F im Bereich -100N bis +100N liegt und es günstiger ist, wenn die meisten Zugehörigkeitsfunktionen einen kleinen Bereich von -15N bis +15N beschreiben. Diese Annahmen legen schon die meisten Stützpunkte fest. Die letztendlich als optimal gefunden Zugehörigkeitsfunktionen können der angegebenen TFCL-Datei entnommen werden (Zeilen 32-43). Gefunden werden diese, indem bei den Termen NG die dritte und NM die erste und zweite Stützstelle variiert werden. Die Terme PM und PG werden symmetrisch dazu behandelt. Die Stützstellen werden solange im Bereich zwischen 15 und 80 verschoben, bis eine schnelle Stabilisierung gefunden ist.

Eine vollständige Angabe der Regelbasis bei zwei AND-verknüpften Variablen mit 5 beziehungsweise 7 Fuzzy-Termen besteht aus 35 Regeln. Diese sind in Tabelle 5 als Matrix angegeben. Die Matrix gibt an, welche Regeln gebildet werden. Für den klassischen Fuzzy-Regler gilt zum Beispiel bei einer negativen mittleren (NM) Auslenkung φ und einer negativen kleinen (NK) Drehgeschwindigkeit $\dot{\varphi}$, dass die Kraft F negativ mittel (NM) sein soll. Für den temporalen Fuzzy-Regler ändern sich nur die Prädikate und es wird eine Zeitangabe hinzugefügt. Die Regeln sehen wie folgt aus:

FCL: **IF (φ IS NM) AND ($\dot{\varphi}$ IS NK) THEN (F IS NM)**

TFCL: **IF (φ IS_{EXISTS next} NM) AND ($\dot{\varphi}$ IS_{EXISTS next} NK) THEN (F IS NM)**

Die beiden Fuzzy-Regler lassen sich durch folgende TFCL-Beschreibung angeben. Angegeben sind die Stützpunkte -79 und -58, -58 (siehe Zeile 33-34) für den temporalen Fuzzy-Regler. Der klassische Fuzzy-Regler hat die Stützpunkte -75, -61 und -61. Der Fuzzy-Zeit-Term in den Zeilen 45-49 und die Angabe der Vorhersagemethode in Zeile 56 werden nur für den temporalen Fuzzy-Regler benötigt. Alle anderen Zeilen sind für beide Fuzzy-Regler identisch.

```

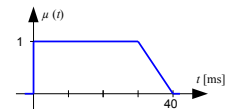
1 FUNCTION_BLOCK
2
3 VAR_INPUT
4   auslenkung: REAL;
5   drehgeschwindigkeit: REAL;
6 END_VAR
7
8 VAR_OUTPUT
9   kraft: REAL;
10 END_VAR
11
12 FUZZIFY auslenkung
13   TERM NG := (-1.60,1)(-1.00, 1)(-0.14, 0);
14   TERM NM := (-1.00,0)(-0.14, 1)(-0.07, 0);
15   TERM NK := (-0.14,0)(-0.07, 1)( 0.00, 0);
16   TERM ZR := (-0.07,0)( 0.00, 1)( 0.07, 0);
17   TERM PK := ( 0.00,0)( 0.07, 1)( 0.14, 0);
18   TERM PM := ( 0.07,0)( 0.14, 1)( 1.00, 0);
19   TERM PG := ( 0.14,0)( 1.00, 1)( 1.60, 1);
20   RANGE := (-1.60 .. 1.6);
21 END_FUZZIFY
22
23 FUZZIFY drehgeschwindigkeit
24   TERM NM := (-6.30,0)(-0.40, 1)(-0.20, 0);
25   TERM NK := (-0.40,0)(-0.20, 1)( 0.00, 0);
26   TERM ZR := (-0.20,0)( 0.00, 1)( 0.20, 0);
27   TERM PK := ( 0.00,0)( 0.20, 1)( 0.40, 0);
28   TERM PM := ( 0.20,0)( 0.40, 1)( 6.30, 0);
29   RANGE := (-6.3 .. 6.3);
30 END_FUZZIFY
31
32 DEFUZZIFY kraft

```

```

33     TERM NG := (-100,1)(-79, 1)(-58, 0);
34     TERM NM := (-58,0)(-15, 1)(-10, 0);
35     TERM NK := (-15,0)(-10, 1)( 0, 0);
36     TERM ZR := (-10,0)( 0, 1)( 10, 0);
37     TERM PK := ( 0,0)( 10, 1)( 15, 0);
38     TERM PM := ( 10,0)( 15, 1)( 60, 0);
39     TERM PG := ( 60,0)( 80, 1)(100, 1);
40     METHOD: COG;
41     DEFAULT := 0;
42     RANGE := (-100 .. 100);
43     END_DEFUZZIFY
44
45     FUZZY_TIME_OBJECT next
46     FACT 1: (0,1) (1,1);
47     TIME 1: (0,0) (0,1)
48             (30,1) (40,0);
49     END_FUZZY_TIME_OBJECT
50     RULEBLOCK control
51     AND:MIN;
52     OR:MAX;
53     ACCU:MAX;
54     ACT:MIN;
55     PREDICTION:LINEARITY;
56     RULE 0: (Regeln siehe Tabelle 5)
57     END_RULEBLOCK
58
59     END_FUNCTION_BLOCK

```



3.3. Ergebnisauswertung

Wir vergleichen nun die fünf vorgestellten Regler anhand von acht verschiedener Kriterien. Zuerst stellen wir die Kriterien vor und dann untersuchen wir jeden Regler anhand dieser Kriterien im Vergleich mit den anderen Reglern.

Das einfachste Kriterium ist die Stabilisierungszeit s , die ein Regler benötigt um den Stab in eine stabile Lage zu bringen. Stabil bedeutet, dass die Auslenkung φ ab dem Zeitpunkt t_0 immer kleiner als 0.005 ($= 0.29^\circ$) bleibt. Ob die Auslenkung gegen Null geht oder innerhalb dieser Schwankungsbreite weiter schwingt spielt dabei keine Rolle.

Ein weiteres Kriterium ist das Integral über die Auslenkung φ oder bei diskreter Messung die Summe über die Auslenkungen. Ist diese Zahl klein, so wird der Stab sehr schnell in die Nullstellung gebracht und bleibt auch in dieser. Ein hoher Wert gibt an, dass der Stab entweder langsam ausbalanciert wird oder der Stab nicht ausbalanciert wird beziehungsweise immer noch kleine Schwingungen ausweist.

Das Integral über die Kraft F (die Summe über die diskrete Kraft) gibt an, wie viel Energie der Regler zum Stabilisieren und anschließenden Halten in dieser Stellung benötigt. Eine hohe Energie ist ineffizient aber nicht unbedingt langsam.

Die Summe über die Winkelgeschwindigkeit $\dot{\varphi}$ gibt den Weg an, den der Stab zurücklegt. Ein Wert von 100 bedeutet, dass der Stab den kürzest möglichen Weg zurückgelegt hat, also den direkten Weg zum Ziel ohne Überschwinger oder Schwingungen genommen hat.

Die Summe über die quadratische Änderung der Kraft gibt an, welcher Regler sehr oft die Kraft stark ändert. Ein hoher Wert entspricht einem hohen Verschleiß der Hardware.

Nach diesen fünf quantitativen Kriterien folgen drei qualitative Beurteilungen. Der Verlauf der Auslenkung beschreibt, wie sich der Stab verhält und wie seine Bewegung aussieht. Dagegen beschreibt der Verlauf der Kraft, wie der Stab ausbalanciert wird. Das letzte Kriterium beschreibt subjektiv den Aufwand, der zum Entwickeln des jeweiligen Regler nötig war.

In Tabelle 6 sind die wichtigsten Ergebnisse der Simulation zusammengefasst. Hell beziehungsweise dunkel hinterlegt sind die besten beziehungsweise schlechtesten Werte einer Spalte.

		Drehgeschwindigkeit $\dot{\varphi}$				
		NM	NK	ZR	PK	PM
Auslenkung φ	NG	NG	NG	NG	NG	NG
	NM	NG	NM	nm	nk	ZR
	NK	NM	NK	nk	zr	pk
	ZR	NM	nk	ZR	pk	PM
	PK	nk	zr	pk	PK	PM
	PM	ZR	pk	pm	PM	PG
PG	PG	PG	PG	PG	PG	

Tabelle 5: Matrix mit den Fuzzy-Regeln für die Kraft F zum Regeln des Stabwagens. Fuzzy-Terme in Großbuchstaben sind in [Bothe95] definiert, der Rest ist von uns hinzugefügt. Die Fuzzy-Terme reichen von NG (= negativ, groß) über ZR (= zero) bis PG (=positiv, groß)

Ähnliche Werte werden ebenfalls hell beziehungsweise dunkel hinterlegt. Als ähnlich schlecht zählen Werte, welche näher am schlechtesten Wert als am Mittelwert von Maximum und Minimum einer Spalte liegen. Für die besten Werte gilt dies analog. In Abbildung 3 sind die Verläufe der Kraft und der Auslenkung der dazugehörigen Regler darstellt. Das Verhalten der PID-Regler (P, PI und PID) ist im Großen und Ganzen recht ähnlich. Deshalb beschreiben wir sie auch nicht getrennt voneinander, sondern vergleichen diese mit den beiden

Fuzzy-Reglern, die jedoch Unterschiede untereinander aufweisen.

Bei quantitativer Untersuchung der Ergebnisse in Tabelle 6 kann man erkennen, dass je nachdem, welches Kriterium wichtig ist entweder die PID- oder einer der Fuzzy-Regler besser sind. Ist es wichtig, dass der Regler schnell ist (Stabilisierungszeit t_0), dann sind die PID-Regler besser, ist es aber wichtig, dass wenig Energie (ΣF) zur Regelung benötigt wird oder die Hardwareabnutzung ($\sqrt{\Sigma dF^2}$) gering ist, dann sind die Fuzzy-Regler besser.

Die qualitative Untersuchung der Graphen der verschiedenen Regler in Abbildung 3 zeigt, dass die Auslenkung (dunkler Graph) sich immer recht ähnlich verhält, aber bei den Kräften (heller Graph) größere Unterschiede vorkommen. Dies spiegelt sich auch in den zuvor beschriebenen quantitativen Ergebnissen wieder. Auffällig ist, dass die Fuzzy-Regler insgesamt ein weiches, aber dadurch auch langsames Regelverhalten aufweisen. Das Verhalten äußert sich darin, dass die Kräfte bei den PID-Reglern kurz vor Stillstand mehrfach zwischen positiven und negativen Kräften hin und her springen. Dies bewirkt ein schnelles aber abruptes abbremsen. Die Fuzzy-Regler dagegen weisen einen recht glatten Kräfteverlauf mit nur sehr kleinen Sprüngen kurz vorm Stillstand auf. Dadurch wirken kleinere Beschleunigungen auf den Stab und die Hardware wird weniger beansprucht, was aber auch in einem langsameren Abbremsen resultiert.

Ein ebenfalls wichtiges Kriterium ist der Aufwand zum Erstellen eines Reglers. Dazu zählt sowohl die Modellierung als auch die anschließende Implementierung in der Programmiersprache C. Gemeinsam ist beiden Reglern der Programmieraufwand. Der PID-Regler ist in Kapitel 3.1 als Gleichungen gegeben und der Fuzzy-Regler als FCL-Programm in Kapitel 3.2. Der eigentliche Aufwand beim PID-Regler besteht darin, die richtige Stellgröße zu finden, da die Kraft F nicht als Stellgröße funktionieren kann. Die Umrechnung der Änderung der Drehgeschwindigkeit in eine Kraft stellte hier den nicht sehr großen Aufwand dar. Der Aufwand beim Erstellen des Fuzzy-Reglers beschränkt sich darauf, die FCL-Datei mit den Fuzzy-Regeln zu schreiben. Da die Fuzzy-Regeln die Regelung so beschreiben, wie sie ein Mensch auch beschreiben könnte, ist dieser Aufwand sehr gering. Aus diesem Grund geben wir der Entwicklung eines Fuzzy-Reglers einen geringeren Aufwand.

Wir bewerten nun die positiven (+1) und negativen (-1) Kriterien. Als Gesamtergebnis ergibt sich, dass die Regelung mit klassischer Fuzzy-Logik am schlechtesten und die Regelung mit temporaler Fuzzy-Logik am besten abschneidet. Die PID-Regler liegen im Gesamtergebnis zwischen den beiden Fuzzy-Reglern.

4. Schlussfolgerungen

Als Fazit stellt sich heraus, dass zum Entwickeln des PID-Reglers fundiertes Wissen über den zu regelnden Prozess nötig ist. Mit diesem Wissen können die möglicherweise nötigen Umrechnungen für eine geschlossene Regelkette berechnet werden. Der PID-Regler hat als Ausgabe die neue einzustellende Drehgeschwindigkeit. Diese muss zuerst in eine Winklebeschleunigung und anschließend in eine Kraft umgerechnet werden. Der Grund hierfür ist, dass wenn sich das Pendel schon in die richtige Richtung bewegt, dann muss es auch bei einer großen Auslenkung nicht weiter beschleunigt werden. Die Kraft muss also auch bei einer großen Auslenkung Null sein können, sie kann also nicht proportional zur Auslenkung gewählt werden. Würde die Kraft proportional zur Auslenkung gewählt werden, dann

Regler	Σ Auslenkung ϕ	Σ Kraft F	$\Sigma \dot{\phi}$	Stabilisierungszeit t_0	$\sqrt{\Sigma dF^2}$	Aufwand	Gesamt
P	37,47	5814	100,8	0,56	189	mittel	+1
PI	37,48	5777	100,8	0,56	186	mittel	+1
PID	37,48	5842	100,8	0,56	210	mittel	+1
FCL	44,29	2353	102,0	0,76	92	wenig	0
TFCL	42,57	2452	101,4	0,74	93	wenig	+2

Tabelle 6: Ergebnisse aus den Experimenten mit den verschiedenen Reglern. Die rechte Spalte gibt das Gesamtergebnis bei hell +1 Punkt und dunkel -1 Punkt an.

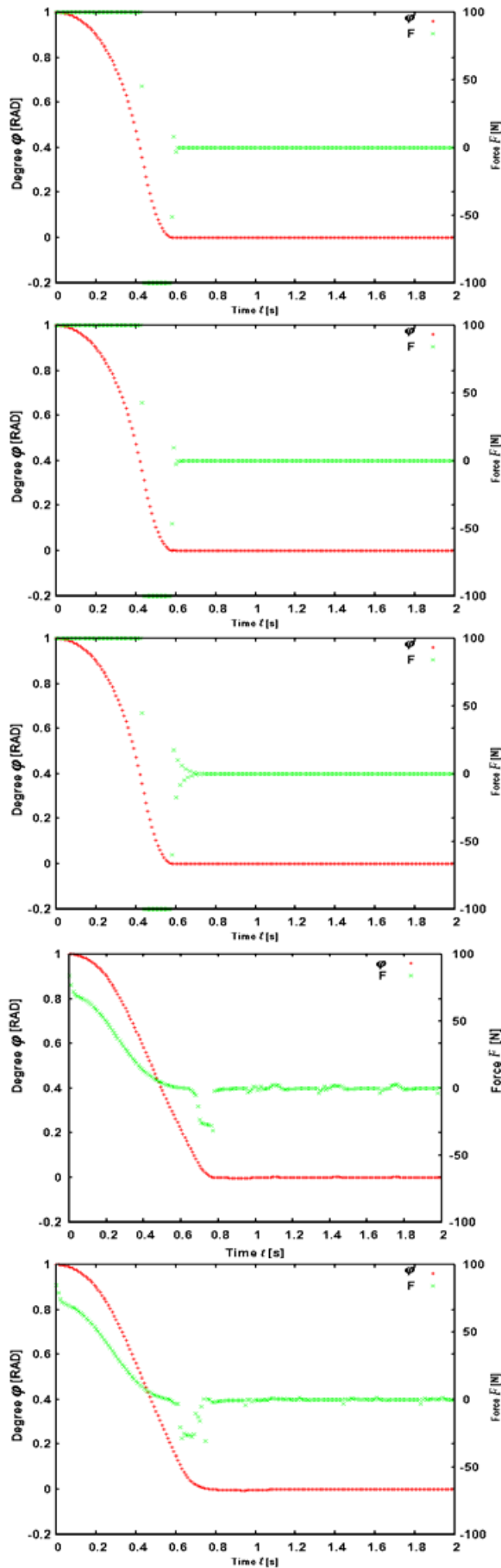


Abbildung 3: Auslenkungs- und Kraftverläufe des Stabwagens für die Regler: P, PI, PID, FCL und TFCL.

würde der P-Regler bei einer großen Auslenkung so lange beschleunigen, bis die Auslenkung Null ist. Erst dann würde auch die Kraft Null werden. Da der Stab aber beschleunigt wurde, besitzt er eine große Geschwindigkeit und schießt über das Ziel hinaus. Da zu Anfangs die Auslenkung klein ist, würde durch den Proportionalanteil nur eine kleine Kraft wirken und den Stab nur sehr langsam abbremsen. Das Ergebnis ist eine sehr große und ungedämpfte Schwingung.

Beim Fuzzy-Regler dagegen genügt es, das Problem als Mensch real oder in Gedanken lösen und auch beschreiben zu können. Eine Lösungsbeschreibung, welche in Fuzzy-Regeln beschrieben ist, kann direkt von einem Fuzzy-Regler verarbeitet werden. Lässt sich das Problem nicht direkt in Fuzzy-Regeln ausdrücken, so muss es zuerst übersetzt werden. Dies kann nur durch einen Experten geschehen, der sich mit Fuzzy-Logik und insbesondere hier mit temporaler Fuzzy-Logik auskennt.

Eine Optimierung ist bei beiden Reglerarten möglich. Beim PID-Regler gibt es drei Parameter, K_P , K_I und K_D die optimiert werden. gute Werte können in einer Simulation sehr schnell gefunden werden, aber auch ohne Simulation ist die Parameterfindung nicht schwierig. Beim Fuzzy-Regler dagegen ergeben die Anzahl der Fuzzy-Terme und die Wahl der Stützstellen für die Zugehörigkeitsfunktionen einen so großen Suchraum, dass eine automatische Suche nach guten Zugehörigkeitsfunktionen nicht mehr möglich ist. Hier im Beispiel wären es 24 Parameter bei der gegebenen Anzahl von Fuzzy-Termen. Wird die Anzahl der Fuzzy-Terme noch variiert, dann steigt somit auch die Anzahl der Parameter weiter an. Es bietet sich an, die Fuzzy-Terme und Zugehörigkeitsfunktionen so zu wählen, wie sie sich aus der Beschreibung eines Menschen ergeben. Bessere Parameter kann man dann erhalten, indem man nur wenige Stützstellen verändert oder aber die Stützstellen nur ein wenig verändert. Erstere Methode haben wir mit besseren Erfolg verwendet.

Beide Arten von Reglern benötigen einen Experten, der sich mit dem verwendeten Regler und dem zu regelnden Prozess auskennt. Jedoch benötigt man weniger Zeit um eine Fuz-

zy-Regelbasis aufzustellen, als für die Umrechnung die bei einem PID-Regler nötig sein kann. Auch liest sich die Implementierung in Fuzzy-Logik einfacher (vergleiche PID-Pseudocode mit TFCL-Datei) und bleibt dadurch wartbarer. Der PID-Regler dagegen ist deutlich effizienter in seiner Leistung, was aber auch zu Lasten der Hardwareabnutzung geht. Wie zu sehen ist, haben beide Arten der Regelung ihre Vor- und Nachteile, die je nach Einsatzgebiet wichtiger oder zu vernachlässigen sind.

Als Gesamtergebnis stellt sich jedoch heraus, dass der temporale Fuzzy-Regler etwas besser als die PID-Regler und der klassischer Fuzzy-Regler etwas schlechter als die PID-Regler ist. Dieses Ergebnis hat aber keine Allgemeingültigkeit, sondern trifft nur auf dieses untersuchte Experiment zu. Eine allgemeingültige Aussage kann auch nie getroffen werden, da immer subjektive Entscheidungen bei einem solchen Vergleich nötig sein werden.

Wenn man sicher gehen muss, dass der Regler in jedem Fall stabil ist, dann sei noch auf die Ergebnisse von [Wang93] verwiesen. Dort werden Erweiterungen vorgestellt, mit welchen bei einem Fuzzy-Regler durch adaptives Verhalten sichergestellt wird, dass er immer stabil ist.

5. Literatur

- [Bothe95] H.-H. Bothe, *Fuzzy-Logik – Einführung in Theorie und Anwendungen*, Springer-Lehrbuch, 2. Auflage, Berlin Heidelberg, 1995
- [Butkiewicz00] B. S. Butkiewicz, *System with Hybrid Fuzzy-Conventional PID Controller*, International Conference on Systems, Man and Cybernetics, Nashville, 2000
- [Bovenkamp98] E. G. P. Bovenkamp, J. C. A. Lubbe, *Temporal Reasoning with Fuzzy-Time-Objects*, 4th International Workshop on Temporal Representation and Reasoning, Daytona Beach, Florida, May 10-11, 1997
- [Cárdenas02] M. A. Cárdenas Viedma, R. Martín Morales, *Syntax and Semantics for a Fuzzy Temporal Constraint Logic*, Annals of Mathematics and Artificial Intelligence, Volume 36, 2002
- [Chang97] X. Chang, W. Li, *Application of hybrid fuzzy logic proportional plus conventional integrall-derivate controller to combustion control of stoker-fired boilers*, International Fuzzy Systems Association, Amsterdam, 2000
- [Fick00] A. Fick, H. B. Keller, *Modellierung des Verhaltens Dynamischer Systeme mit erweiterten Fuzzyregeln*, Proceedings 10. Workshop Fuzzy Control des GMA-FA 5.22, Dortmund, Germany, 18. - 20. Oktober 2000
- [Giron02] G. Ortega, J. M. Giron-Sierra, *A survey of fuzzy logic control with aerospace applications*, 15th IFAC (International Federation of Automatic Control) World Congress, Barcelona, 2002
- [IEC97] IEC TC65/WG 7/TF8, *Fuzzy Control Programming*, International Technical Electronical Commission (IEC), 1997
- [Karjoth87] G. Karjoth, *Prozessalgebra und temporale Logik – angewandt zur Spezifikation und Analyse von komplexen Protokollen*, Diss. Mathematik/Informatik, Universität Stuttgart, 1987
- [Schmidt04] T. W. Schmidt, D. Henrich, *Temporal erweiterte Prädikate der Fuzzy-Logik zur Überwachung und Wartung*, Proceedings 14. Workshop Fuzzy Control des GMA-FA 5.22, Dortmund, Germany, 10. - 12. November 2004
- [Schmidt05] T. W. Schmidt, D. Henrich, *Inferenz mit Fuzzy-Zeit-Termen*, Proceedings 15. Workshop Fuzzy Computational Intelligence des GMA-FA 5.22, Dortmund, Germany, 16. - 18. November 2005
- [Schneider00] U. Schneider, *Das invertierte Pendel*, <http://www.informatik.htw-dresden.de/~iwe/Belege/SchneiderPendel/pendel.html>, August 2000
- [Takagi85] T. Takagi, M. Suego, *Fuzzy identification of systems and its application to modeling and control*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 15, No. 1, pp. 116-132, 1985.
- [Wang93] L.-X. Wang, *Stable Adaptive Fuzzy Control of Nonlinear Systems*, IEEE Transactions on Fuzzy Systems, Vol 1, No. 2, May 1993

Adaptive Optimization of the Number of Clusters in Fuzzy Clustering

Jürgen Beringer and Eyke Hüllermeier

Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg
Tel.: (0391) 6718665
Fax: (0391) 6712020
E-Mail: {beringer,huellerm}@iti.cs.uni-magdeburg.de

Abstract

In this paper, we present a local, adaptive optimization scheme for finding the best number of clusters in fuzzy C -means clustering. Even though this method is especially motivated by online applications in which a potentially changing clustering structure must be maintained over time, it turns out to be useful also in the static case. As part of the method, we propose a new validity measure for clustering structures which overcomes deficiencies of the commonly used Xie-Beni index.

1 Introduction

Learning in online scenarios has been a focus of research in the data mining field in recent years. In particular, the mining of so-called *data streams* has attracted a lot of attention [1, 2, 3]. Corresponding algorithms must be able to learn in an incremental way and to adapt to a changing environment. For example, clustering algorithms should be able to properly react to changes of the underlying clustering structure and, in particular, to a changing number of clusters [4]. Obviously, this hampers the application of standard clustering algorithms such as K -means and fuzzy C -means, as these algorithms assume the number of clusters to be a fixed (user-specified) parameter which is known in advance.

Several methods for determining an optimal number of clusters have been proposed in the literature (e.g., [5, 6]). Assuming a lower bound K_{min} and upper bound K_{max} to be given, the simplest approach is to try all potential numbers $K \in \{K_{min} \dots K_{max}\}$: A clustering structure is derived for every potential K and evaluated in terms of a quality measure such as the well-known Xie-Beni index [7]. Finally, the result which appears to be optimal according to this measure is adopted.

This approach as well as more sophisticated variants thereof are computationally quite complex, as they have to test every K independently. This drawback is further amplified by the problem of local optima, which is commonly encountered for clustering algorithms like fuzzy C -means: Running the algorithm twice will typically produce two different results. Thus, in order to evaluate a fixed cluster number K , it is not enough to apply the algorithm to the data once. Instead, it must be applied repeatedly, say, N times, in order to produce a reasonably good and confident approximation to the best quality that can be obtained for K . Consequently, the overall number of runs of the clustering algorithm amounts to $N \times (K_{max} - K_{min} + 1)$.

Needless to say, the above optimization strategy cannot be used in an online context where an optimal K has to be found repeatedly, and possibly under hard time constraints. In this paper, we propose a method for determining an optimal number of clusters in an *adaptive* and efficient way, focusing on fuzzy C -means as an underlying clustering algorithm. This method will be introduced in the next section, followed by an empirical evaluation in section 3.

2 Local Optimization of the Cluster Number

Recall that fuzzy C -means seeks to minimize the following objective function [8]:

$$J = \sum_{i=1}^n \sum_{j=1}^K \|x_i - c_j\|^2 (u_{ij})^m, \quad (1)$$

where $u_{ij} = u_j(x_i)$ is the membership of the i -th object x_i in the j -th cluster, and c_j is the j -th center. In the commonly employed *probabilistic* version of fuzzy C -means (FCM), it is assumed that

$$\sum_{j=1}^K u_{ij} = \sum_{j=1}^K u_j(x_i) = 1 \quad (2)$$

for all x_i [9]. The constant $m > 1$ in (1) is called the *fuzzifier* and controls the overlap (“smoothness”) of the clusters (a common choice is $m = 2$).

As mentioned before, the simple strategy outlined in the introduction is not practicable in an online setting as it requires the consideration of too large a number of candidate values and, hence, applications of the clustering algorithm. To minimize the latter, the idea of this paper is to pursue a *local* adaptation process that tries to adapt the cluster number K on the basis of a starting point K_0 in the style of a hill-climbing procedure. This strategy appears particularly appealing in an online setting in which the optimal cluster number is a function of time $K^*(t)$ ($t = 1, 2, \dots$): Assuming that the cluster number does not change abruptly over time, which means that $K^*(t+1) \approx K^*(t)$, the number $K_0 = K^*(t)$ will provide a good point of departure for finding $K^*(t+1)$.

Thus, starting with $K = K_0$, each iteration of our method consists of a test that checks whether the cluster model can be improved by increasing or decreasing K . To this end, we make use of a suitable quality measure (validity function) $Q(\cdot)$. Let $Q(K)$ denote the quality for the cluster number K , that is, for the cluster model obtained for this number. In each iteration, K is then updated as follows:

$$K \leftarrow \arg \max \{Q(K-1), Q(K), Q(K+1)\}$$

This is repeated until K remains unchanged, i.e., $Q(K) > \max \{Q(K-1), Q(K+1)\}$. Essentially, this approach requires two elements: Firstly, a suitable validity function $Q(\cdot)$, and secondly, a means for going from a clustering structure with K clusters to structures with $K-1$ and $K+1$ clusters, respectively. We shall discuss the first point in section 2.1 and the second one in section 2.2.

standard variant	fuzzy variant	complexity
$\max_k \max_{x,y \in C_k} \ x - y\ $	$\max_k \max_{i,j} u_{ik}^m u_{jk}^m \ x_i - x_j\ ^2$	$O(Kn^2)$
$\max_k \max_{x \in C_k} \ x - c_k\ $	$\max_k \max_i u_{ik}^m \ x_i - c_k\ ^2$	$O(Kn)$
$\sum_k \max_{x,y \in C_k} \ x - y\ $	$\sum_k \max_{i,j} u_{ik}^m u_{jk}^m \ x_i - x_j\ ^2$	$O(Kn^2)$

Table 1: Measures of intra-cluster variability. C_k denotes the k -th cluster and c_k its center.

standard (= fuzzy) variant	complexity
$\min_{k,\ell} \min_{x \in C_k, y \in C_\ell} \ x - y\ ^2$	$O(K^2n^2)$
$\sum_k \min_\ell \min_{x \in C_k, y \in C_\ell} \ x - y\ ^2$	$O(K^2n^2)$
$\sum_k \min_\ell \ c_k - c_\ell\ ^2$	$O(K^2)$
$\min_{k,\ell} \ c_k - c_\ell\ ^2$	$O(K^2)$

Table 2: Measures of intra-cluster variability. C_k denotes the k -th Cluster and c_k its center.

2.1 Fuzzy Validity Function

Regarding the evaluation of a cluster model in terms of a measure $Q(\cdot)$, several proposals can be found in the literature. Unfortunately, most of these measures have been developed for the non-fuzzy case. Indeed, validity functions of that kind might still be (and in fact often are) employed, namely by mapping a fuzzy cluster model to a crisp one first (i.e., assigning each object to the cluster in which it has the highest degree of membership) and deriving the measure for this latter structure afterwards. However, this approach can of course be criticized as it comes along with a considerable loss of information. On the other hand, many of the non-fuzzy measures can be adapted to the fuzzy case in a natural way.

Validity functions typically suggest finding a tradeoff between intra-cluster and inter-cluster variability (see tables 1 and 2, respectively, for some examples), which is of course a reasonable principle. Besides, our approach gives rise to a number of additional requirements, notably the following:

- (a) Since the number K of clusters is only changed locally by ± 1 , i.e., in the style of hill-climbing, our adaptation procedure might get stuck in local optima. Consequently, the convexity (resp. concavity) of the validity function is highly desirable. That is, $Q(K)$ should be maximal (resp. minimal) for the optimal number K^* of clusters and decrease (resp. increase) in a monotone way for smaller and larger values (at least within a certain range around K^*). Unfortunately, most existing measures do not have this property and instead show a rather irregular behavior (see Fig. 1 for a typical example).
- (b) As already explained above, to adapt the cluster number K , we provisionally consider two alternative structures that we obtain, respectively, by removing and adding a cluster. However, as will become clear later on, both candidate structures might not be fully optimized with regard to the FCM objective function (1). Instead, in an online setting, this optimization might take place only *after* the apparently best structure (cluster number) has been selected. In order to avoid this optimization to invalidate the previous selection, the validity measure $Q(\cdot)$ should well harmonize with the objective function (1).

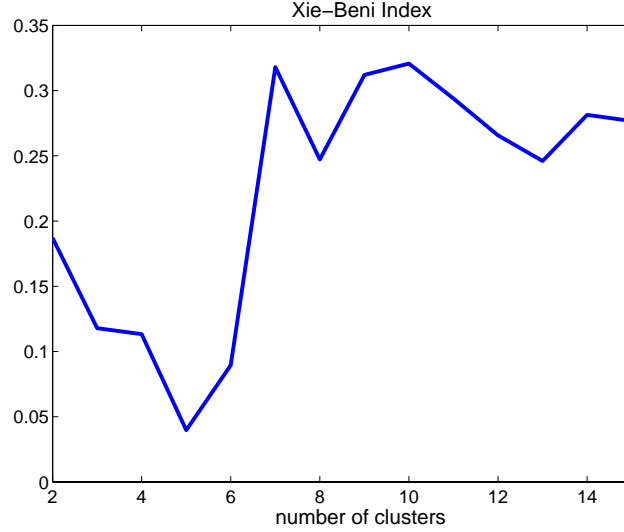


Figure 1: Example of the Xie-Beni validity measure.

- (c) Since the validity function is frequently evaluated in our application, its computation should be efficient. This disqualifies measures with a quadratic complexity, such as the maximal distance between two objects within a cluster.

A widely used validity function is the so-called Xie-Beni index, separation [7], which is defined as

$$\frac{\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K (u_{ik})^m \|x_i - c_k\|^2}{\min_{k,\ell} \|c_k - c_\ell\|^2}. \quad (3)$$

As most validity measures do, (3) puts the intra-cluster variability (numerator) in relation to the inter-cluster variability (denominator). In this case, the latter is simply determined by the minimal distance between two cluster centers. Obviously, the smaller the separation, the better the cluster model.

Since the nominator of (3) just corresponds to the objective function (1), the Xie-Beni index looks quite appealing with regard to point (b) above. Moreover, it is also efficient from a computational point of view. Still, point (a) remains problematic, mainly due to the minimum operator in the denominator.

To remedy this problem, we replace the minimum by a summation over all (pairwise) cluster dissimilarities, with smaller dissimilarities having a higher weight than larger ones. Simply defining the dissimilarity between two clusters by the distance between the corresponding centers is critical, however, since it neglects the variability (size) of these clusters. Therefore, we define the variability of a cluster in terms of the average (squared) distance from the center,

$$V_k \stackrel{\text{df}}{=} \frac{\sum_i u_{ik} \|x_i - c_k\|^2}{\sum_i u_{ik}}$$

and the dissimilarity between two clusters as

$$D(C_k, C_\ell) \stackrel{\text{df}}{=} \frac{\|c_k - c_\ell\|^2}{V_k \cdot V_\ell}.$$

These dissimilarities are aggregated by means of

$$\sum_{1 \leq k < \ell \leq K} \frac{1}{D(C_k, C_\ell)}, \quad (4)$$

thereby putting higher weight on smaller dissimilarities. Replacing the denominator in (3) by (4), we thus obtain

$$\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K \|x_i - c_k\|^2 (u_{ik})^m \cdot \sum_{1 \leq k < \ell \leq K} \frac{1}{D(C_k, C_\ell)}. \quad (5)$$

It is of course not possible to prove the concavity of (5) in a formal way. Still, our practical experience so far has shown that it satisfies our requirements in this regard very well and compares favorably with alternative measures. Corresponding experimental results will be presented in section 3 below.

Algorithm 2.1: Local optimization of the cluster number

Input: data, initial cluster number K

Output: optimal cluster number with associated partition

Data: partition P

initialize P by K cluster centers selected at random;

count $(\cdot) \leftarrow 0$;

best $\leftarrow P$;

repeat

repeat

 for every object, derive the membership degrees in the clusters in P ;

 derive the centers of the new clusters in P ;

until $\Delta J < \varepsilon$;

 find the best $K - 1$ partition $\rightarrow P_{-1}$;

 find the best $K + 1$ partition $\rightarrow P_{+1}$;

$P \leftarrow$ best partition among $\{P, P_{-1}, P_{+1}\}$;

$K \leftarrow$ number of clusters in P ;

 count $(K) \leftarrow$ count $(K) + 1$;

if P better than best **then** best $\leftarrow P$;

until K unchanged or count $(K) > K$;

return best;

2.2 Adapting the Clustering Structure

The second element of our adaptive optimization scheme is a means for replacing the current clustering structure, consisting of K clusters, by structures with $K - 1$ and $K + 1$ clusters, respectively. Intuitively, decreasing the cluster number by 1 means that one of the current clusters has disappeared. Thus, we propose to derive a candidate cluster model with $K - 1$ clusters as follows: One of the current clusters is tentatively removed, and M iterations of the fuzzy C -means algorithm are executed in order to re-assign the elements of this cluster to the remaining cluster centers (and to adjust these centers correspondingly). The quality of the cluster model thus obtained is then computed. This is repeated

K times, i.e., each of the current clusters is removed by way of trial. The best cluster model is then chosen, i.e., $Q(K - 1)$ is defined by the quality of the best model (see Algorithm 2.2).

Algorithm 2.2: Optimal reduction of cluster size

Input: T is partition with K clusters

Output: N is partition with $K - 1$ clusters

foreach Cluster c **do**

$S_c = T$ without cluster c ;

for $i = 1$ **to** M **do**

 for every object, derive the membership degrees in the $K - 1$ clusters in S_c ;

 derive the centers of the new clusters in S_c ;

end

 compute the validity measure for S_c ;

end

set $N =$ best partition according to validity measure S_c ;

Going from K to $K + 1$ assumes that an additional cluster has emerged. To create this cluster we complement the existing K centers by one center that is defined by a randomly chosen element. The probability of an element to be selected is reasonably defined as an increasing function of the elements's distance from its associated cluster center. More specifically, we define the probability of an element x_i by $p(x_i) \propto \sum_{k=1}^K \|x_i - c_k\| (u_{ik})^m$. In order to compute $Q(K + 1)$, we try out a fixed number of randomly chosen elements and select the one that gives the best cluster model.

3 Evaluation

This section presents first empirical evaluations consisting of two experimental studies. First, we investigate the suitability of our approach for determining the optimal cluster number in the static case. The second study is devoted to performance evaluations in a dynamic scenario.

3.1 Experiment 1

In the static case we compared our local optimization approach with the simple (global) strategy which tries every cluster number in the range $\{2 \dots 19\}$. In the latter case, we called the fuzzy C -means algorithm 10 times for each K in order to avoid local minima and then took the best result [10].

As test data, we used randomly generated synthetic data with rather simple clustering structures, each consisting of 1,000 data points located in the d -dimensional space $[0, 200]^d$. To generate a data set, we first determine the cluster number K at random, using a uniform distribution over $\{2 \dots 19\}$. The cluster centers are generated analogously (a distribution of the centers in $[0, 200]^d$ is accepted only if the distance between every pair of centers is at least 6 times as large as the standard deviation of the clusters). For every cluster, $1000/K$ data points are generated using a normal distribution with standard deviation 5.

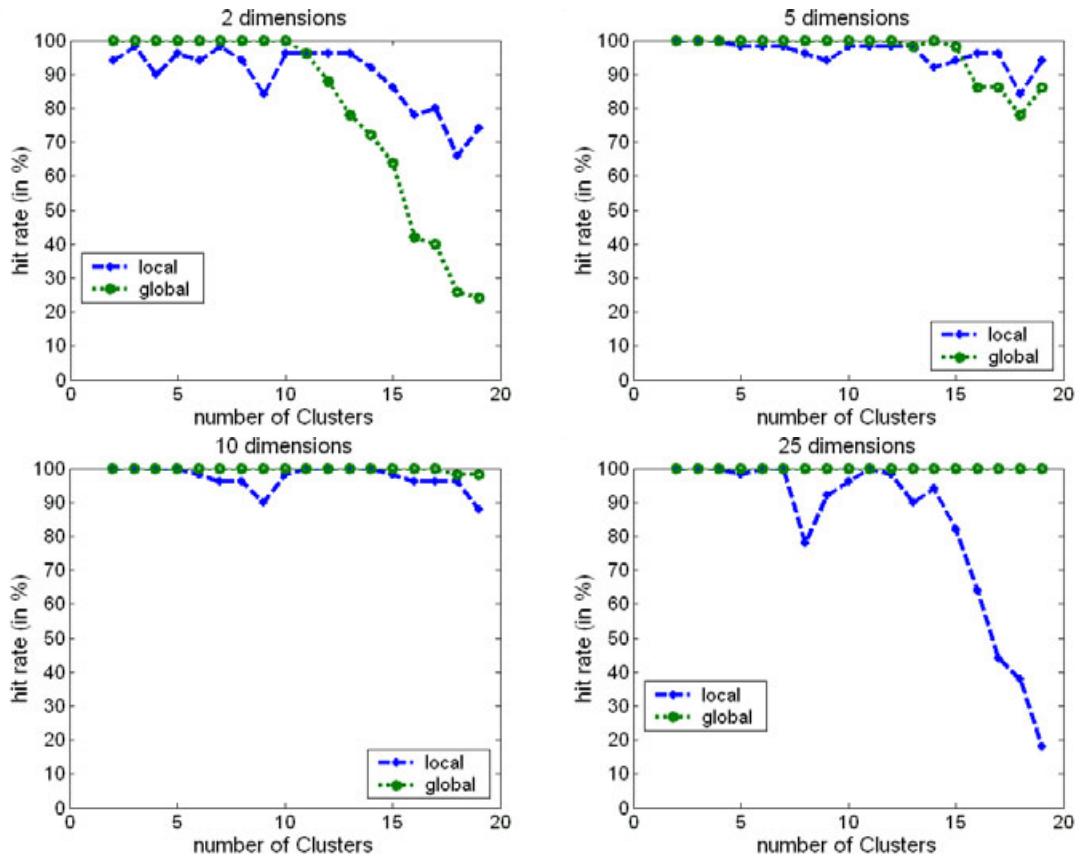


Figure 2: Comparison of the hit rates

50 data sets have been generated for each dimension $d \in \{2, 5, 10, 25\}$ and each cluster number K . Our local approach is initialized with $K_0 = 10$ while the global strategy searches the complete range $\{2 \dots 19\}$. To obtain a reference for the quality measure, we initialized the standard fuzzy C -means algorithm with the correct cluster centers and derived the validity index for the clustering structure thus obtained. For both approaches (local and global) we then checked whether the respective solution is at least as good as the reference result and derived a corresponding hit rate. Additionally, we measured the runtimes of both approaches.

Regarding the hit rate, it was to be expected that the global strategy outperforms the local one, as it tries all potential cluster numbers. However, as can be seen in Fig. 2, our local strategy is quite competitive and in some cases even superior. As a potential explanation, we suspect that our strategy of adding and removing clusters in a systematic way is more effective in escaping local minima of the fuzzy C -means objective function than restarting the algorithm at random. On the other hand, the local approach obviously runs into trouble if the true cluster number is much higher than the initialization, especially in the case of high-dimensional data.

With regard to the runtimes, our local approach clearly outperforms the global one (see Fig. 3). In fact, even though the runtimes slightly increase with the number of clusters, they remain on a comparably low level for the local approach, independently of the dimension. In contrast, the runtimes of the global method are much higher.

Fig. 4 shows a comparison with the Xie-Beni index. As expected, the results strongly depend on the initialization: the closer the true cluster number to the initialization $K_0 =$

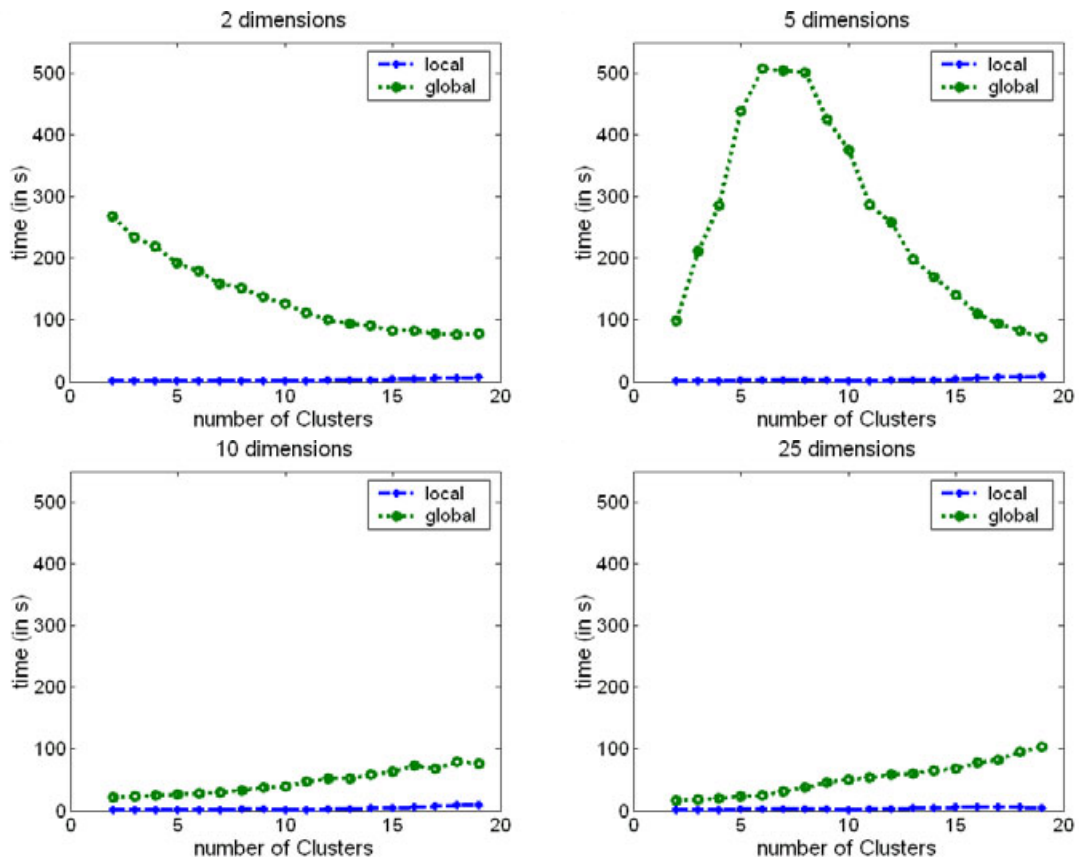


Figure 3: Comparison of the runtimes

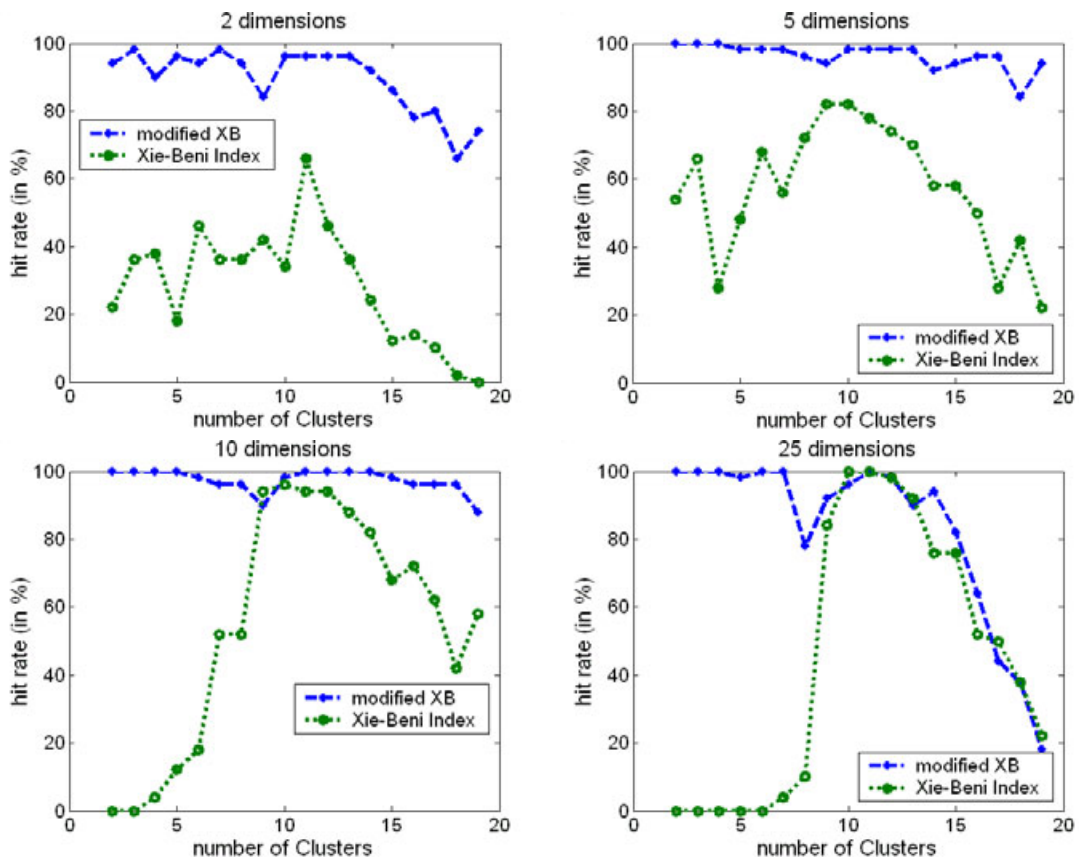


Figure 4: Comparison of the validity measures

10, the higher the hit rate becomes. More importantly, however, the figure clearly shows the positive influence of our modified version of the Xie-Beni index. In fact, using the measure (5) significantly improves the hit rate.

3.2 Experiment 2

In this experiment, we tested our method’s ability to discover the emergence of a new or disappearance of an existing cluster. To this end, we generated coupled data sets as follows: First, a single data set with $K \in \{2 \dots 18\}$ is generated as described above. Then, two variants of this data set are produced, one by generating additional data for the last cluster, and the other one by generating data for an additional (random) cluster center. Thus, the second variant contains $K + 1$ clusters, one more than the first variant.

To test the ability of detecting the increase of the cluster number, the first variant is clustered by initializing fuzzy C -means with the correct cluster centers. Then, we replaced the data set by the second variant and checked whether or not the additional cluster was discovered by the adaptation step outlined in section 2.2. To test the ability of detecting the decrease of the cluster number, the two variants of the data set are simply exchanged.

dimension	$K \rightarrow K - 1$	$K \rightarrow K + 1$	rate
2	100%	98.7%	99.3%
5	99.9%	98.9%	99.4%
10	100%	99.9%	100%
25	100%	99.9%	100%

Table 3: Detection rate

Table 3 shows average detection rates for 5,000 tests. As can be seen, the rates are close to 100%. As expected, the removal of a cluster is even somewhat easier to detect than the emergence of an additional cluster. In fact, in the latter case we only tried 10 data points as additional cluster centers, which makes it is quite likely that to overlook the new cluster. As in the first experimental study, the results seem to be even better for higher dimensions. This can be explained by the fact that local minima of the FCM objective function occur more frequently in the low-dimensional case.

4 Conclusions

We have introduced a local (adaptive) optimization scheme used to find the correct number of clusters in fuzzy C -means clustering, including a modified version of the Xie-Beni index for measuring the validity of a fuzzy partition. Our new approach seems to be especially competitive for low-dimensional data, though satisfactory results are also achieved for higher dimensions, given a reasonably good initialization. The experiments presented in this paper have furthermore shown that the modified validity measure is indeed preferable to the original Xie-Beni index in the context of a local search procedure.

Our adaptive optimization scheme is particularly motivated by online scenarios in which a potentially changing clustering structure must be maintained over time. The results obtained in the second experimental study indeed suggest that the approach is well-suited

for applications of this kind, at least if dramatic and abrupt changes of the clustering structure are unlikely to occur. More specifically, these results suggest that a change of the cluster number by ± 1 is detected with a high level of confidence. Admittedly, the scenario considered in this study was a quite simple one. However, first promising results in a more complex application, namely fuzzy clustering of parallel data streams, have already been obtained [11].

References

- [1] Domingos, P.; Hulten, G.: A General Framework for Mining Massive Data Streams. *Journal of Computational and Graphical Statistics* 12 (2003) 4, S. 945–949.
- [2] M. Garofalakis, J. G.; Rastogi, R.: Querying and mining data streams: you only get one look a tutorial. In: *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, S. 635–635. New York, NY, USA: ACM Press. ISBN 1-58113-497-5. 2002.
- [3] Golab, L.; Özsu, M.: Issues in data stream management. *SIGMOD Rec.* 32 (2003) 2, S. 5–14.
- [4] Beringer, J.; Hüllermeier, E.: Online clustering of parallel data streams. *Data Knowl. Eng.* 58 (2006) 2, S. 180–204.
- [5] Tibshirani, R.; Walther, G.; Hastie, T.: Estimating the number of clusters in a dataset via the gap statistic. *JRSSB* (2000).
- [6] Hamerly, G.; Elkan, C.: Learning the k in k-means. In: *NIPS*. 2003.
- [7] Xie, X. L.; Beni, G.: A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (1991) 8, S. 841–847.
- [8] Bezdek, J.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers. ISBN 0306406713. 1981.
- [9] F. Höppner, F. Klawonn, R. K.; Runkler, T.: *Fuzzy Cluster Analysis*. Wiley. 1999.
- [10] Bradley, P. S.; Fayyad, U. M.: Refining Initial Points for K-Means Clustering. In: *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, S. 91–99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. 1998.
- [11] Beringer, J.; Hüllermeier, E.: Online fuzzy clustering of parallel data streams. J. Valente de Oliveira and W. Pedrycz (eds.): *Advances in Fuzzy Clustering and Its Application*. John Wiley and Sons, 2007.

Zielpunktorientierte Kollisionsvermeidung für autonome mobile Systeme

Reinhold Warias und Michael Gerke

Prozeßsteuerung und Regelungstechnik

FernUniversität in Hagen

Universitätsstr. 27, 58097 Hagen

Reinhold.Warias@t-online.de, Michael.Gerke@FernUni-Hagen.de

1 Motivation und Zielsetzung

Intensive Forschungen und Fortschritte in der Technik haben autonome mobile Systeme (AMS) in den letzten Jahren in die Lage versetzt, mit Hilfe von Selbstlokalisierung, Weltmodellierung, Wegplanung und Kollisionsvermeidung innerhalb teilweiser oder komplett unbekannter Umgebungen zufriedenstellend zu navigieren. Unvorhersehbare, als dynamisch auftretende Hindernisse wirkende Objekte dürfen weder das AMS gefährden noch die Fahrmission zum Abbruch bringen. An dieser Stelle setzt die lokale Kollisionsvermeidung (LKV) an, die als eine der wichtigsten Komponenten mobiler Systeme direkt für die Sicherheit des AMS verantwortlich zeichnet. Sie ist zweckmäßig als on-board-Komponente realisiert, die unmittelbar auf vorhandene Sensorik zugreift.

Nach erfolgter Hindernis-Detektion initiiert die LKV über die Vorgabe von Soll-Daten an die Steuerung ein Ausweichmanöver, das ein Verlassen des vorgegebenen Weges bedingt. In Abhängigkeit von der Komplexität der Hindernisanordnung ist die Möglichkeit gegeben, dass das AMS zum geplanten Weg nicht zurückfindet. Es erscheint sinnvoll, die LKV mit einer zielgerichteten Komponente zu versehen, aufgrund derer ein auf dem geplanten Weg liegender Punkt, der Zielpunkt, aufgefunden werden kann.

Unser Ansatz besteht in einer Fuzzy-basierten LKV, die aufgrund direkter Sensorinformationen einen Lenkwinkel zur Vermeidung von Kollisionen ermittelt. Diesem Lenkwinkel überlagert der Controller einen korrigierenden Driftwinkel, der die Kursabweichung des AMS in Richtung des Zielpunkts berücksichtigt. Der Einfluss des Driftwinkels wird in Abhängigkeit von der jeweils aktuellen Hindernisdistanz mittels eines Wichtungsfaktors gesteuert.

Unsere Untersuchung der LKV gliederte sich in drei Stufen:

1. Aufbau des Fuzzy-Controllers
 - Anzahl der Zugehörigkeitsfunktionen
 - Überlappung der Zugehörigkeitsfunktionen ($\sum\mu = 1$; $\sum\mu > 1$)
2. Wahl der Ausweichstrategie
 - minimale Hindernisdistanz (1 Punkt bzw. 2 Punkte)
 - maximale Hindernisdistanz
 - Min-Max-Strategie (Kombination der ersten beiden Strategien)

jeweils ohne, im zweiten Durchlauf mit automatischer Modifizierung der Regelbasis

3. Rückkehrstrategie (Auffinden des ursprünglichen Wegs)
 - Driftvektor-Steuerung

Diese Strategien werden nachfolgend skizziert, ihre Versuchsabläufe und deren Ergebnisse sind ausführlich in [1] dokumentiert und diskutiert.

Zur Untersuchung unseres Ansatzes entwickelten wir eine unter Windows ablauffähige Simulations-Software "AmsSim, Version 1.0" mit Hilfe des Microsoft-Compilers "Visual C++". Sie verfügt über zahlreiche Möglichkeiten, Parameter von Umgebung, Sensorik und Fuzzy-Controller zu modifizieren.

2 Modellierung der Simulationskomponenten

2.1 Geländekarte

Die Geländekarte stellt das Modell des AMS-Arbeitsbereichs dar. Sie ist in der vorliegenden Software-Version V1.0 als eine die gesamte Hinderniskonfiguration enthaltende Bitmap angelegt, die dem AMS nicht bekannt ist; es bewegt sich in ihm vollkommen unbekannter Umgebung.

Zur Untersuchung des Fahrverhaltens bei unterschiedlichen Hinderniskonstellationen und –schwierigkeitsgraden sind insgesamt vier Geländekarten implementiert. Geländekarte 1 (Abb. 2.1) enthält eine einfache, aus Kreisen und Quadraten bestehende Hindernisumgebung.

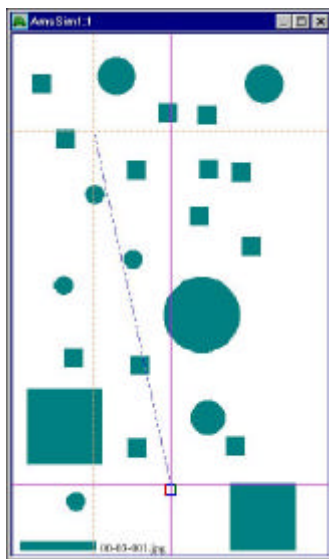


Abb. 2.1: Geländekarte 1

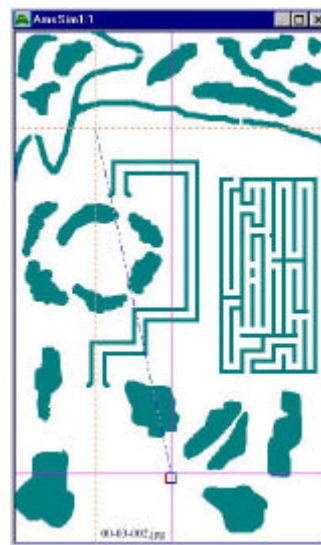


Abb. 2.2: Geländekarte 2

Geländekarte 2 (Abb. 2.2) ist mit Hindernissen unterschiedlichen Schwierigkeitsgrads ausgestattet. Die im oberen Bereich liegenden Zonen sind ausschließlich über sehr schmale Öffnungen erreichbar. Zusätzlich sind ein Labyrinth, ein gleichförmig breiter und ein kurzer, sich verengender Korridor verfügbar.

Die Nachbildung einer natürlichen Umgebung mit unregelmäßigen, willkürlich verteilten Hindernissen ist in Geländekarte 3 implementiert. Geländekarte 4 repräsentiert eine sehr anspruchsvolle Umgebung, die aus Korridoren und Labyrinthen besteht.

Eingeblendet in die Karten sind AMS, Weltkoordinatensystem, Fadenkreuz des Zielpunkts sowie der das AMS und Zielpunkt verbindende Driftvektor. Ursprung der

Weltkoordinaten und Fadenkreuz des Zielpunkts können wahlfrei mit Hilfe eines Umgebungs-Editors positioniert werden. Die Darstellung dieser Komponenten in Bezug auf Farbe und Linienstil ist ebenso modifizierbar wie die des Driftvektors. Zusätzlich besteht die Möglichkeit, jede der Komponenten ein- bzw. auszublenden.

2.2 Autonomes mobiles System

In der vorliegenden Software-Version V1.0 ist ein abstraktes, aus einem Punkt (1 Pixel) bestehendes AMS, das über die entsprechende Sensorik verfügt, implementiert. Der Punkt ist masselos und verfügt über keinerlei kinematische Eigenschaften. Aufgrund einer Lenkwinkelbegrenzung von $\pm 45^\circ$, die bei Ausweichmanövern eine Kurvenfahrt erzwingt, stellt sich das AMS dennoch als nicht-holonome Komponente dar.

Aus Gründen der besseren Sichtbarkeit ist das AMS als Rechteck mit am Bug befindlichem Hotspot dargestellt (Abb. 2.3). Die Orientierung ist durch die grüne Steuerbordseite und die rote Backbordseite verdeutlicht. Das Rechteck löst keine Kollisionen aus; diese werden hier ausschließlich durch den Hotspot verursacht.

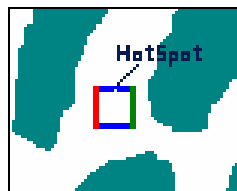


Abb. 2.3: mobiles System

Ein AMS-Editor bietet die Möglichkeit, nicht nur Startposition und Start-Orientierung, sondern auch die Farben des Hotspots und der AMS-Bordkanten sowie deren Rahmenstärke festzulegen. Die Darstellung des AMS innerhalb der jeweils aktuellen Geländekarte gleicht sich den im AMS-Editor vorgegebenen Daten an; Verhaltensänderungen treten, auch bei Modifizierung der AMS-Abmessungen, nicht auf.

2.3 Steuerpult des AMS

Die Steuerung des AMS erfolgt über ein Steuerpult (Abb. 2.4), das nicht nur die Möglichkeit bietet, manuellen oder Automatikbetrieb vorzugeben, sondern auch Einzelschrittbetrieb auszuführen und Haltepunkte zu setzen.

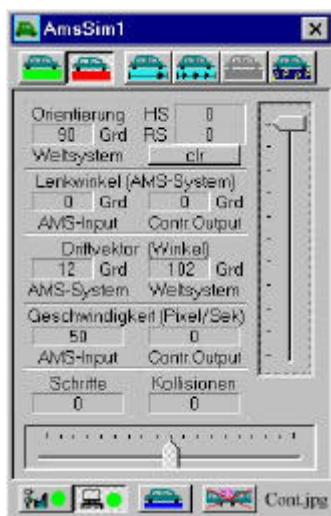


Abb. 2.4: "Steuerpult" des AMS

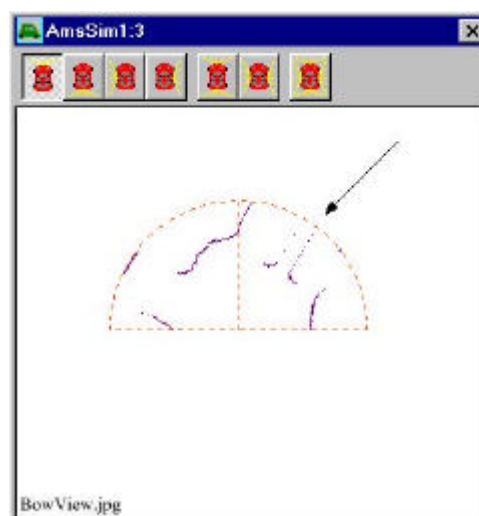


Abb. 2.5: Wahrnehmung der Umgebung durch das AMS

Zusätzlich liefert es Informationen über den AMS-Zustand, wie z.B. Orientierung und Driftwinkel. Weiterhin gibt es nicht nur die aktuelle Schritt-Nummer und die Anzahl der erfolgten Kollisionen an, sondern auch die Schritt-Nummer (HS) des Haltepunkts sowie die Anzahl (RS) der bis zum Haltepunkt noch zurückzulegenden Restschritte an.

Das "AMS-Gesichtsfeld" stellt die Umgebung in dem seitens des AMS wahrgenommenen Stil dar. Simuliert wurde in der vorliegenden Software ein Laser-Scanner mit variablem Erfassungsbereich. In Abb. 2.5 bewegt sich das AMS in Geländekarte 2 (vgl. Abb. 2.2), sieht auf der rechten Seite den Eingang des Korridors und steuert auf ein Hindernis zu.

2.4 Sensorik

Die modellierte Sensorik besteht aus Laser-Scannern, die den Erfassungsbereich von 190° in 528 Sektoren $\hat{=} 0,36^\circ$ unterteilen. In unserem Ansatz wurden ausschließlich Sektoren im Bereich 15...514 verwendet. Der Meßbereich ist im Intervall $l \in [1, 500]$ einstellbar.

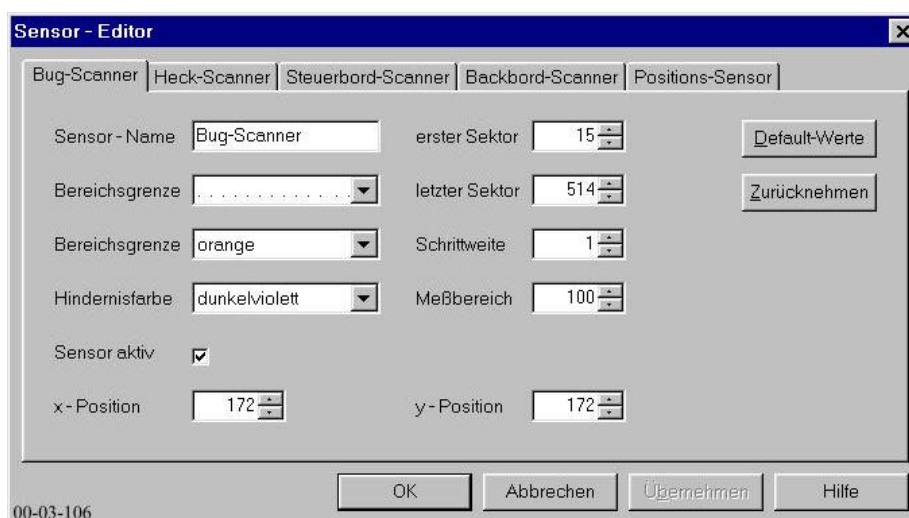


Abb. 2.6: Parameter der Laser-Scanner

Abb. 2.6 zeigt die Parametrierungsmöglichkeiten der Laser-Scanner. Neben Meßbereich, Schrittweite sowie erstem und letztem Sektor können nicht nur Farbe und Stil der Bereichsgrenzen, sondern auch die Farbe der Hindernisdarstellung im AMS-Gesichtsfeld (Abb. 2.5) bestimmt werden.

Heck-, Steuerbord- und Backbordscanner sind in dieser Software-Version implementiert und liefern Meßwerte, wirken jedoch noch nicht auf die Steuerung ein.

2.5 Fuzzy-Controller

Der Fuzzy-Controller besteht aus den die Zugehörigkeitsfunktionen beinhaltenden Fuzzy-Sets, dem Regelwerk, der Inferenzmaschine und der Defuzzifizierung.

Der Fuzzy-Set-Editor kann Daten, die den gesamten Set betreffen (Abb. 2.7) wie z.B. die Darstellung des Crisp-Werts oder einen Set-Namen, modifizieren. Ferner bietet er die Möglichkeit, die Zugehörigkeitsfunktionen zu parametrieren (Abb. 2.8). Festgelegt ist die Trapezform mit ihren Endformen "Dreieck" und "Rechteck".

Zusätzlich kann zur Laufzeit des Versuchs eine Ein- oder Ausgangsvariable des Controllers eingeblendet, deren Fuzzy-Set dargestellt und (in Abb. 2.9 als violette Senkrechte) ihr aktueller Crisp-Wert angezeigt werden.

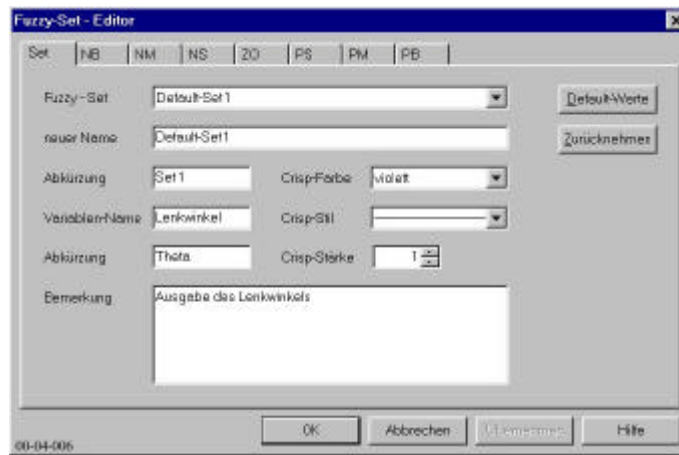


Abb. 2.7: Daten des Fuzzy-Sets

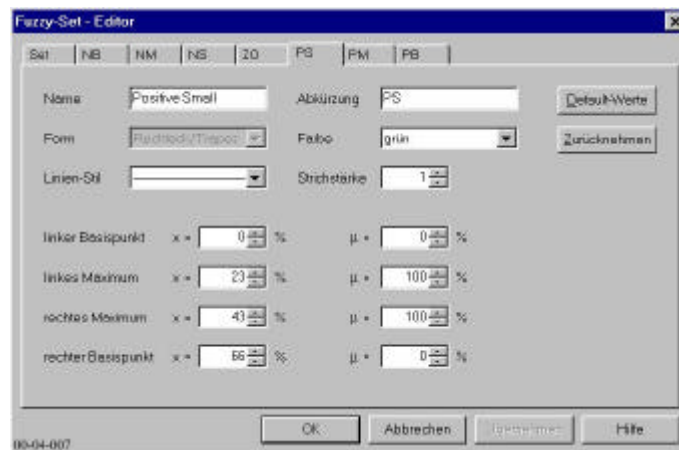


Abb. 2.8: Daten einer Zugehörigkeitsfunktion

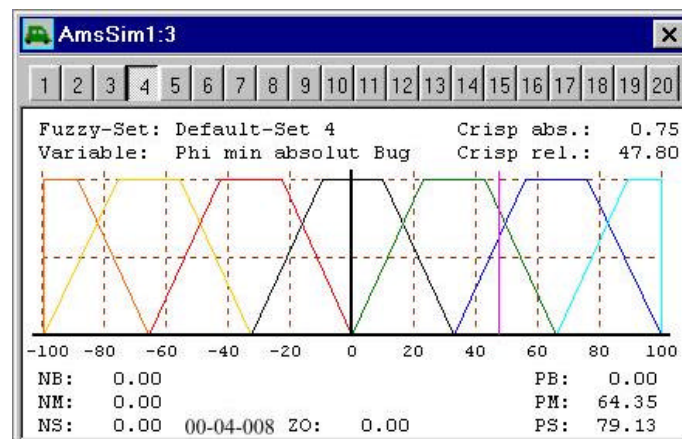


Abb. 2.9: Fuzzy-Information zur AMS-Laufzeit

Die numerische Angabe der fuzzifizierten Anteile der Zugehörigkeitsfunktionen ist in diesem Fenster ebenso selbstverständlich wie die Ausgabe des Crisp-Wertes als Zahl sowohl absolut als auch relativ.

Die Methoden der Inferenzmaschine können mit dem Inferenz-/Defuzzy-Editor (Abb. 2.10) eingestellt werden. In der vorliegenden Software-Version wirkt sich diese Einstellung ausschließlich auf die Operatoren der Aggregation aus. Die Operatoren der Akkumulation lassen sich zwar auch einstellen, sind zur Zeit jedoch unwirksam; die entsprechende Erweiterung bleibt späteren Software-Versionen vorbehalten.

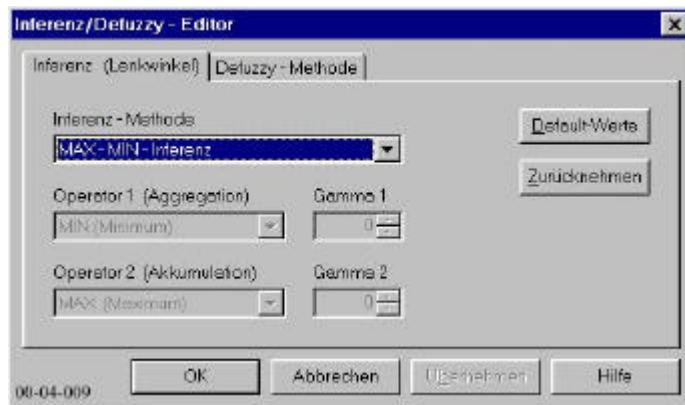


Abb. 2.10: Parameter der Inferenzmaschine

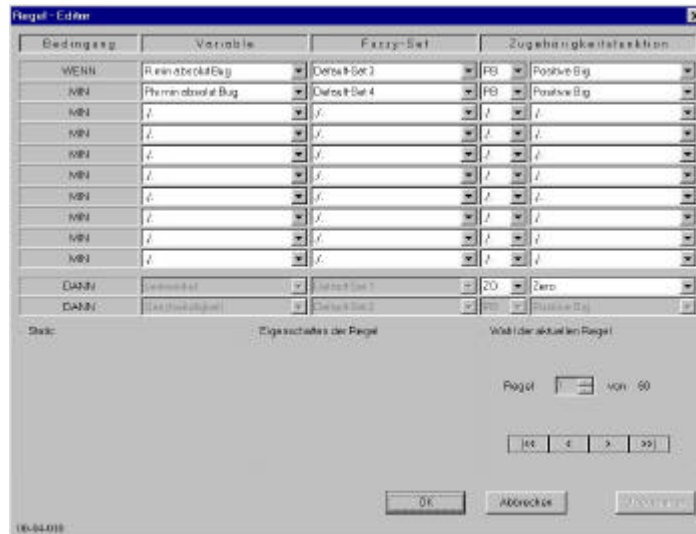


Abb. 2.11: Regel-Editor zur Modifizierung des Regelwerks

Die Anfangs-Regelwerke diverser Strategien sind in der Software als Default-Regelbasen hinterlegt. Zur Optimierung der Strategien können diese Basen mit Hilfe des Regel-Editors (Abb. 2.11) modifiziert werden.

3 Controller und Strategien

3.1 Aufbau des Fuzzy-Controllers

Die Zugehörigkeitsfunktionen des Controllers werden mit dem Fuzzy-Set-Editor (Abb. 2.8) modifiziert. Regelbasen, Inferenzmaschine und Defuzzifizierung setzen in der vorliegenden Implementierung entweder $\mu = 0$ oder $\mu = 1$ voraus; die Verarbeitung von Zwischenwerten sei späteren Versionen vorbehalten. Der Editor rundet den Zugehörigkeitsgrad entsprechend auf oder ab.

Die Eingangswerte der Sensorik sind normiert (siehe auch Abb. 2.6): der Winkel auf $\pm\pi/2$, der Hindernisabstand auf die Sensor-Reichweite r_{\max} .

Für jede in der Regelbasis enthaltene Regel wird die Aggregation entsprechend dem Operator der eingestellten Inferenz-Methode durchgeführt und liefert den Erfüllungsgrad der aktuellen Regel, der mit der zugeordneten Zugehörigkeitsfunktion des Ausgangs-Fuzzy-Sets verknüpft wird. Das Ergebnis besteht in den vier charakteristischen Punkten der Ausgangs-Membership. Daraus ermittelt sich als Inferenz-Ergebnis sowohl die Fläche der Ausgangszugehörigkeit als auch die

x-Komponente ihres Schwerpunkts. Die Inferenzmaschine liefert als Ergebnis bei n Regeln auch n Erfüllungsgrade und n Schwerpunkte der Ausgangsflächen.

Die Defuzzifizierung ermittelt aus diesen Daten den Crisp-Wert des Lenkwinkels mit Hilfe einer modifizierten Center-of-Gravity-Methode. Der Crisp-Wert x_c der Stellgröße errechnet sich aus den Erfüllungsgraden μ_v und den Schwerpunktkoordinaten x_v der n Regeln zu:

$$x_c = \frac{\sum_{n=1}^n x_n m_n}{\sum_{n=1}^n m_n}$$

Die Methode weist den Nachteil auf, dass sich überlappende Flächen mehrfach in den Crisp-Wert eingehen, bietet jedoch den gravierenden Vorteil der einfacheren Implementierung und der geringeren Rechenzeit.

3.2 Strategien der minimalen Distanz

Die Strategie der minimalen Distanz wertet die Lage des Punktes aus, der die geringste Distanz zum AMS aufweist.

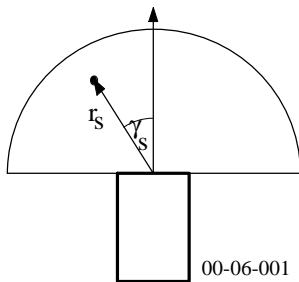


Abb. 3.1: minimale Distanz eines Punktes

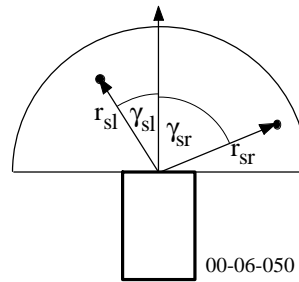


Abb. 3.2: minimale Distanz zweier Punkte

Aus dem im Bereich $-90^\circ < \gamma_s < +90^\circ$ liegenden Spektrum aller Hindernisdistanzen wird der Punkt gewählt, der dem AMS der nächste ist. Dieser Punkt (Abb. 3.1) ist durch den Betrag r_s und den Winkel γ_s in Sensorkoordinaten eindeutig festgelegt. r_s und γ_s werden fuzzifiziert und der Steuerung zugeführt.

Eine Erweiterung liegt in der Einbeziehung eines zweiten Punktes: die minimalen Hindernisdistanzen werden gemäß Abb. 3.2 im ersten und vierten Quadranten der Sensor-Koordinaten ermittelt und auf die Steuerung geschaltet.

Da die vollständige Regelbasis mit 256 Regeln sehr umfangreich und unübersichtlich ist, besteht unser Ansatz darin, die erste Hälfte der Regelbasis ausschließlich mit r_{sl} und γ_{sl} , deren zweite Hälfte ausschließlich mit r_{sr} und γ_{sr} aufzubauen.

Die vollständige Regelbasis umfaßt 32 Regeln und eine Zusatzregel. Das bedeutet eine Reduzierung der Anzahl der Regeln von 256 auf 33. Durch die Inferenzmaschine wird der Einfluss aller Eingangsvariablen sichergestellt.

Die Zusatzregel erzeugt eine Unsymmetrie, die eine Kollision in dem Fall verhindert, in dem das AMS senkrecht auf eine gerade Wand zufährt. Die Kollision ist aufgrund der symmetrischen Regelbasis ansonsten nicht zu verhindern.

3.3 Strategie der maximalen Distanz

Diese Strategie wertet die Lage des Punktes aus, der die größte Distanz zum AMS aufweist.

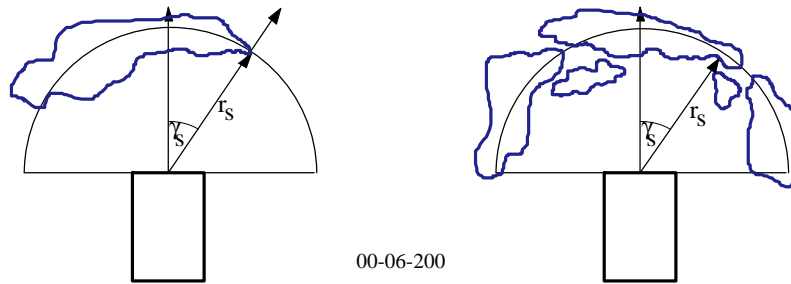


Abb. 3.3: Punkte maximaler Hindernisdistanz bei unterschiedlichen Konstellationen

In Abb. 3.3 (rechts) bewegt sich das AMS auf eine Hinderniskonfiguration zu und detektiert die Abstände zu den Hinderniskonturen. Der durch r_s und γ_s festgelegte Punkt ist derjenige von allen Punkten, der den größten Abstand zum AMS hat. Bewegt sich das mobile System auf ein einzelnes Hindernis zu (Abb. 3.3, links), detektiert es neben der rechten Hindernisbegrenzung keine weiteren Objekte. In diesem Fall wird der Betrag des Radiusvektors r_s gleich dem maximalen Meßbereich des Scanners gesetzt.

Eine Kollision kann bei senkrechter Fahrt auf eine gerade Wand nicht erfolgen. Weiterhin ergibt sich mit vier Zugehörigkeitsfunktionen für den Betrag des Radiusvektors r_s und sieben Memberships für den Winkel γ_s eine mit 28 Regeln übersichtliche Regelbasis.

3.4 Min-Max-Strategie

Die Min-Max-Strategie ist eine Hybridstrategie, die die "minimale Distanz (1 Punkt)" mit der "maximalen Distanz" kombiniert.

Die "minimale Distanz" mit dynamischer Regelbasis zeigt sehr gute Ergebnisse im Bereich der Kollisionsvermeidung, über Fernsichteigenschaften verfügt sie nicht. Durch Kombination mit der "maximalen Hindernisdistanz" könnte sich eine Strategie entwickeln lassen, die nicht nur gute Kollisionsvermeidungseigenschaften aufweist, sondern zusätzlich "vorausschauend" steuert.

3.5 Zielpunktorientierte Strategien

Das Ziel dieser Strategien besteht darin, das AMS einen vorgegebenen Punkt ansteuern und erreichen zu lassen. Zur Anpeilung dieses Zielpunkts wird ein Driftvektor \underline{r}_{DW} definiert, der sich aus dem Zielvektor \underline{r}_Z und dem Positionsvektor \underline{r}_W ermittelt (Abb. 3.4).

Der Zielvektor beginnt im Startpunkt und endet im Zielpunkt. Der Positionsvektor führt vom Startpunkt zum Hotspot des AMS. Die Differenz dieser Vektoren stellt den im Hotspot beginnenden und im Zielpunkt endenden Driftvektor dar.

In unserem Ansatz ermittelt der Fuzzy-Controller den Lenkwinkel Θ_C , der auf AMS-Koordinaten basiert. Dem AMS sind sowohl Zielpunkt als auch aktuelle eigene Position bekannt; es ist somit in der Lage, die Richtung des Driftvektors, also den Driftwinkel φ_{DW} zu bestimmen, der auf Weltkoordinaten (Abb. 3.5) basiert.

Mit der Beziehung $\varphi_{DA} = \varphi_{DW} - \Psi$ wird der Driftwinkel in das AMS-Koordinatensystem transformiert. Der Betrag des Driftvektors ist für die weiteren Überlegungen irrelevant.

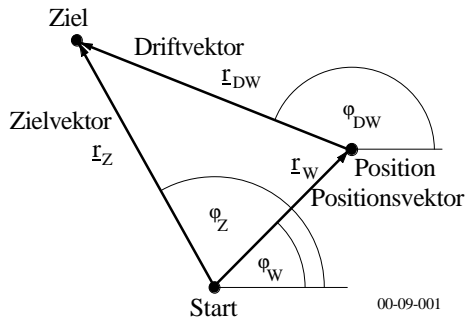


Abb. 3.4: Definition des Driftvektors

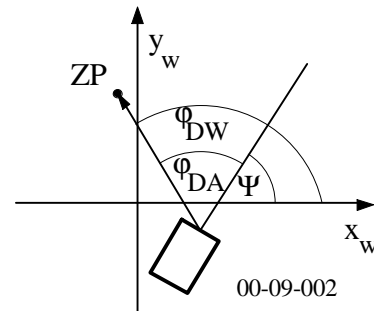


Abb. 3.5: Driftwinkel des AMS

Abb. 3.5 ist zu entnehmen, dass φ_{DA} beliebige Werte im Intervall $\varphi_{DA} \in [-180^\circ, +180^\circ]$ annehmen kann, während der Lenkwinkel Θ des mobilen Systems auf das Intervall $\Theta \in [-45^\circ, +45^\circ]$ begrenzt ist. Unser Ansatz besteht darin, den Driftwinkel bis zu einem Grenzwert von $\pm 45^\circ$ zu übernehmen, außerhalb dieses Intervalls gemäß Abb. 3.6 auf einen Maximalwert zu begrenzen.

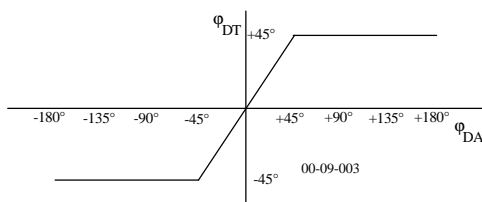


Abb. 3.6: Begrenzung des Driftwinkels

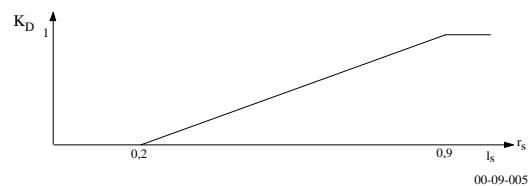


Abb. 3.7: Verlauf der Wichtung K_D

Damit ergibt sich der einzustellende Lenkwinkel Θ aus dem Ausgangswert Θ_C des Controllers und dem transformierten Driftwinkel φ_{DT} zu $\Theta = \Theta_C + K_D \cdot \varphi_{DT}$. Die Wichtung $K_D \in [0,0, 1,0]$ wurde eingeführt, um den Einfluss des Driftwinkels zu relativieren, d.h. die Höhe der Wichtung wird während der Laufzeit festgelegt. Bei vollkommener Hindernisfreiheit gilt $K_D = 1$, in unmittelbarer Hindernisnähe muß $K_D = 0$ betragen. Daraus ergibt sich die Eignung des Abstands r_s zum Hindernis als Kriterium, das auf die Sensorreichweite l_s normiert wird. Abb. 3.7 zeigt den Verlauf des Wichtungsfaktors unseres Ansatzes als Funktion der Hindernisdistanz.

4 Versuchsablauf und Ergebnisse

Zur Durchführung der Versuche wurde der Hauptbildschirm (Abb. 4.1) der Software derart eingerichtet, dass sämtliche benötigten Daten zur Verfügung standen:

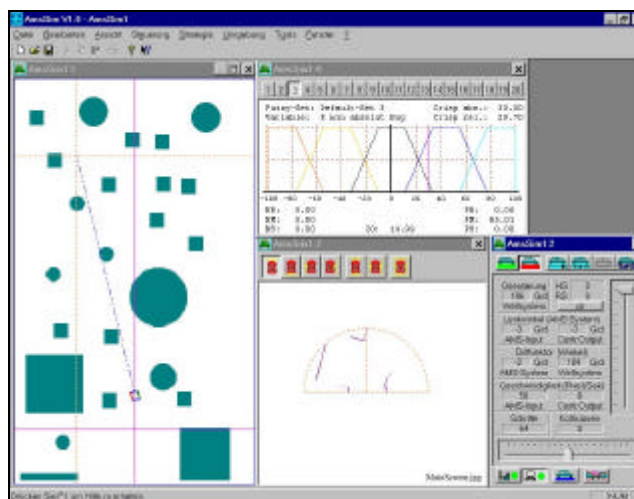


Abb. 4.1: Hauptbildschirm der Anwendung "AmsSim" bei Durchführung der Simulationen

4.1 Analyse der Fuzzy-Sets und Zugehörigkeitsfunktionen

Die Untersuchung der Fuzzy-Sets im Rahmen der ersten Versuchsreihen zeigte, dass ein Fuzzy-Controller mit sieben Zugehörigkeitsfunktionen je Set nicht optimal arbeitet. Aufgrund der steilen Flanken verursacht eine kleine Änderung der Eingangsgröße bereits eine große Änderung der Ausgangsgröße.

Auch sich überlappende Zugehörigkeitsfunktionen ($\sum \mu > 1$) beeinflussen die Steuerung des AMS nachteilig. Auch in diesem Fall scheinen einfließende Nichtlinearitäten eine große Änderung der Ausgangsgröße zu initiieren.

Bei großer Sensor-Reichweite stellte sich ein derart aufgebauter Fuzzy-Controller als nicht geeignet heraus.

Prinzipiell konnten die Versuchsergebnisse durch Reduzierung der Sensor-Reichweite verbessert werden. Da der jeweilige Hindernisabstand auf die Reichweite r_{\max} normiert wurde, entspricht die Verringerung von r_{\max} einer Reduzierung der Hindernisdichte.

Eine (in unserem Ansatz noch nicht untersuchte Möglichkeit) besteht in der Normierung auf einen festen Wert, wobei durch eine Grenzwertbildung sichergestellt werden muß, dass x_c keine Werte über 100% annehmen kann.

4.2 Wirkung der Strategie der minimalen Distanz (1 Punkt)

Der Ursprung des Weltsystems wurde vor das Labyrinth der Geländekarte 2 gelegt. Mit statischer Regelbasis, geringer Sensor-Reichweite und 7 Zugehörigkeitsfunktionen je Set fuhr das AMS in das Labyrinth, erkannte die Rechtskurve nicht und vollzog an dieser Stelle eine Linkswendung (Abb. 4.2).

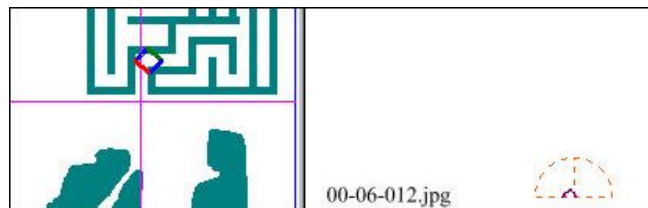


Abb. 4.2: Linkswendung in der Rechtskurve

Anschließend verließ es das Labyrinth durch dessen Eingang. Bei einer Mission von 68 Schritten erfolgten 2 Kollisionen.

Die Reduzierung der Zugehörigkeitsfunktionen auf 5 je Set ergab, dass das AMS zwar nach wie vor Schwierigkeiten hatte, Kurven zu detektieren, das Labyrinth jedoch nach 377 Schritten ohne Kollisionen verließ.

Mit 5 Zugehörigkeitsfunktionen je Set und Dynamisierung der Regelbasis fuhr das AMS in den Eingang, verursachte einige Kollisionen, bewegte sich aber unter starken Oszillationen freizügig innerhalb des Labyrinths. Gemäß Abb. 4.3 wurde das Ende der Sackgasse erkannt, und unter zweimaligem Kreisen sowie einer Kollision fand die Wendung mit anschließender Aufwärtsfahrt statt.

Die Mission umfaßte 2030 Schritte und löste 17 Kollisionen aus. Die Umstellung auf hohe Sensor-Reichweite ergab bei einer Missionsdauer von 2950 Schritten eine Anzahl von 42 Kollisionen.

Das AMS bewältigte mit dieser Steuerung sowohl den Korridor als auch das Labyrinth der Geländekarte 2 sogar mit hoher Sensor-Reichweite. Der Fuzzy-Controller mit 5 Zugehörigkeitsfunktionen je Set und dynamischer Regelbasis ist somit für die Strategie der minimalen Distanz (1 Punkt) als optimal einzustufen. Auftretende

Oszillationen sind ebenso eine Eigenschaft dieser Strategie wie die Kollisionen in einem spitzwinklig verlaufenden Korridor. Hier wären zusätzliche Algorithmen notwendig, die in unserem Ansatz nicht untersucht wurden.

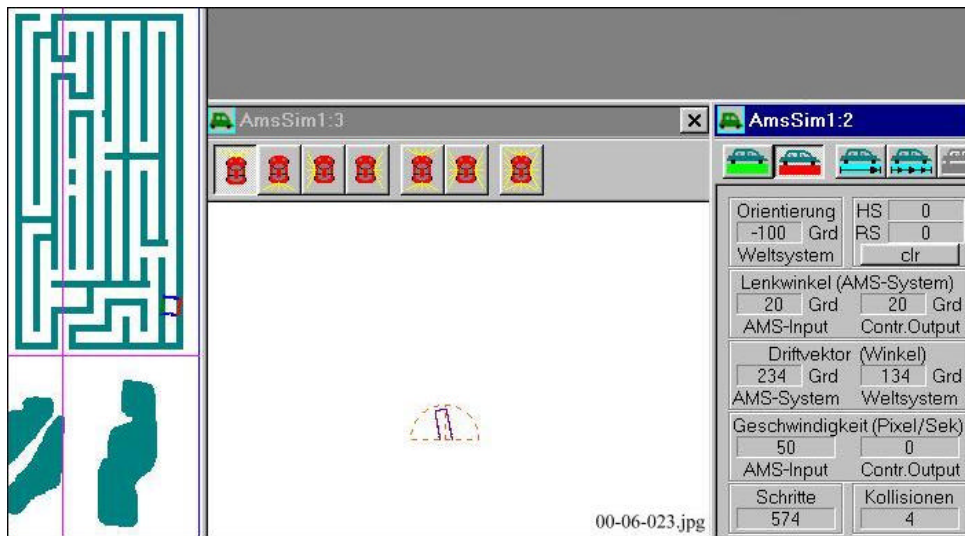


Abb. 4.3: Bewältigung des Labyrinths

4.3 Untersuchung der Strategie der minimalen Distanz (2 Punkte)

Auch hier zeigte sich, daß der Controller mit 5 Zugehörigkeitsfunktionen je Set das bessere Fahrverhalten mit sehr geringer Oszillationsneigung aufwies.

Der Nachteil dieser Strategie liegt darin, daß auch eine um Regeln zur Detektion von Ecken erweiterte Regelbasis deutliche Schwierigkeiten zeigte, Kurven zu erkennen. Die Ergebnisse in dieser Hinsicht waren derart schlecht, dass das Labyrinth nicht erprobt werden konnte.

Die Strategie der minimalen Distanz (2 Punkte) ist mit akzeptablem Umfang des Regelwerks nicht in der Lage, ein mobiles System sicher zu steuern.

4.4 Analyse der Strategie der maximalen Distanz

Der mit 5 Zugehörigkeitsfunktionen je Set arbeitende Controller steuerte das AMS bei hoher Sensor-Reichweite zunächst entlang der Ordinate und ließ es den großen Kreis (Abb. 4.5) unter Auslösung von 4 Kollisionen passieren.

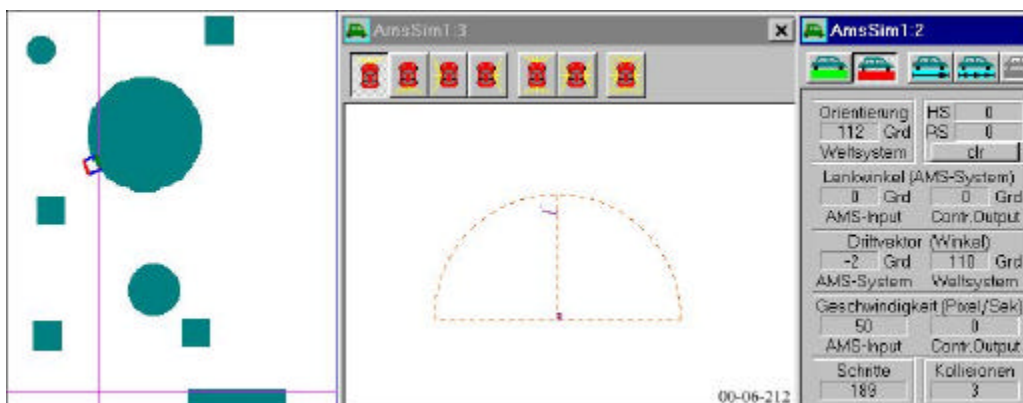


Abb. 4.5: Kollisionen am großen Kreis

Es zeigte sich, dass der Nachteil dieser Strategie offensichtlich in einer asymptotischen Annäherung an ein Objekt liegt, die zum "Schleifen" an den Hinderniskonturen führt;

hier liegt eine Eigenschaft der "maximalen Distanz" vor. Das Abdrehen vor Objekten erfolgte oft derart spät, dass es in manchen Fällen unter einer 360°-Drehung innerhalb des Hindernisses stattfand.

Die Strategie der maximalen Distanz ist daher allein nicht in der Lage, ein mobiles System sicher zu steuern.

4.5 Untersuchung der Min-Max-Strategie

Bei hoher Sensor-Reichweite zeigte das mobile System ein im Vergleich zur "minimalen Distanz (1 Punkt)" etwas schlechteres Fahrverhalten. Eine gewisse Verbesserung konnte durch Modifizierung des r_{\min} zugeordneten Fuzzy-Sets erzielt werden, d.h. der Einfluß der "maximalen Hindernisdistanz" wurde erhöht.

Bei geringer Sensor-Reichweite schien das AMS nach subjektivem optischen Eindruck über ein geringes (aber doch vorhandenes) Maß an Vorausschau zu verfügen. Kollisionen mit dem "Rand der Welt" traten nicht mehr auf.

Die Min-Max-Strategie ist in der vorliegenden Form in der Lage, ein mobiles System sicher zu steuern. Die vorhandenen Schwächen sind mit großer Wahrscheinlichkeit als Optimierungsproblem mit Modifizierungen von Regelwerk und Fuzzy-Sets zu lösen.

4.6 Wirkung der zielpunktorientierten Strategie

Mit 5 Zugehörigkeitsfunktionen je Set, einer dynamischen Regelbasis und hoher Sensor-Reichweite steuerte das AMS in den ersten drei Schritten zunächst vom kleinen Quadrat gemäß Abb. 4.5 fort. Nach 17 Schritten wirkte der Driftvektor gemäß Abb. 4.6 auf die Steuerung und führte das AMS nach links. Der Einfluß des Driftvektors ließ das AMS zwischen kleinem Quadrat und großem Kreis (Abb. 4.7) hindurchfahren.

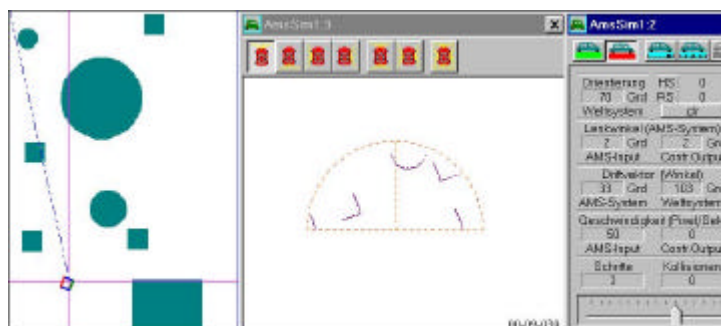


Abb. 4.5: Abdrift vom linken kleinen Quadrat

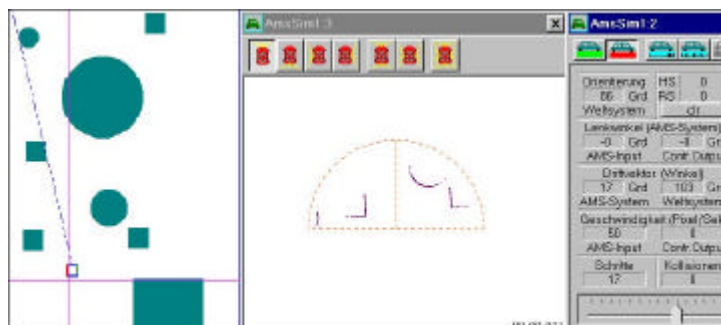


Abb. 4.6: Wirkung des Driftvektors

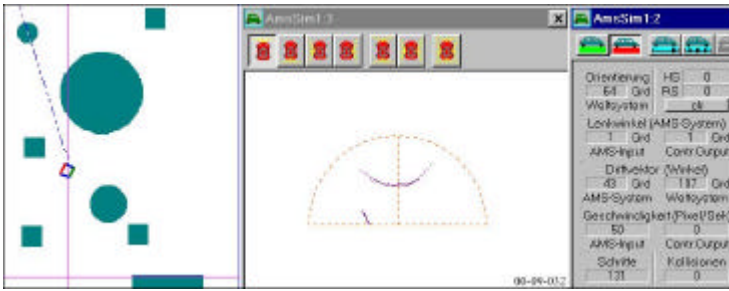


Abb. 4.7: Ansatz zur Fahrt zwischen Quadrat und Kreis



Abb. 4.8: Driftvektor - Wirkung

Der Driftvektor zog das AMS "wie ein Gummiband" zum Zielpunkt (Abb. 4.8). Da der kleine Kreis den Weg versperrte, suchte das AMS einen Umweg und steuerte nach 748 Schritten (Abb. 4.9) auf den Zielpunkt zu und fuhr in diesem Fall an ihm vorbei (Abb. 4.10).

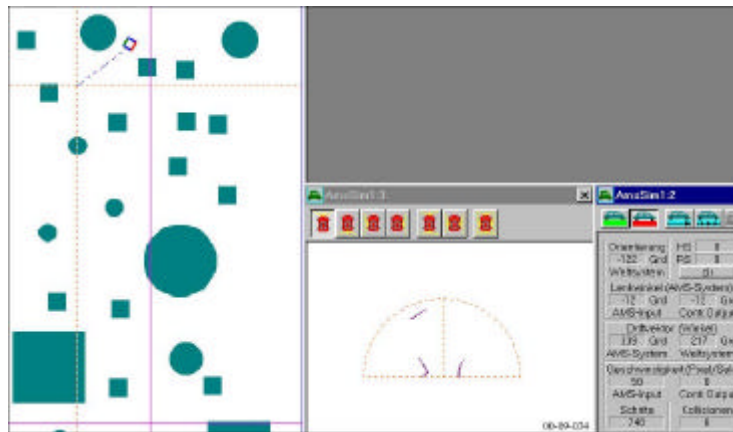


Abb. 4.9: Kurs auf den Zielpunkt

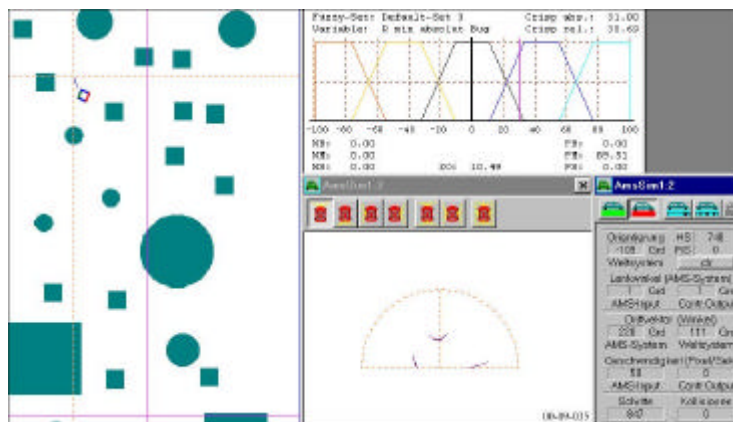


Abb. 4.10: Fahrt am Zielpunkt vorbei

Ursache war das in unmittelbarer Nähe des Zielpunkts angeordnete kleine Quadrat, durch dessen Einwirkung die Kollisionsvermeidung die Steuerung wieder allein übernahm. Das mobile System zeigte eindeutig zielgerichtetes Verhalten; dabei fuhr es kollisionsfrei.

Bei Reduzierung auf geringe Sensor-Reichweite geht das mobile System sofort, über den Driftvektor gesteuert, auf Zielkurs. Es weicht gemäß Abb. 4.11 dem kleinen Kreis aus und trifft nach 451 Schritten auf den Zielpunkt (Abb. 4.12).

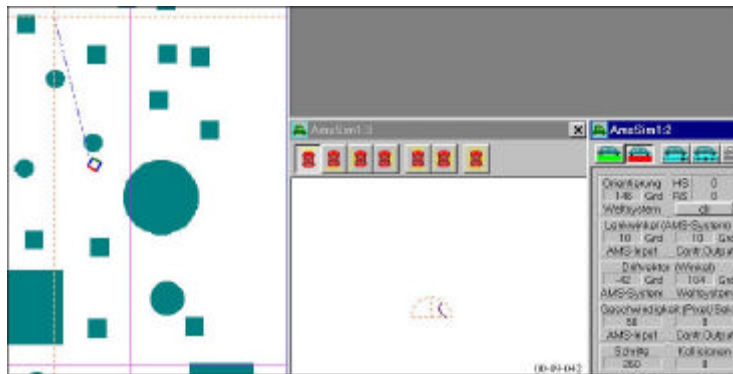


Abb. 4.11: Ausweichkurs des AMS

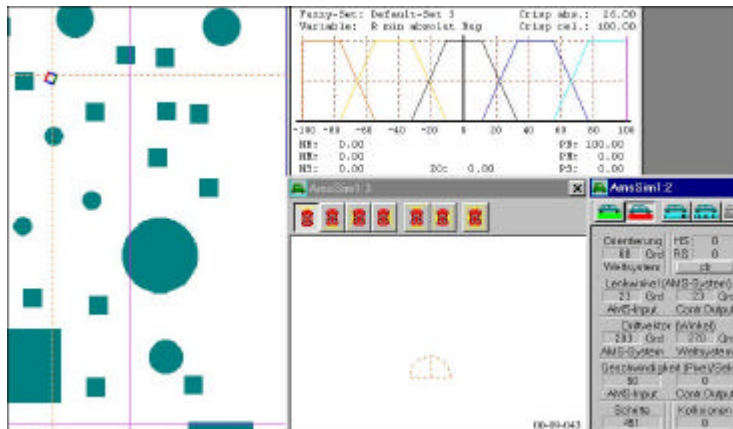


Abb. 4.12: Der Zielpunkt ist erreicht

Im weiteren Verlauf wird der Zielpunkt direkt mit einem Lenkwinkel von $\Theta = 23^\circ$ umkreist; Kollisionen traten nicht auf.

In Geländekarte 2 wurde das AMS in Richtung des Korridors (Abb. 4.13) gezogen.

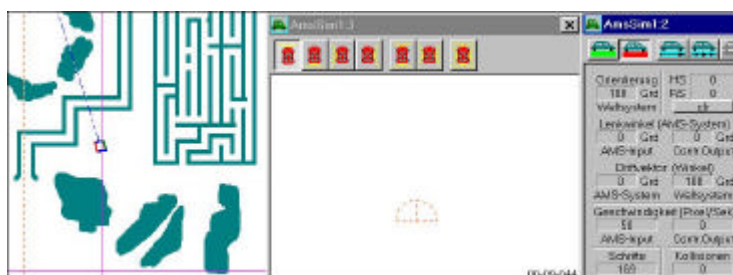


Abb. 4.13: Kurs auf den Korridor

Das mobile System drehte ab, detektierte die zwischen Korridor und Labyrinth angeordnete Passage und nutzte sie gemäß Abb. 4.14. Nach etwa 600 Schritten hatte es den Zielpunkt erreicht und umkreiste ihn auch in diesem Fall.

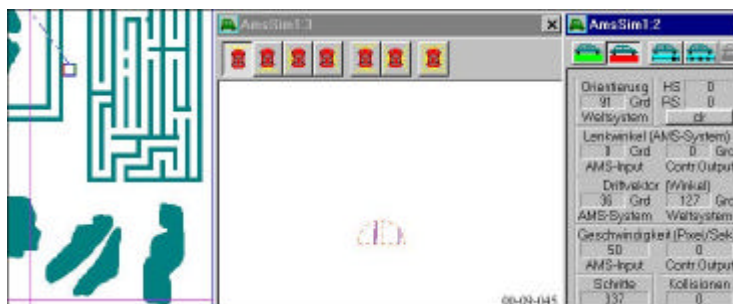


Abb. 4.14: Nutzung der Passage

Bei geringer Sensor-Reichweite erfüllte diese Strategie sämtliche Erwartungen. Das mobile System nahm Kurs auf das Ziel; es wich Hindernissen aus, ohne die Gesamttrichtung aufzugeben. Bei mittlerer und hoher Sensor-Reichweite traten in einigen Fällen Schwierigkeiten auf.

Die Strategie der minimalen Distanz (1 Punkt) mit dynamischer Regelbasis und dynamischer Überlagerung des Driftwinkels ist in der Lage, ein mobiles System kollisionsfrei zu einem Ziel zu führen.

Die noch auftretenden Unzulänglichkeiten dürften sich als Optimierungsproblem erweisen, das sich durch Modifizierung des Verlaufs der Wichtung K_D (siehe Abb. 3.7) unter Umständen lösen läßt. Ein weiterer Ansatz besteht darin, nicht nur den Hindernisabstand r_{\min} , sondern auch dessen Winkel in die Wichtungsrechnung einfließen zu lassen.

5 Zusammenfassung der Ergebnisse

Unser Ansatz begann mit dem Studium einfacher Strategien, deren einzige Aufgabe darin bestand, das mobile System kollisionsfrei durch eine Hindernisumgebung zu führen. Dabei stellte sich heraus, daß es hinreichend war, den Punkt der minimalen Hindernisdistanz als Kriterium zu wählen. Die Optimierung erfolgte durch Auswahl entsprechender Fuzzy-Sets sowie Dynamisierung des Regelwerks.

Die darauf aufbauende Hybridstrategie erfüllte in der vorliegenden Form die Erwartungen nicht. Sie kann durch Optimierungen der Zugehörigkeitsfunktionen und eventuell des Regelwerks mit großer Wahrscheinlichkeit verbessert werden.

Die auf der minimalen Distanz basierende zielpunktorientierte Strategie mit dynamischer Regelbasis und dynamischer Überlagerung des Driftwinkels erzielte sehr gute Ergebnisse. Auch ihre aufgezeigten Schwächen dürften sich als Optimierungsproblem erweisen.

Unsere Untersuchung hat gezeigt, daß sich ein mobiles System mit der Strategie der "minimalen Distanz (1 Punkt)" mit dynamischer Regelbasis und dynamischer Überlagerung des Driftwinkels kollisionsfrei bewegt und bis auf einige Ausnahmen den Zielpunkt erreicht.

Diese Strategie ist zu optimieren und anschließend weiterzuentwickeln.

6 Literatur

- [1] Warias, R.: *Entwicklung von Fuzzy-Kollisionsvermeidungs-Strategien einschließlich zugehöriger Simulations-Software für mobile Roboter*. Diplomarbeit, FernUniversität in Hagen, 2004

Formulierung problemspezifischer Einstellregeln für das Fuzzy-ROSA-Verfahren

Martin Sternke, Peter Krause¹

Lehrstuhl für Regelungssystemtechnik
Universität Dortmund, 44221 Dortmund
Tel.: (0231) 755-2761 Fax: (0231) 755-2752
E-Mail: martin.sternke@uni-dortmund.de

¹NuTech Solutions GmbH
Martin-Schmeisser-Weg 15, 44227 Dortmund
Tel.: (0231) 725463-10 Fax: (0231) 725463-29
E-Mail: krause@nutechsolutions.de

Zusammenfassung

Eine Methode für die datenbasierte Generierung von Mamdani-Fuzzy-Systemen ist das Fuzzy-ROSA-Verfahren¹. Das Verfahren basiert auf dem Michigan-Ansatz und erzeugt daher nur solche Regeln, die einen entsprechenden statistischen Test bestehen, dessen Schwere über Parameter gesteuert wird. Je nach gewähltem Regeltest und Parametereinstellung ändert sich die Zusammensetzung der erzeugten Regelbasis und damit die Qualität der Modellierung. Die grundsätzlichen Auswirkungen der Parametereinstellungen auf die Erzeugung der Regelbasis sind bekannt und es existieren Erfahrungswerte, wie sich Veränderungen der Regelbasis auf die Qualität der Modellierung auswirken. Es fehlen aber Erkenntnisse darüber, wie die Parameter für ein konkretes Problem möglichst günstig einzustellen sind. Dieser Aspekt ist insbesondere für industrielle Anwender von großem Interesse, da diese aus Zeit- und Kostengründen möglichst rasch gute Modelle erstellen wollen, ohne viele Einstellvarianten untersuchen zu müssen. In diesem Beitrag wird ein Ansatz vorgestellt, mit dem problemspezifische Einstellregeln sowohl wissensbasiert als auch datenbasiert hergeleitet werden. Die datenbasierte Suche nach Einstellregeln wird dabei mit dem Fuzzy-ROSA-Verfahren selbst durchgeführt.

1 Einleitung

Mit wachsender Komplexität moderner Prozesse besteht zunehmend ein Bedarf an Verfahren zur Modellierung solcher Prozesse. Dabei werden in den Bereichen des Data-Mining und der datenbasierten Modellierung verstärkt Verfahren aus der Computational Intelligence eingesetzt. Fuzzy-Systeme haben sich dabei für viele Aufgabenstellungen als geeignet im Hinblick auf die erzielte Modellgenauigkeit erwiesen und sie werden von Anwendern wegen ihrer Transparenz häufig bevorzugt akzeptiert. Es gibt unterschiedliche Typen von Fuzzy-Systemen, die sich im Hinblick auf Interpretierbarkeit und Genauigkeit unterscheiden. Grundsätzlich sind beide Eigenschaften wünschenswert, bei vielen Verfahren muss jedoch einer der Vorteile zugunsten des anderen aufgegeben werden.

Die zwei gängigen Typen von Fuzzy-Systemen mit den Eingangsgrößen X_1, \dots, X_n und der Ausgangsgröße U sind das Mamdani-System [1] und das TSK-System [2]. Ein Mamdani-System verwendet Regeln folgender Art:

$$\text{WENN } \langle P \rangle \text{ DANN } \langle C \rangle \quad . \quad (1)$$

¹Regelorientierte statistische Analyse

Dabei ist P eine vom Eingangsgrößenvektor $\mathbf{x} = (x_1, \dots, x_n)$ abhängige Prämisse und C eine von der Ausgangsgröße abhängige Konklusion der Form $u = \mathcal{S}_{U,i}$, wobei $\mathcal{S}_{U,i}$ ein linguistischer Ausgangsgrößenwert ist. Im Gegensatz dazu haben die Regeln in einem TSK-System die Form:

$$\text{WENN } \langle P \rangle \text{ DANN } \langle f(\mathbf{x}) \rangle \quad , \quad (2)$$

das heißt, sie verwenden Prämissen derselben Form, aber Konklusionen in Form einer wählbaren Funktion $u = f(\mathbf{x})$. Der Aufbau bzw. die Auswertung der Prämisse P einer Regel geschieht bei beiden Systemtypen in gleicher Weise. Da sie sich jedoch im Aufbau der Konklusion einer Regel unterscheiden, gilt dies folglich auch für die Berechnung des scharfen Ausgangsgrößenwertes u_D des jeweiligen Fuzzy-Systems. Zu einem Eingangsgrößenvektor \mathbf{x} wird bei einem Mamdani-System zunächst die ausgangsseitige Zugehörigkeitsfunktion

$$\mu(\mathbf{x}, u) = \bigvee_{i=1}^R [\mu_{P_i}(\mathbf{x}) \wedge \mu_{U_{C_i}}(u)] \quad (3)$$

berechnet. Dabei ist $\mu_{P_i}(\mathbf{x})$ der Erfülltheitsgrad der Prämisse P_i einer Regel. Bei einem Mamdani-System ist $\mu_{U_{C_i}}(u)$ die Zugehörigkeit des Ausgangsgrößenwertes u zu dem in der Konklusion der betrachteten Regel spezifizierten linguistischen Ausgangsgrößenwert. Die ausgangsseitige Zugehörigkeitsfunktion wird dann mit einem wählbaren Verfahren defuzzifiziert, beispielsweise mit der Schwerpunkt-Methode. Bei einem TSK-System erfolgt die Berechnung des scharfen Ausgangsgrößenwertes u_D meist durch eine gewichtete Mittelwertbildung über die Empfehlungen aller durch den Eingangsgrößenvektor \mathbf{x} aktivierten Regeln:

$$u_D(\mathbf{x}) = \frac{\sum_{i=1}^R \mu_{P_i}(\mathbf{x}) \cdot f_i(\mathbf{x})}{\sum_{i=1}^R \mu_{P_i}(\mathbf{x})} \quad . \quad (4)$$

2 Das Fuzzy-ROSA-Verfahren

Zur automatischen datenbasierten Generierung von Mamdani-Systemen kann beispielsweise das Fuzzy-ROSA-Verfahren [3, 4, 5] eingesetzt werden. Dieses Verfahren basiert auf der statistischen Relevanzanalyse von Hypothesen zur Generierung von Regeln. Jede so erzeugte Regel beschreibt jeweils einen relevanten Teilaspekt des zu modellierenden Prozesses. Auf diese Weise wird eine kompakte Regelbasis erzeugt. Durch eine nachgeschaltete Transformation des so erzeugten Mamdani-Modells in ein TSK-Modell kann die Modellgenauigkeit oftmals noch deutlich verbessert werden, wobei die Interpretierbarkeit der Regeln jedoch erhalten bleibt [6].

Die datenbasierte Modellierung von MISO-Systemen² mithilfe des Fuzzy-ROSA-Verfahrens gliedert sich in vier Teilaufgaben: Auswahl der Eingangsgrößen, Strukturierung des Suchraumes, Regelgenerierung und Optimierung der Regelbasis. Im Rahmen der systematischen Exploration der Einstellmöglichkeiten und der Formulierung problemspezifischer Einstellregeln wird auf eine vorgeschaltete Selektion von Eingangsgrößen [7] verzichtet, weil industrielle Anwender meist zunächst alle verfügbaren Eingangsgrößen berücksichtigen möchten. Auf eine nachgeschaltete Optimierung der Regelbasis [8] wird

²Multiple Inputs Single Output

ebenfalls verzichtet, weil diese so zeitintensiv ist, dass sie bei der großen Anzahl der hier erzeugten Modelle nicht in angemessener Zeit durchführbar ist. Die systematische Untersuchung erstreckt sich somit auf die Suchraumstrukturierung und die Regelgenerierung.

2.1 Suchraumstrukturierung für die Regelgenerierung

Die geeignete Strukturierung des Suchraumes hat entscheidenden Einfluss auf den Erfolg der nachfolgenden Regelsuche. Bei einer schlecht gewählten Strukturierung schlägt die Regelgenerierung fehl, auch wenn sich entsprechende Zusammenhänge in den Daten prinzipiell abzeichnen.

Die Strukturierung des Suchraumes besteht in der Festlegung der Anzahl der linguistischen Werte je Eingangs- bzw. Ausgangsgröße sowie der Festlegung der Zugehörigkeitsfunktionen für die Modellierung der linguistischen Werte. Dies kann wissensbasiert, datenbasiert oder heuristisch geschehen. Im Prinzip kann die Festlegung von linguistischen Werten und Zugehörigkeitsfunktionen für jede Eingangs- und die Ausgangsgröße individuell erfolgen. Die daraus resultierende kombinatorische Vielfalt impliziert jedoch sehr viele Freiheitsgrade. In praktischen Anwendungen, in denen kein spezielles Problemwissen vorliegt, hat sich die Beschränkung auf gewisse Standard-Suchraumstrukturierungen als ausreichend erwiesen. Die Anzahl linguistischer Werte wird dabei für alle Größen gleich – typischerweise zu drei, fünf oder sieben – gewählt. Ebenso werden die Zugehörigkeitsfunktionen für alle Größen in gleicher Weise – äquidistant oder äquifrequent – über den Wertebereich einer Größe platziert.

Für eine automatische Strukturierung des Suchraumes wird im Fuzzy-ROSA-Verfahren ein unidimensionales Clusterverfahren eingesetzt, das auf einem bekannten K -Means Algorithmus [9, 10] basiert. Für jede Größe werden dabei eine günstige Anzahl linguistischer Werte und entsprechende Zugehörigkeitsfunktionen hierzu erzeugt. Im Rahmen der Untersuchungen [11] wurde zunächst gezeigt, dass eine so erzeugte Suchraumstrukturierung für viele Modellierungsprobleme zu vergleichbar guten, teilweise auch zu besseren Ergebnissen führt wie die jeweils am besten geeigneten Standard-Suchraumstrukturierungen. Daher wird im Folgenden die automatische Suchraumstrukturierung verwendet. Alle nachfolgenden Ausführungen beziehen sich auf die Ergebnisse, die bei dieser Form der Suchraumstrukturierung erzielt werden.

2.2 Datenbasierte Generierung relevanter Mamdani-Regeln

Eine potenziell relevante Mamdani-Regel wird im Fuzzy-ROSA-Verfahren als Hypothese formuliert und einem statistischen Test unterzogen, um festzustellen, ob sie einen relevanten Teilaspekt des zu modellierenden Prozesses beschreibt. Ist dies der Fall, wird die Hypothese zur Regel erklärt, geeignet bewertet und der Regelbasis hinzugefügt. Andernfalls wird die Hypothese verworfen. Auf diese Weise wird die Regelbasis des Mamdani-Systems sukzessive aus einzelnen relevanten Regeln aufgebaut.

Das Ergebnis des Regelgenerierungsprozesses hängt – bei gegebener Suchraumstrukturierung – von dem gewählten Test- und Bewertungsverfahren und dessen Parametrierung ab. Für das Testen von Regelhypothesen und die Bewertung von Regeln wurden unterschiedliche Test- und Bewertungsverfahren entwickelt. Detaillierte Ausführungen hierzu

finden sich in [12, 13, 14, 15, 16]. Der Entwicklung der verschiedenen Test- und Bewertungsverfahren liegen unterschiedliche Anforderungen und Zielsetzungen zu Grunde, die nachfolgend kurz beschrieben werden.

Relevanz:

Ziel dieser Strategie ist es, Regeln zu finden, die einen kausalen Zusammenhang zwischen der in der Regelprämisse spezifizierten linguistischen Eingangssituation und dem in der Regelkonklusion spezifizierten linguistischen Ausgangsgrößenwert beschreiben.

Treffsicherheit:

Ziel dieser Strategie ist es, Regeln zu finden, die zu der in der Regelprämisse spezifizierten linguistischen Eingangssituation den korrekten linguistischen Ausgangsgrößenwert empfehlen.

Approximationsgüte:

Ziel dieser Strategie ist es, Regeln zu finden, die zu der in der Regelprämisse spezifizierten linguistischen Eingangssituation denjenigen linguistischen Ausgangsgrößenwert empfehlen, der im Sinne einer Minimierung des Approximationsfehlers am besten geeignet ist.

Die oben genannten Anforderungen und Zielsetzungen sind teilweise widersprüchlich oder gegenläufig. In der Praxis können sie daher im Allgemeinen nicht alle gleichzeitig voll erfüllt werden.

Im Fuzzy-ROSA-Verfahren stehen insgesamt sieben Test- und Bewertungsverfahren zur Verfügung, die sich in zwei Klassen zusammenfassen lassen. Eine Klasse bilden diejenigen Verfahren, die auf Grundlage der relativen Datenhäufigkeiten arbeiten. Eine weitere Klasse bilden diejenigen Verfahren, die sich auf Datenverteilungen stützen. Die Strategien der erstgenannten Klasse verfolgen eher die oben genannten Zielsetzungen Relevanz und Treffsicherheit, die der anderen Klasse primär das Ziel einer hohen Approximationsgüte.

2.3 Einstelloptionen für die Regelgenerierung

Der Anwender kann das Ergebnis der Regelgenerierung über die folgenden drei Parameter beeinflussen: Die Wahl eines geeigneten Test- und Bewertungsverfahrens J , die Schwere κ des eingesetzten statistischen Tests und den Schwellwert τ für die relevanzabhängige Übernahme von Regeln in die Regelbasis.

In Abschnitt 2.2 wird auf die sieben Test- und Bewertungsverfahren (J_{RI} , J_{RHR} , J_{NHR} , J_{CHR} , J_{MVB} , J_{VVB} und J_{CVB}) hingewiesen, die im Fuzzy-ROSA-Verfahren zur Verfügung stehen. Der Anwender muss zunächst die diskrete Entscheidung treffen, welches der Verfahren eingesetzt werden soll.

Jedes Test- und Bewertungsverfahren verfügt über einen Parameter κ , über den die Schwere des statistischen Tests gesteuert wird. Der Wert für κ kann innerhalb eines vorgegebenen zulässigen Bereichs frei gewählt werden.

Jeder generierten Mamdani-Regel wird im Fuzzy-ROSA-Verfahren durch die Bewertung ein Relevanzgrad $\rho \in]0, 1]$ zugewiesen. Dieser gibt an, wie gut eine Regel durch die zu

Grunde liegenden Lerndaten statistisch abgesichert ist. Der Relevanzgrad kann in einem Mamdani-System bei der Inferenz entsprechend berücksichtigt werden:

$$\mu(\mathbf{x}, u) = \bigvee_{i=1}^R [\rho_i \wedge \mu_{P_i}(\mathbf{x}) \wedge \mu_{U_{C_i}}(u)] \quad . \quad (5)$$

Über den Parameter $\tau \in [0, 1]$ wird die Relevanzschwelle festgelegt, ab der eine Regel tatsächlich der Regelbasis hinzugefügt wird. Falls $\rho_i < \tau$ gilt, wird die Regel verworfen.

2.4 Erfahrungswerte über die Auswirkungen der Einstelloptionen auf die Regelbasis und die resultierende Modellierungsqualität

Die Auswirkungen der gewählten Einstelloptionen auf die Modellierungsqualität lassen sich nicht allgemein und auch nicht exakt vorhersagen. Es existieren aber umfangreiche Erfahrungswerte, die durch den Einsatz des Fuzzy-ROSA-Verfahrens für Benchmarkprobleme und industrielle Problemstellungen gewonnen wurden.

Im Hinblick auf die geeignete Wahl des Test- und Bewertungsverfahrens lassen sich bisher nur sehr eingeschränkt Aussagen machen. Die besten Modellierungsergebnisse werden häufig mit dem Relevanzindex J_{RI} [12] oder dem Mittelwertbasierten Test J_{MVB} [13] erzielt. Der Relevanzindex ist für die Bearbeitung von Klassifikationsaufgaben meist besser geeignet, wohingegen der Mittelwertbasierte Test insbesondere bei Approximationsaufgaben vorteilhaft ist.

Die Auswirkungen des Parameters κ für die Schwere des statistischen Tests auf die Anzahl der generierten Regeln sind bekannt. Je leichter der statistische Test gewählt wird, desto mehr Regeln werden tendenziell generiert. Wird der statistische Test hingegen sehr schwer gewählt, so ist die Anzahl der generierten Regeln tendenziell eher gering. In dem Fall, dass der Test sehr leicht gewählt und daher sehr viele Regeln generiert werden, lässt sich oftmals beobachten, dass dies für die Modellierungsqualität eher nachteilig ist. Der Grund hierfür liegt darin, dass so auch viele nur dürftig abgesicherte Regeln erzeugt werden. Dadurch treten in der Regelbasis verstärkt Widersprüche auf, die die Modellierungsqualität beeinträchtigen. In dem Fall, dass der Test sehr schwer gewählt und daher sehr wenige Regeln generiert werden, lässt sich ebenfalls oftmals eine eher mäßige Modellierungsqualität beobachten. Der Grund liegt in diesem Fall meist darin, dass für viele linguistische Eingangssituationen keine Regeln erzeugt werden und bei der Modellierung bzw. Prognose auf eine Default-Regel zurückgegriffen werden muss. Da diese in der Regel nur eine Behelfslösung ist, gelingt die Modellierung entsprechender Daten meist nur sehr ungenau.

Die Auswirkungen des Parameters τ für die relevanzabhängige Übernahme generierter Regeln auf die Größe der Regelbasis sind ebenfalls bekannt. Je höher der Wert für τ gewählt wird, desto weniger Regeln werden in die Regelbasis übernommen. In dem Fall, dass die Schwelle sehr niedrig gewählt wird, lässt sich oftmals beobachten, dass dies für die Modellierungsqualität eher nachteilig ist, weil so auch nur sehr schwach gestützte Regeln in die Regelbasis übernommen werden und so verstärkt Widersprüche auftreten. In dem Fall, dass die Schwelle sehr hoch gewählt wird, lässt sich ebenfalls oftmals eine eher mäßige Modellierungsqualität beobachten. Der Grund hierfür liegt in diesem Fall typischerweise darin, dass für viele linguistische Eingangssituationen auf eine Default-Regel zurückgegriffen werden muss.

3 Formulierung problemspezifischer Einstellregeln

3.1 Motivation

Für die Akzeptanz des Fuzzy-ROSA-Verfahrens in der Industrie ist entscheidend, dass es auch von weniger versierten Anwendern ohne großen Aufwand zur Lösung komplexer Probleme eingesetzt werden kann. Häufig fordern Anwender, dass die Anwendung des Verfahrens ohne besondere Kenntnisse im Bereich der datenbasierten Fuzzy-Modellierung möglich sein muss. Oft wird eine *Ein-Knopf-Lösung* für den Einsatz eines Verfahrens gefordert. Um dennoch eine hohe Flexibilität zu bieten, wird meist gefordert, dass die Wahl der Einstellwerte problemabhängig erfolgt. Deshalb kommt der Entwicklung von Einstellregeln eine große Bedeutung zu.

Ein Ziel besteht darin, anhand von Daten, die durch systematische Untersuchungen an unterschiedlichen Modellierungsproblemen gewonnenen werden, qualitative Zusammenhänge zwischen Kenngrößen der Modellierungsprobleme, den vorgenommenen Einstellungen und den damit erzielten Resultaten aufzudecken. Daraus werden problemspezifische Empfehlungen für die Einstellwerte freier Parameter abgeleitet, die industriellen Anwendern helfen, das Verfahren zur Lösung komplexer praktischer Probleme erfolgreich zu nutzen. Ein anspruchsvolleres Ziel ist es, die vermuteten Zusammenhänge in den gewonnenen Daten durch das Verfahren selbst in Form von Regeln automatisch aufzufinden und daraus ein Modul zu generieren, das anhand von Kenngrößen eines gegebenen neuen Modellierungsproblems für jede Wahl der Einstellparameter den zu erwartenden Modellierungsfehler prognostiziert. Damit können vor der eigentlichen Modellgenerierung unterschiedliche Parametereinstellungen simulativ untersucht und eine möglichst gute Parametereinstellung ausgewählt werden, um so den Prozess der Modellbildung insgesamt zu beschleunigen oder vollständig zu automatisieren. Mit einem entsprechenden Modul können auch solche Parametereinstellungen simulativ bewertet werden, die im Rahmen der vorherigen Exploration noch gar nicht untersucht worden sind. Bei entsprechender Prognosegenauigkeit kann das Modul somit auch für eine Optimierung der Parametereinstellungen genutzt werden. Für eine solche Optimierung können Verfahren aus dem Bereich der Computational Intelligence eingesetzt werden, beispielsweise Evolutionsstrategien [17, 18] oder Partikel-Schwarm-Optimierungsverfahren [19].

3.2 Beschreibung der durchgeführten Experimente

Als Orientierung für die durchgeführten Untersuchungen dienen Vorarbeiten [15], die im Rahmen des Sonderforschungsbereichs 531 der Universität Dortmund durchgeführt wurden. Diese Untersuchungen haben wichtige grundlegende Zusammenhänge zwischen den angestrebten Zielen Relevanz und Treffsicherheit von Regeln sowie der Approximationsgüte des Modells einerseits und der Wahl und Parametrierung des Test- und Bewertungsverfahrens andererseits aufgedeckt. In [15] wird für ein gegebenes Modellierungsproblem zunächst eine als geeignet angesehene Suchraumstrukturierung vorgenommen. Die Anzahl der linguistischen Werte je Eingangs- bzw. Ausgangsgröße wird wissensbasiert festgelegt und die Erzeugung der Zugehörigkeitsfunktionen hierzu erfolgt datenbasiert. Es werden vier unterschiedlichen Test- und Bewertungsverfahren für die Modellgenerierung eingesetzt. Der Parameter κ wird für jedes der Test- und Bewertungsverfahren auf dem

vorgesehenen Standardwert belassen oder aber aufgrund von Zusatzwissen problemspezifisch angepasst. Für den Parameter τ werden für jedes Test- und Bewertungsverfahren die Werte $\tau_1 = 0.0, \tau_2 = 0.1, \dots, \tau_{10} = 0.9, \tau_{11} = 1.0$ vorgesehen. Es ergeben sich so 11 Parametrierungen für ein gewähltes Test- und Bewertungsverfahren und damit für die vier eingesetzten Test- und Bewertungsverfahren insgesamt 44 Parametrierungen.

In diesen Untersuchungen werden zu jedem eingesetzten Test- und Bewertungsverfahren sieben verschiedene Werte für κ angesetzt. Diese decken die Schwierigkeitsgrade *einfach*, *normal* und *anspruchsvoll* ab. Für den Parameter τ werden hier die zehn Werte $\tau_1 = 0.0, \tau_2 = 0.1, \dots, \tau_9 = 0.9, \tau_{10} = 0.9$ verwendet. Es ergeben sich so 70 Parametrierungen für ein gewähltes Test- und Bewertungsverfahren und damit für die sieben Test- und Bewertungsverfahren insgesamt 490 Parametrierungen. Diese werden im Folgenden als Standard-Parametrierungen bezeichnet.

Aufgrund der relativ großen Anzahl an Standard-Parametrierungen wird die Suchraumstrukturierung vorab einmalig festgelegt (siehe Abschnitt 2.1), da sich durch die kombinatorische Vielfalt aus Suchraumstrukturierungen und Parametrierungen eine zu große Anzahl an Kombinationsmöglichkeiten ergibt. Auch aus einem anderen Grund erscheint es nicht sinnvoll, alle möglichen Kombinationen aus Suchraumstrukturierungen und Parametrierungen zu untersuchen. Hinsichtlich der Parametrierungen kann nämlich vorab gesagt werden, welchen qualitativen Einfluss Veränderungen der Parametrierung auf die Regelbasis und die Modellierungsqualität haben (siehe Abschnitt 2.4). Die Auswirkungen einer veränderten Suchraumstrukturierung erweisen sich dagegen als wesentlich problemspezifischer und lassen sich weitaus weniger gut abschätzen. Eine direkte Nutzung der zu allen Suchraumstrukturierungen erhobenen Daten für die Aufdeckung der gesuchten Zusammenhänge ist problematisch, weil eine Parametrierung in Verbindung mit einer Suchraumstrukturierung gut, in Verbindung mit einer anderen Suchraumstrukturierung aber schlecht geeignet sein kann. Die Gesamtheit all dieser Daten kann daher verstärkt Mehrdeutigkeiten und Widersprüche beinhalten, die für die angestrebte datenbasierte Generierung eines Moduls zur Prognose des Modellierungsfehlers nachteilig sind.

3.3 Charakterisierung von Modellierungsproblemen und Bewertung der erzielten Modellierungsergebnisse

Es werden insgesamt zwölf bekannte Modellierungsprobleme (z. B. Mackey-Glass Zeitreihe, Boston Housing Problem, etc.) untersucht. Zehn davon sind Approximationsprobleme, zwei sind Klassifikationsprobleme. Sie überdecken ein breites Spektrum, beispielsweise im Hinblick auf die Anzahl der zu beachtenden Eingangsgrößen. Allen erzielten Modellierungsergebnissen liegt eine Kreuzvalidierung zu Grunde, die auf einer Aufteilung der verfügbaren Daten in zwei gleich große Datensätze basiert.

Die einer datenbasierten Modellierung zu Grunde liegenden Daten unterscheiden sich typischerweise stark von Problem zu Problem, beispielsweise in der Anzahl der Eingangsgrößen. Außerdem sind die Wertebereiche der einzelnen Größen einerseits von Problem zu Problem verschieden, meist aber auch innerhalb des Datensatzes eines Problems selbst. Um eine überschaubare Anzahl aussagekräftiger Kenngrößen zu erhalten, die zudem eine Vergleichbarkeit verschiedener Probleme ermöglichen, wird bei der Berechnung der Kenngrößen wie folgt vorgegangen: Die Eingangsgrößen und die Ausgangsgröße eines

Datensatzes werden zunächst jeweils individuell auf den Wertebereich $[0, 1]$ normiert³. Nach der Normierung werden für jede Eingangsgröße verschiedene Kenngrößen, wie beispielsweise die Standardabweichung oder die Korrelation mit der Ausgangsgröße, berechnet⁴. Da die Anzahl der Eingangsgrößen von Problem zu Problem variiert, gilt dies folglich auch für die Anzahl der hierzu insgesamt gebildeten Kenngrößen. Die Abhängigkeit der Anzahl der Kenngrößen von der Anzahl der Eingangsgrößen wird hier für den Entwurf des angestrebten Prognosemoduls als ungünstig angesehen. Deshalb werden die Kenngrößen geeignet aggregiert, beispielsweise werden die Standardabweichungen der normierten Eingangsgrößen zu einer *mittleren* Standardabweichung zusammengefasst. Damit einzelne Eingangsgrößen das Gesamtbild eines Datensatzes nicht allzu stark verzerren können, wird für die Aggregation nicht der Mittelwert, sondern der Median verwendet. In dieser Weise werden auch für die Ausgangsgröße entsprechende Kenngrößen auf Grundlage der normierten Daten berechnet. Diese Vorgehensweise liefert für jedes Modellierungsproblem stets die gleiche Anzahl von Kenngrößen, unabhängig von der Anzahl der Eingangsgrößen. Zudem bewegen sich die so berechneten Kenngrößen stets innerhalb bekannter Wertebereiche, da sie anhand normierter Daten berechnet werden.

Zur Bewertung der Modellierungsergebnisse werden normierte Fehlerwerte ϵ_{norm} herangezogen. Dabei wird beispielsweise der mittlere absolute Fehler eines Fuzzy-Modells in das Verhältnis zu dem Fehler eines sogenannten Trivialmodells gesetzt, das unabhängig von dem Eingangsgrößenvektor x stets den gleichen Ausgabewert liefert. Gilt $\epsilon_{norm} = 1$, so arbeitet das Fuzzy-Modell nicht besser als dieses Trivialmodell und bringt daher offensichtlich keinen Zugewinn an Modellgenauigkeit. Der betriebene Mehraufwand im Vergleich zu dem Trivialmodell erscheint daher nicht gerechtfertigt. Erst bei $\epsilon_{norm} < 1$ lohnt der mit einem Fuzzy-Modell verbundene Mehraufwand.

Die berechneten Kenngrößen und erzielten Modellierungsergebnisse werden für jedes untersuchte Problem in eine Matrix eingetragen. Die einzelnen Matrizen zu den untersuchten Problemen werden in einer Übermatrix zusammengefasst. Die Analyse dieser Übermatrix zielt nicht auf die Untersuchung eines speziellen Problems ab, sondern vielmehr darauf, allgemeine Zusammenhänge aufzudecken, die sich für unterschiedliche Probleme in ähnlicher Form abzeichnen. Eine solche Übermatrix ist daher für die Suche nach den vermuteten Zusammenhängen geeignet.

3.4 Wissensbasierte Formulierung von Einstellregeln

Die Suche nach Zusammenhängen zwischen Problemkenngrößen, vorgenommenen Einstelloptionen und damit erzielten Resultaten kann wissensbasiert erfolgen, beispielsweise durch eine geeignete Visualisierung der gewonnenen Daten und darauf aufbauenden Schlussfolgerungen, und führt zu qualitativen Regeln, die die Zusammenhänge beschreiben.

Die wissensbasierte Formulierung von Einstellempfehlungen gestaltet sich aufgrund der großen Anzahl potenziell relevanter Kenngrößen schwierig. Um überhaupt eine Chance hierfür zu haben, erscheint es angebracht, diese auf ihre Relevanz hin zu untersuchen und die Anzahl der zu beachtenden Kenngrößen ggf. zu reduzieren. Hierfür wird das in [7]

³Ein normierter Wert von 0 entspricht dem kleinsten unnormierten Wert und ein normierter Wert von 1 entsprechend dem größten unnormierten Wert einer Größe.

⁴Auf die Auflistung aller berechneten Kenngrößen wird an dieser Stelle verzichtet.

beschriebene evolutionäre Verfahren für die Merkmalsselektion eingesetzt. Das Verfahren stuft dabei zwei Kenngrößen eines Problems als besonders relevant ein: zum einen die aggregierte Korrelation zwischen einer Eingangsgröße und der Ausgangsgröße, zum anderen die aggregierte Standardabweichung der Eingangsgrößen. Die erste Kenngröße kann als ein Indikator dafür aufgefasst werden, wie eindeutig sich Zusammenhänge zwischen Eingangsgrößen und Ausgangsgröße in dem Problem abzeichnen und wie schwierig sich eine Modellierung wohl grundsätzlich gestalten wird. Die zweite Kenngröße ist ein Maß für die Streuung. Bei einer starken Streuung sind die der Modellierung zu Grunde liegenden Daten ungleichmäßig verteilt und es ist zu vermuten, dass es sich in solchen Fällen schwieriger gestaltet, datenbasiert Regeln zu generieren. Die von dem Verfahren auf Grundlage von Daten als relevant eingestuften Kenngrößen eines Problems erscheinen somit auch bei einer wissensbasierten Betrachtung durchaus plausibel.

Die untersuchten Modellierungsprobleme wurden unter anderem anhand der beiden genannten Kenngrößen in einer Art Landkarte visualisiert. In weitergehenden Analysen zeigt sich, dass für unterschiedliche Modellierungsprobleme, die sich im Hinblick auf diese Kenngrößen auf dieser Landkarte zusammen gruppieren, meist ein und dasselbe Test- und Bewertungsverfahren besonders gut geeignet ist. Offenbar sind diese Kenngrößen für die problemabhängige Wahl eines Test- und Bewertungsverfahrens relevant.

Bezüglich der Parameter κ und τ zeigt sich bei der Auswertung der gewonnenen Daten folgende Tendenz: Es ist oftmals günstig, die Schwere κ des statistischen Tests etwas leichter zu wählen, als dies bisher standardmäßig vorgeschlagen wird. Gleichzeitig sollte die Relevanzschwelle aber etwas höher gewählt werden, als dies bisher standardmäßig vorgeschlagen wird. Ein Wert von $\tau = 0.4$ erweist sich in vielen Fällen als gut geeignet. Diese Beobachtungen decken sich mit den bereits vorhandenen Erfahrungswerten (siehe Abschnitt 2.4). Allerdings haben diese Untersuchungen konkretere Hinweise gegeben, wie die Schwere des Tests und die Relevanzschwelle zu wählen sind.

Anhand der in diesem Abschnitt beschriebenen Erkenntnisse werden so qualitative Regeln für die problemspezifische Wahl und Parametrierung eines günstigen Test- und Bewertungsverfahrens formuliert. Damit wird das einleitend beschriebene Zwischenziel eingelöst.

3.5 Datenbasierte Generierung eines Prognosemoduls

Die Suche nach Einstellregeln für freie Modellparameter kann auch datenbasiert erfolgen, beispielsweise mit dem Fuzzy-ROSA-Verfahren selbst. Das Erstellen eines entsprechenden Moduls, das die Zusammenhänge zwischen den Kenngrößen eines Modellierungsproblems, den Einstellungen freier Modellparameter und den erzielten Ergebnissen widerspiegelt, ist das hier verfolgte weiterführende Ziel.

Für ein gegebenes neues Modellierungsproblem soll eine möglichst günstige Einstellung freier Modellparameter – das angewandte Test- und Bewertungsverfahren sowie die Werte für dessen Parameter κ und τ – vorgenommen werden, um so ein möglichst genaues und korrektes Modell zu erstellen. Grundsätzlich kommen zwei unterschiedliche Ansätze für Typen von Modulen für diese Aufgabe in Frage. Der erste Ansatz besteht darin, charakteristische Kenngrößen des Modellierungsproblems zu bestimmen und einem Modul zuzuführen, das Empfehlungen für die Einstellung der freien Modellparameter als Ausgabe liefert. Ein solches Modul hat somit drei Ausgabewerte. Der andere Ansatz besteht

darin, ein Modul zu entwerfen, das die charakteristischen Kenngrößen des Modellierungsproblems *und* die Einstellung der freien Modellparameter als Eingangsgrößen verwendet und als Ausgabe eine Prognose des in diesem Fall zu erwartenden Modellierungsfehlers liefert. Ein solches Modul hat somit nur einen Ausgabewert. Der erste Ansatz ist aus unterschiedlichen Gründen eher ungeeignet. Ein Nachteil dieses Ansatzes liegt darin, dass eine modellgestützte Optimierung der Einstellparameter so nicht möglich ist. Für einen gegebenen Satz von Kenngrößen eines Modellierungsproblems liefert ein solches Modul genau einen bestimmten Satz von Empfehlungen für die Einstelloptionen. Dieser Ansatz wird daher nicht verfolgt. Als weiterführendes Ziel wird hier die datenbasierte Generierung eines Moduls zur Prognose des Modellierungsfehlers angestrebt. Im praktischen Einsatz werden die Kenngrößen des Modellierungsproblems vorab berechnet und nachfolgend als Eingangsgrößen des Moduls konstantgehalten. Die Einstelloptionen sind ebenfalls Eingangsgrößen des Moduls. Sie werden variiert und simulieren so die unterschiedlichen Parametrierungen und die zugehörigen Modelle. Jedem simulierten Modell wird entsprechend der jeweils prognostizierte Modellierungsfehler zugewiesen. Aus der Menge aller simulierten Modelle wird dann beispielsweise das Modell mit dem kleinsten zu erwartenden Modellierungsfehler ausgewählt und tatsächlich generiert. Anhand des prognostizierten und des tatsächlichen Modellierungsfehlers kann man beurteilen, ob das Prognosemodul hinreichend exakt arbeitet und ob diesem vertraut werden kann.

Es zeigt sich, dass ein akzeptabel genaues Prognosemodul mit dem Fuzzy-ROSA-Verfahren erzeugt werden kann. Zunächst wird versucht, *alle* in der Übermatrix gesammelten Daten für das Erlernen der Regeln zu verwenden. Dass dies nicht den gewünschten Erfolg bringt, lässt sich unter anderem dadurch erklären, dass innerhalb der Übermatrix Mehrdeutigkeiten und Widersprüche auftreten, die die Regelgenerierung erschweren. Zur Überwindung dieses Problems wird der für das Erlernen verwendete Datensatz ausgedünnt. Zu jedem im Rahmen der systematischen Exploration der Einstelloptionen untersuchten Modellierungsproblem werden 490 Modelle (siehe Abschnitt 3.2) generiert, davon werden aber jeweils nur die 25 besten betrachtet. Dies stellt sicher, dass alle untersuchten Modellierungsprobleme in den Lerndaten für die Generierung des Moduls berücksichtigt werden und damit der Raum der Problemkenngrößen gut abgedeckt wird. Ferner gewährleistet dies, dass ein breites Spektrum von Modellgüten abgedeckt wird, denn das beste für ein Modellierungsproblem erhaltene Modell weist nicht notwendigerweise auch eine hohe Modellierungsgüte auf. Auf Grundlage der so zusammengestellten Daten werden ein Mamdani-Modell und ein TSK-Modell generiert. Die Genauigkeit des Mamdani-Modells ist noch nicht ausreichend, die des TSK-Modells ist aber akzeptabel. Dieses TSK-Modell ist ein Prototyp des angestrebten Moduls zur Prognose des zu erwartenden Modellierungsfehlers. Seine Validierung an einem neuen Modellierungsproblem⁵, dessen Kenngrößen nicht Bestandteil des Lerndatensatzes sind, ergibt als Differenz zwischen dem tatsächlichen und dem prognostizierten Modellierungsfehler⁶ in einem Großteil der 490 Fälle (Standard-Parametrierungen) einen Wert kleiner 10%, in einigen Fällen zeigen sich jedoch auch größere Abweichungen. Dies sind überwiegend die Fälle, in denen ein realer normierter Modellierungsfehler $\epsilon_{norm} \geq 1$ erzielt wird. Die Prognosen sind

⁵Es handelt sich hierbei um das Benchmarkproblem *Cars* mit sieben Eingangsgrößen, bei dem der durchschnittliche Treibstoffverbrauch unterschiedlicher Autotypen modelliert werden soll (siehe auch <http://www.ics.uci.edu/~mlearn/MLSummary.html> und <http://lib.stat.cmu.edu/datasets/>). Auf diesen Seiten finden sich auch die meisten der hier untersuchten zwölf Modellierungsprobleme.

⁶Die Werte beziehen sich auf die gemittelten normierten Modellierungsfehler auf Validierungsdaten, die im Rahmen der Kreuzvalidierung genutzt werden.

für diese Fälle sehr ungenau, daher werden zunächst nur die Standard-Parametrierungen bestimmt, die für dieses Modellierungsproblem zu einem realen normierten Modellierungsfehler $\epsilon_{norm} < 1$ führen. Dies sind ca. 380 der 490 Standard-Parametrierungen. In Bild 1 sind hierfür die prozentualen betraglichen Abweichungen zwischen den realen und prognostizierten Modellierungsfehlern in Form eines Histogramms dargestellt. In Bild 2 sind die Abweichungen noch einmal in anderer Form dargestellt. Die realen normierten Modellierungsfehler werden in aufsteigender Reihenfolge sortiert und es werden hierzu der $\pm 10\%$ - und der $\pm 20\%$ -Toleranzbereich als gestrichelte Linien dargestellt. Die Darstellung der zugehörigen prognostizierten normierten Modellierungsfehler zeigt, dass sich der Großteil der prognostizierten Fehlerwerte innerhalb dieser Toleranzbereiche bewegt (siehe hierzu auch Bild 1). Auffällig ist, dass für kleinere reale Modellierungsfehler die Prognosen tendenziell eher zu hoch und für größere reale Modellierungsfehler eher zu niedrig ausfallen. Weiterhin ist zu erkennen, dass die Parametrierung, die von dem Prognosemodul in diesem Fall als am geeignetsten eingestuft wird, also den kleinsten prognostizierten Modellierungsfehler erzielt, in diesem Fall nicht den kleinsten realen Modellierungsfehler liefert.

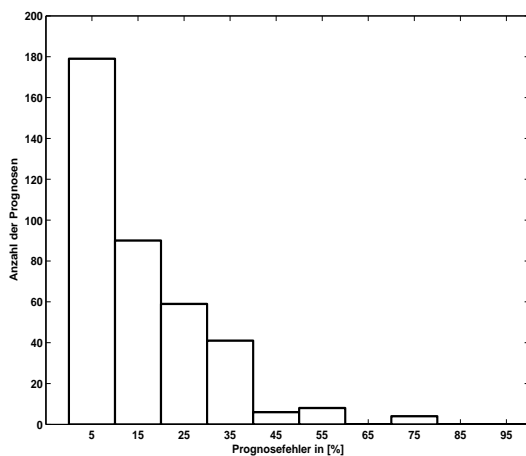


Bild 1: Histogramm der Prognosefehler für das Modellierungsproblem *Cars*.

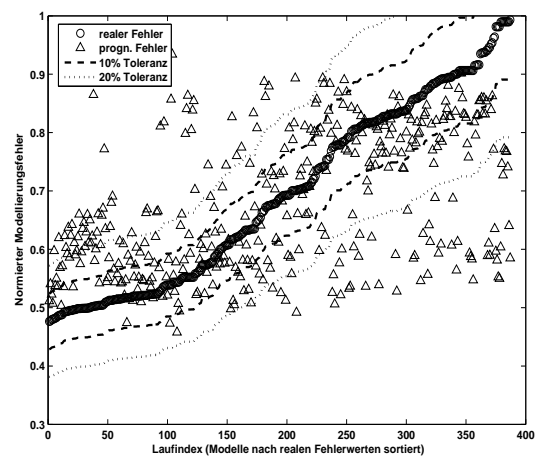


Bild 2: Reale und prognostizierte Fehlerwerte für das Modellierungsproblem *Cars*.

Insgesamt kann man feststellen, dass das Prognosemodul die realen Modellierungsfehler in der grundlegenden Tendenz nachbildet, insbesondere wenn man die Auswertung auf die Werte innerhalb der Toleranzbereiche fokussiert. Es stellt sich die Frage, welche realistischen Erwartungen an ein solches Prognosemodul grundsätzlich gestellt werden können. Es erscheint eher unrealistisch, exakte Prognosen erzielen zu können, da jedes neue Modellierungsproblem sich von den bisher untersuchten Modellierungsproblemen in irgendeiner Art und Weise soweit unterscheidet, dass das Zusammenspiel aus automatischer Suchraumstrukturierung und Regelgenerierung zu einem Ergebnis führt, dass sich nicht genau vorhersagen lässt. Für das hier betrachtete Beispiel wäre aber eine Verbesserung des Prognosemoduls dahingehend erstrebenswert, dass die Bandbreite der Abweichungen zwischen realen und prognostizierten Modellierungsfehlern geringer wird. Für dieses Beispiel erscheint es anhand der bereits erzielten Ergebnisse durchaus möglich zu sein, das Prognosemodul soweit zu verbessern, dass keine prozentualen Abweichungen auftreten, die größer als $\pm 20\%$ sind. Es wird nie gänzlich auszuschließen sein, dass ein solches Prognosemodul nicht die tatsächliche optimale, sondern eine suboptimale Parametrierung vorschlägt. Bei einer Prognosegenauigkeit, die sich in der oben skizzierten

Bandbreite bewegt, dürfte dies aber für industrielle Anwender oftmals durchaus ausreichend sein. Insbesondere auch dann, wenn man den u. U. nicht unerheblichen Zeitgewinn gegenüber einer manuellen oder batchartigen Modellgenerierung bedenkt.

4 Zusammenfassung und Ausblick

Zusammenfassend lässt sich feststellen, dass es tatsächlich möglich erscheint, sowohl wissensbasiert als auch datenbasiert Regeln für die problemspezifische Einstellung freier Modellparameter des Fuzzy-ROSA-Verfahrens aufzustellen. Der hier verfolgte Ansatz zielt auch darauf ab, die Ergebnisse, die bei der Erprobung des datenbasiert generierten Moduls gewonnen werden, für dessen Weiterentwicklung zu nutzen. Die Kenngrößen eines untersuchten Problems und die Modellierungsfehler der hierzu real erzeugten Modelle werden dabei der bereits vorhandenen Datenbasis hinzugefügt und das Prognosemodul auf der so erweiterten Grundlage neu erstellt. Weiteres Verbesserungspotenzial dürfte in der Entwicklung weiterer Ansätze für die geeignete Bildung und Auswahl von Problemmerkmalen liegen.

Danksagung

Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft im Rahmen des Transferbereiches "*Datenbasierte Generierung und Optimierung von Modellen mit Methoden der Computational Intelligence*" (37) gefördert.

Literatur

- [1] Mamdani, E. H.; Gaines, B. R.: *Fuzzy Reasoning and its Applications*. London: Academic Press. 1981.
- [2] Takagi, T.; Sugeno, M.: Fuzzy Identification of Systems and its Application to Modeling and Control. *IEEE Transactions on Systems, Man, and Cybernetics* 15 (1985) 1, S. 116–132.
- [3] Kiendl, H.; Krabs, M.: Ein Verfahren zur Generierung regelbasierter Modelle für dynamische Systeme. *at – Automatisierungstechnik* 37 (1989) 11, S. 423–430.
- [4] Krone, A.; Kiendl, H.: Automatic Generation of Positive and Negative Rules for Two-Way Fuzzy Controllers. In: *Proceedings of the Second European Congress on Intelligent Techniques and Soft Computing (EUFIT '94)*, Bd. 1, S. 438–447. Aachen: Verlag Mainz. 1994.
- [5] Slawinski, T.; Krone, A.; Krause, P.; Kiendl, H.: The Fuzzy-ROSA Method: A Statistically Motivated Fuzzy Approach for Data-Based Generation of Small Interpretable Rule Bases in High-Dimensional Search Spaces. In: *Data Mining and Computational Intelligence* (Last, M.; Kandel, A.; Bunke, H., Hg.), S. 141–166. Heidelberg: Physica-Verlag. 2001.

- [6] Krause, P.: *Datenbasierte Generierung von transparenten und genauen Fuzzy-Modellen für mehrdeutige Daten und komplexe Systeme*. Fortschritt-Berichte VDI, Reihe 10, Nr. 691. Düsseldorf: VDI Verlag. 2002.
- [7] Schauten, D.; Nicolaus, B.; Kiendl, H.: Evolutionäres Verfahren zur Selektion relevanter Merkmalsätze für die datenbasierte Fuzzy-Modellierung. In: *Proceedings 11. Workshop Fuzzy Control des GMA-FA 5.22*, S. 133–147. VDI/VDE GMA-FA 5.22, Karlsruhe: Wissenschaftliche Berichte Forschungszentrum Karlsruhe (FZKA 6509). 2001.
- [8] Krause, P.; Krone, A.; Slawinski, T.: Fuzzy System Identification by Generating and Evolutionary Optimizing Fuzzy Rule Bases Consisting of Relevant Fuzzy Rules. Techn. Ber., SFB 531, Universität Dortmund. Reihe CI 84/00. 2000.
- [9] McQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings 5th Berkely Symposium on Mathematical Statistics and Probability*, S. 281–298. 1967.
- [10] Hartigan, J. A.; Wong, M. A.: Algorithm AS 136: A K -Means Clustering Algorithm. *Applied Statistics* 28 (1979) 1, S. 100–108.
- [11] Sternke, M.; Kiendl, H.; Krause, P.; Schwefel, H.-P.: Transferprojekt T1: Datenbasierte Generierung und Optimierung von Fuzzy-Modellen mit dem Fuzzy-ROSA-Verfahren für komplexe Systeme und Prozesse. In: *SFB 531: Design und Management komplexer technischer Prozesse und Systeme mit Methoden der Computational Intelligence, Finanzierungsantrag 2006-2008*, S. 647–674. Bonn, Deutschland: Deutsche Forschungsgemeinschaft. 2005.
- [12] Krone, A.: *Datenbasierte Generierung von relevanten Fuzzy-Regeln zur Modellierung von Prozesszusammenhängen und Bedienstrategien*. Fortschritt-Berichte VDI, Reihe 10, Nr. 615. Düsseldorf: VDI Verlag. 1999.
- [13] Jessen, H.: *Test- und Bewertungsverfahren zur regelbasierten Modellierung und Anwendung in der Lastprognose*. Fortschritt-Berichte VDI, Reihe 8, Nr. 836. Düsseldorf: VDI Verlag. 2000.
- [14] Kiendl, H.: Implizite Modellierung, inkrementeller Relevanzindex und Rauigkeitsmaß: neue Strategieelemente für die datenbasierte Modellierung. In: *Proceedings 10. Workshop Fuzzy Control des GMA-FA 5.22*, S. 1–14. VDI/VDE GMA-FA 5.22, Karlsruhe: Wissenschaftliche Berichte Forschungszentrum Karlsruhe (FZKA 6509). 2000.
- [15] Slawinski, T.: *Analyse und effiziente Generierung von relevanten Fuzzy Regeln in hochdimensionalen Suchräumen*. Fortschritt-Berichte VDI, Reihe 10, Nr. 686. Düsseldorf: VDI Verlag. 2001.
- [16] Sternke, M.; Krause, P.; Kiendl, H.: Erweiterung des Fuzzy-ROSA-Verfahrens für die datenbasierte Regelgenerierung bei kleinen und stark gestörten Datensätzen. In: *Proceedings 14. Workshop Fuzzy-Systeme und Computational Intelligence*, S. 55–68. VDI/VDE GMA-FA 5.22 und GI, Karlsruhe: Universitätsverlag Karlsruhe. 2004.

- [17] Rechenberg, I.: *Evolutionsstrategie '94, Werkstatt Bionik und Evolutionstechnik*, Bd. 1. Stuttgart: Frommann–Holzboog. 1994.
- [18] Schwefel, H.-P.: *Evolution and Optimum Seeking*. New York, USA: Sixth–Generation Computer Technology Series. 1995.
- [19] Kennedy, J.; Eberhart, R.: Particle Swarm Optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, Bd. IV, S. 1941–1948. Piscataway, NJ, USA: IEEE-Press. 1995.

Klassifikation von Zeitreihen und Sequenzen

Frank Höppner

Fachhochschule Braunschweig/Wolfenbüttel
Robert Koch Platz 10-14, 38440 Wolfsburg
Tel.: (05361) 83 1539
E-Mail: f.hoepfner@fh-wolfenbuettel.de

Zusammenfassung

Bei Klassifikationsproblemen, deren Klassenattribut von der *zeitlichen Entwicklung* oder Historie anderer Attribute abhängt (Krankheitsgeschichte, Kursverlauf, Schaltverhalten, etc.), sind Intervallsequenzen als Basis für die Formulierung von Bedingungen an die Historie besonders gut geeignet. Die meisten bekannten Ansätze zur Analyse von Intervallsequenzen sind in der Ausdrucksfähigkeit stark limitiert, weil der Raum der möglichen Intervallmuster eine immense Größe besitzt. In diesem Papier wird ein Klassifikator vorgeschlagen, der gegenüber existierenden Ansätzen eine mächtigere Beschreibung der Intervallbeziehungen zulässt.

1 Einführung

Unter einem Klassifikator versteht man ein Verfahren, das aus den Ausprägungen bekannter Attribute A_i (Spalten einer Datenbank) den Wert eines bestimmten Attributs K vorhersagt (Klassenattribut), das in der realen Anwendung sonst nicht oder zu spät verfügbar ist. Beispiele für Klassifikatoren sind Entscheidungsbäume, Support Vector Maschinen, der Naive Bayes'sche Klassifikator, Neuronale Netze, etc. [1]. Induktive Lernverfahren entwickeln Klassifikatoren aus Beispielen für die Ausprägungen der Attribute A_i und K . In dieser Arbeit geht es ebenfalls um Klassifikation, allerdings soll diese anhand der *Historie zeitlich veränderlicher Attribute* erfolgen. Eine solche Historie kann z.B. durch die Bindung eines Kunden an bestimmte Produkte (etwa Versicherungen), durch die Medikation eines Patienten mit verschiedenen Arzneimitteln, durch die Betriebszeiten verschiedener Geräte oder durch die Kennlinien bestimmter Messgrößen über der Zeit gegeben sein. Während in der üblichen, statischen Klassifikation jede Datenbankzeile einen neuen Fall darstellt (Kunde, Patient, Maschine, etc.) sind nun mehrere Ereignisse zu einer Historie zusammengefasst (alle Bestellungen eines Kunden, alle Arzneiverordnungen für einen Patienten, alle Kennlinien einer Maschine, etc.).

Eine ähnliche Situation liegt z.B. bei der Warenkorbanalyse eines Kunden vor [2], bei der eine einzelne Datenbankzeile nicht mehr aus einem Kunden oder einem Produkt besteht, sondern aus einer Menge von gekauften Waren. Hier werden i.a. so viele binäre Attribute eingeführt, wie es verschiedene Waren gibt, die als Indikatoren für den Kauf dienen. Eine Klassifikationsregel kann dann aus einer Menge von Produkten bestehen, die charakteristisch für den Erwerb eines anderen Produkts ist. Soll die Abfolge mehrerer Einkäufe berücksichtigt werden, so reicht die Indikator-Darstellung nicht mehr aus: Die einzelnen Einkäufe sind dann mit einem Zeitstempel versehen, anhand dessen eine (halbgeordnete) Sequenz erzeugt wird, die zum Gegenstand der Analyse wird [3]. In diesem Fall besteht eine Klassifikationsregel aus einer Abfolge von bestimmten Ereignissen (z.B. Käufen), die ihrerseits ein Folgeereignis (z.B. Folgekauf) nahelegen.

Diese verbreiteten Repräsentationen zeitlicher Daten und Muster reichen jedoch nicht aus, um allgemeinere Datenhistorien zu erfassen. Beobachten wir den Zustand einer Informationseinheit von einem Bit (ein/aus) über der Zeit, so ergeben sich durch Zustandswechsel Intervalle, in denen der Wert 0 bzw. 1 angenommen wird. Diese Zustandswechsel können in eine Folge von Ereignissen überführt werden, zum Beispiel 'Beginn der Ein-Phase' und 'Ende der Ein-Phase'. Wenn es nun mehrere dieser binären Attribute gibt und es für ein Fehlerereignis charakteristisch ist, dass die Geräte A, B, C in dieser Reihenfolge eingeschaltet werden, aber in der Reihenfolge A, C, B ausgeschaltet werden, so könnte man dies durch nachstehende Ereignisfolge zu erfassen versuchen:

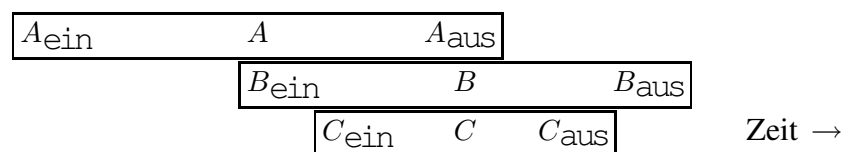
$$A_{\text{ein}} \ B_{\text{ein}} \ C_{\text{ein}} \ A_{\text{aus}} \ C_{\text{aus}} \ B_{\text{aus}}$$

Die in der Literatur bekannten Verfahren zur Analyse von Ereignissequenzen benutzen die Teilfolgen-Eigenschaft als Indikator für das Auftreten einer Ereignisfolge. Daher würde das obige Muster auch in

$$A_{\text{ein}} \ B_{\text{ein}} \ A_{\text{aus}} \ C_{\text{ein}} \ A_{\text{ein}} \ A_{\text{aus}} \ C_{\text{aus}} \ B_{\text{aus}}$$

identifiziert. Hierbei handelt es sich aber nicht mehr um eine Sequenz, die der Intention 'Geräte A,B,C werden in dieser Reihenfolge eingeschaltet' entspricht, weil A vor dem Einschalten von C bereits wieder ausgeschaltet und erst danach wieder in Betrieb genommen wurde. Einfache Sachverhalte lassen sich also auf diese Weise nicht semantisch einwandfrei umsetzen. Der Grund dafür liegt darin, dass das Ein- und das zugehörige Ausschalten durch zwei unabhängige Ereignisse modelliert wurden. So kann nicht sichergestellt werden, dass sich das vierte Ereignis A_{aus} in der Originalsequenz auf das erste Ereignis A_{ein} bezieht, d.h., keine weiteren A-Ereignisse in der Zwischenzeit auftreten dürfen.

Dieser inhaltliche Zusammenhang kann erreicht werden, indem eine Betriebsphase eben nicht durch zwei unabhängige Ereignisse, sondern als ein Betriebsintervall aufgefasst wird, wie in folgender Abbildung durch drei Intervalle dargestellt. Die ursprüngliche Ereignissequenz ergibt sich aus der Projektion der Intervallendpunkte auf die Zeitachse.



Wenn es gelingt, die (komplexeren) Beziehungen zwischen den Intervallen (statt der Halbordnung der Ereignisse) formal zu erfassen, um daraus Klassifikatoren zu erstellen, so ist der zuvor erwähnte Sachverhalt ohne weiteres induktiv lernbar. Statt Ereignisfolgen betrachten wir daher Intervallsequenzen. Diese haben zusätzlich den Vorteil, dass sie auch Attribute aus anderen Skalen geeignet repräsentieren können, wie in Bild 1 angedeutet. Diese Tatsache wiederum erlaubt eine Gleichbehandlung verschiedener Attribute in einer einheitlichen Repräsentation, wodurch sich Spielraum für Muster wie 'Anstieg des Pegels bei geschlossenem Ventil außerhalb der Wartungsphase' ergibt.

Die weiteren Abschnitte gliedern sich wie folgt: Zunächst werden in Abschnitt 2 Intervallsequenzen etwas genauer formalisiert, und die Stärken und Schwächen einiger früherer

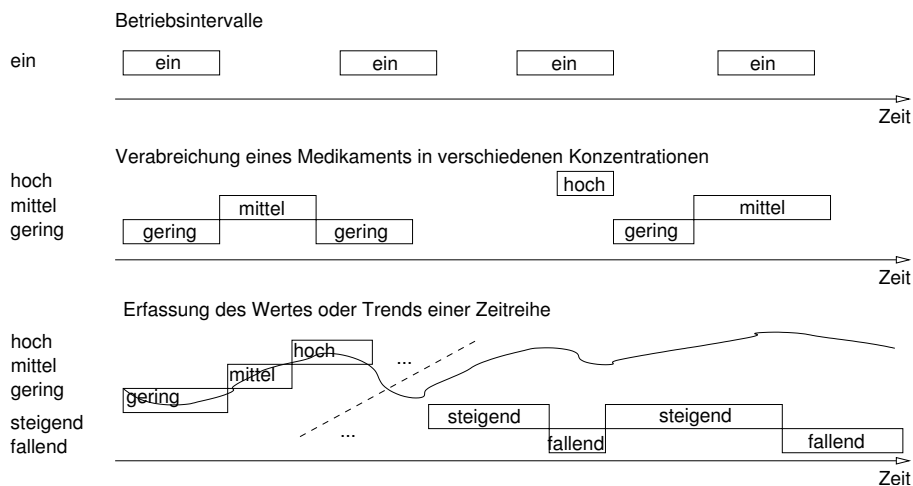


Bild 1: Unabhängig von der Skala eines Attributs läßt sich der zeitliche Verlauf immer geeignet als Intervallsequenz beschreiben.

Arbeiten heraus gearbeitet. So läßt sich als Beispiel 'A, B und C werden in dieser Reihenfolge eingeschaltet, aber B und C werden in umgekehrter Reihenfolge ausgeschaltet' aufgrund er fehlenden Information über den Ausschaltzeitpunkt von A nicht formulieren. In Abschnitt 3 wird daher ein Vorschlag für eine flexiblere Festlegung der in einem Muster erlaubten Intervallbeziehungen gemacht, der diesen Mangel behebt. Der Abschnitt mündet in einer Modifikation des ITRule-Klassifikators. Abschnitt 4 schließt mit einem Fazit.

2 Analyse von Intervallsequenzen

Eine Intervallsequenz ist eine Folge von Tripeln $(l_i, s_i, e_i)_{i=1..n}$, wobei ein Tripel (l_i, s_i, e_i) für ein Intervall $[s_i, e_i]$ mit Beschriftung (Label) l_i steht. Die Beschriftungen stammen aus einer domänenabhängigen, endlichen Menge L , z.B. $L = \{\text{hoch, mittel, niedrig, steigend, fallend}\}$. Ein Datensatz (record) besteht nun aus einer Historie in Form einer Intervallsequenz (z.B. Patientenakte, Schaltzeiten, Kursentwicklung, etc.) sowie der Klassifizierung dieser Historie (Therapie erfolgreich/nicht erfolgreich, Zustand fehlerhaft/ok, etc.).

Eine Intervallsequenz kann (wie bei der Warenkorbanalyse) in einen klassischen Datensatz überführt werden, indem Indikator-Attribute A_l mit $l \in L$ erzeugt werden, die genau dann wahr werden, wenn es ein Tripel (l, s, e) in der aktuellen Historie gibt. Für die Erfassung der relativen Lage der Intervalle zueinander sind die Indikatorvariablen nicht geeignet. Die Beziehung zweier Intervalle untereinander kann entweder durch weitere numerische Attribute (wie die Zeitdauer zwischen dem Ende von A und dem Anfang von B) oder nominale Attribute (Allen's Intervalbeziehungen [4], siehe Bild 2) eingeführt werden. Im letzteren Fall steht in einer gegebenen Historie jedes Intervall mit jedem anderen Intervall der Historie in genau einer Beziehung (*before, meets, overlaps, etc.*), daher können wir die Gesamtheit der Beziehungen in einer Matrix erfassen, in der die Matrixelemente einen aus 13 möglichen Beziehungstypen angeben. Um auszudrücken, dass zwei Intervalle A und B in der Beziehung *before* oder *after* stehen, vereinbaren wir die Schreibweise $A\langle b, a\rangle B$, wobei b und a die (in Bild 2 in eckigen Klammern notierten) Abkürzungen von *before* und *after* darstellen. Die Menge aller 13 *elementaren Intervallbeziehungen* aus Bild 2 bezeichnen wir mit \mathcal{I} .

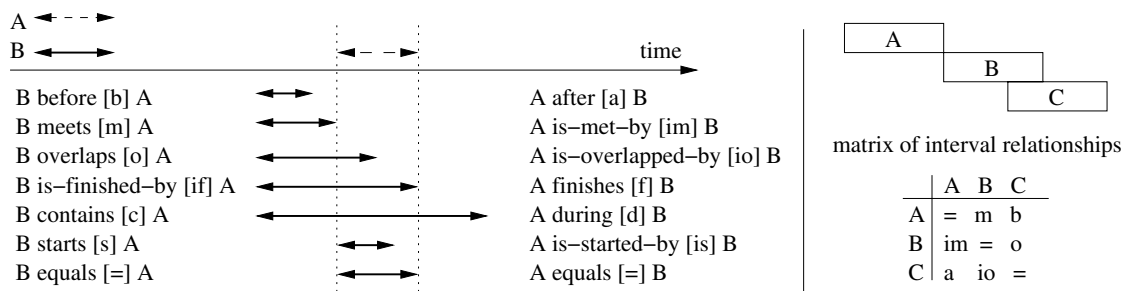


Bild 2: Allen's qualitative Intervall-Beziehungen: Zwei beliebige Intervalle stehen stets in einer der 13 genannten Beziehungen zueinander.

Existierende Ansätze zur Analyse von Intervallsequenzen unterscheiden sich in der Art und Weise, mit der die Intervallbeziehungen berücksichtigt werden. In [5] wird immer nur die Beziehung eines Intervalls zur Vereinigung aller vorherigen in der Regel benutzten Intervalle betrachtet. In [6] wird nur die Enthaltensein-Beziehung genauer betrachtet. In [7] wird der Raum der Intervallmuster durch die oben erwähnte Matrix-Schreibweise erfasst und über Assoziationsanalyse werden alle Regeln in diesem Suchraum aufgezählt.

Während die Mächtigkeit der Suchräume in den beiden zuerst genannten Verfahren stark begrenzt ist, erlaubt die Verwendung der Beziehungsmatrix die volle Kontrolle über alle paarweisen Beziehungen. Die hohe Ausdrucksfähigkeit hat auf der anderen Seite aber auch Schwächen (die zum Teil auf den verwendeten Basis-Techniken beruhen und daher von mehreren Ansätzen geteilt werden, die auf gleichen Algorithmen beruhen):

- Es sind – wie meistens im Umfeld von Assoziationsregeln – keine Aussagen über das Nichtauftreten von Produkten, Ereignissen oder eben Intervallen möglich. Der Ausschluss von Intervallen kann wertvolle Dienste leisten, etwa wenn bestimmte Muster auch während Wartungsphasen auftreten (dort aber Teil der Wartungsprozedur sind), außerhalb der Wartung aber nicht auftreten sollten (z.B. Klasse K tritt nur auf, wenn Intervall X *nicht* beobachtet wird).
- Bei zeitlichen Abläufen treten Translations- und Dilatationseffekte auf, d.h. im Wiederholungsfall können Intervalle leicht verschoben oder gestaucht/gestreckt vorkommen. Bis zu einem gewissen Grad ist die Erfassung eines Intervallmusters durch die Allen'schen Beziehungen robust gegen solche Änderungen, eine andere Beziehungsmatrix ergibt sich erst, wenn Intervallgrenzen überschritten werden. Je größer jedoch das Intervallmuster ist (d.h., je mehr Intervalle beteiligt sind), desto größer ist die Wahrscheinlichkeit für diesen Fall.
- Die Beziehungsmatrix erfasst **alle** paarweisen Beziehungen, auch wenn manche für die Klassifikation vielleicht unerheblich sind. So könnte es sein, dass zwei Intervalle A und B während eines dritten Intervalls C auftreten müssen, wobei aber die relative Lage von A und B zueinander unerheblich ist. Muss die gesamte Beziehungsmatrix mit elementaren Intervallbeziehungen gefüllt werden, so lässt sich diese Bedingung nicht formulieren.
- Die Beziehungsmatrix erfasst **nur elementare** Beziehungen, disjunktive Kombinationen sind nicht möglich. Eine Beziehung der Art ' A und B dürfen nicht gleichzeitig auftreten ' können nicht durch eine elementare Beziehung erfasst werden, wohl

aber durch die Disjunktion $A\langle b, a\rangle B$. In [8] wurde der Versuch unternommen, Regeln disjunktiv zu kombinieren, der allerdings erheblichen Rechenaufwand erforderte.

3 Erhöhung der Beziehungs-Flexibilität

Das Ziel dieses Abschnittes ist es, einen Klassifikator zu entwickeln, der auf der einen Seite die Ausdrucksfähigkeit der Beziehungsmatrix nicht verliert, auf der anderen Seite aber die zuvor genannten Schwächen nicht aufzeigt. Die Grundidee ist, die Beziehungsmatrix nicht auf elementare Beziehungen festzulegen, sondern anfangs beliebige Beziehungen zuzulassen. (Das entspricht einer Initialisierung jedes Matrixelements mit der Disjunktion aus allen 13 Intervallbeziehungen.) Die möglichen Intervallbeziehungen werden nur explizit eingeschränkt, wenn dadurch die Güte einer Vorhersageregel verbessert werden kann.

Diese Grundidee läßt sich in bestehende Regel- oder Entscheidungsbaum-Lernverfahren integrieren. Während traditionell die Menge der zur Verfügung stehenden Attribute bei einer (Regel- oder Baum-) Verfeinerung von Anfang an bekannt ist und sich nur dadurch ändert, dass ein bereits benutztes Attribut nicht erneut benutzt werden sollte, haben wir es nun mit einer veränderlichen Attributmenge zu tun. Sobald in einer Regel zwei Intervalle bereits vorhanden sind, kann als Verfeinerung eine bestimmte Beziehung zwischen diesen Intervallen gefordert werden (vgl. Bild 3).

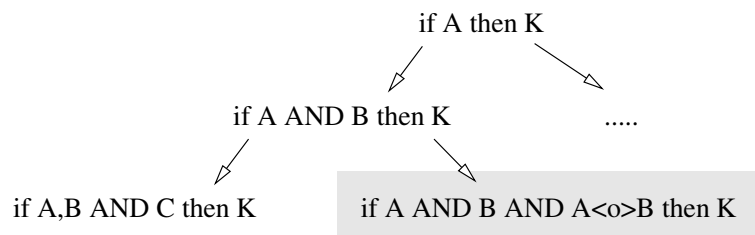


Bild 3: Intervallbeziehungen eröffnen neue Möglichkeiten der Regelverfeinerung (schattiert).

Für die Verfeinerung der Intervallbeziehungen sind mindestens zwei Ansätze denkbar. Ist ein Intervall A mit Grenzen $[A_{\text{ein}}, A_{\text{aus}}]$ und B mit Grenzen $[B_{\text{ein}}, B_{\text{aus}}]$ gegeben, so kommen eine Reihe von *numerischen* Attributen $A_{\text{ein}} - B_{\text{ein}}, A_{\text{ein}} - B_{\text{aus}}, A_{\text{aus}} - B_{\text{ein}},$ etc in Betracht, um die relative Lage der Intervallendpunkte quantitativ zu erfassen. Damit lassen sich Bedingungen der Art $A_{\text{ein}} - B_{\text{ein}} > \vartheta$ fordern. Die andere Möglichkeit besteht in der Einführung eines mengenwertigen Attributs, das die erlaubten Intervallbeziehungen zwischen A und B nennt. Entsprechende Bedingungen haben die Form $A\langle R\rangle B$ mit $R \subseteq \mathcal{I}$.

Nehmen wir an, die Lage eines Intervalls X ist für die Klassifikation von Bedeutung.

- *qualitativer Ansatz über Intervallbeziehungen:* Verwendet eine Regel bereits n Intervalle, so wird die Lage von X zu den anderen Intervallen in n Elementen der Beziehungsmatrix codiert. Für jedes dieser Elemente gibt es 13 mögliche elementare Ausrichtungen, wovon theoretisch jede beliebige Teilmenge charakteristisch sein kann (2^{13} Möglichkeiten). Ein blindes Ausprobieren aller Möglichkeiten muss an ihrer Anzahl scheitern.

- *quantitativer Ansatz über Intervallendpunkte*: Für jede der n Intervallkombinationen haben wir 4 numerische Attribute zu untersuchen. Eine übliche Technik in Entscheidungsbaumverfahren ist ein binäres Splitting [9]. Dabei ist die Bestimmung des Schwellwertes mit erheblichem Aufwand verbunden (u.a. müssen je Verfeinerung alle Werte gespeichert und sortiert werden).

Die Aussagekraft beider Vorgehensweisen ist nicht identisch. Kehren wir noch einmal zum Beispiel aus der Einleitung zurück, bei dem u.a. die Beziehung $B\langle e\rangle C$ (B contains C) vorkam, die es nun von einem induktiven Lernverfahren zu entdecken gilt. Wenn wir a priori wissen, dass diese Beziehung vorliegt, dann folgt natürlich, dass wir in jedem Fall $B_{\text{aus}} - C_{\text{aus}} > 0$ und $C_{\text{ein}} - B_{\text{ein}} > 0$ erhalten. Anderherum, wenn es diese *contains*-Beziehung erst noch zu entdecken gilt, folgt aus positiven Schwellwerten für die beiden Endpunktabstände nicht notwendig, dass eine *contains*-Beziehung vorliegt: Auch wenn die Beziehungen $B\langle o\rangle C$ und $B\langle io\rangle C$ gleichermaßen Indikatoren für die Klasse sind, würden wir jeweils eine der beiden Bedingungen $B_{\text{aus}} - C_{\text{aus}} > 0$ und $C_{\text{ein}} - B_{\text{ein}} > 0$ erhalten. Die Regeln, die wir aus dem qualitativen Ansatz erhalten, sind damit eindeutiger und leichter zu interpretieren.

Wir wollen daher den qualitativen Ansatz über die Intervallbeziehungen verfolgen und durch eine Heuristik die Möglichkeiten zur Verfeinerung sinnvoll einschränken. Dazu ermitteln wir, wie häufig zwei Intervalle in welcher Intervallbeziehung zueinander standen (in Abhängigkeit vom Klassenattribut). Im folgenden Abschnitt 3.1 beschäftigen wir uns damit, wie diese Zählung geeignet durchzuführen ist. Anschließend betrachten wir in Abschnitt 3.2, wie auf Basis dieser Zählungen geeignete Regelverfeinerungen zu finden sind.

3.1 Häufigkeit von Intervallbeziehungen

In diesem Abschnitt gehen wir von einer Regel aus, die mindestens zwei Intervalle A und B verwendet, aber noch keine Einschränkungen hinsichtlich ihrer Lage besitzt. Wir interessieren uns für die Menge von Historien, in denen A und B in einer Beziehung aus der Menge $R \subseteq \mathcal{I}$ vorkommen. Unter diesen Fällen zählen wir, wie oft die Konklusion der Regel erfüllt ist und bezeichnen diese Anzahl mit p_R (positive Fälle) bzw. n_R (negative Fälle). Wir gehen nun der Frage nach, welche Informationen wir ermitteln müssen, um eine geeignete Spezialisierung der Beziehung zwischen A und B ermitteln zu können.

Für die Spezialisierung der betrachteten Regel mit *einer* elementaren Beziehung $r \in \mathcal{I}$ genügt die Erstellung von zwei Histogrammen mit $|\mathcal{I}|$ Einträgen (für p_R und n_R mit $R = \{r\}$, $r \in \mathcal{I}$). Über p_R und n_R lassen sich viele gängige Regelmaße wie Support, Konfidenz, etc. bestimmen. Durch Vergleich der Regelmaße für alle $r \in \mathcal{I}$ kann die – je nach Regelmaß – optimale Spezialisierung ausgewählt werden. Damit haben wir aber nur Informationen für *eine* elementare Beziehung.

Wollen wir *mehrere* Beziehungen erlauben, zum Beispiel $A\langle b, m, o\rangle B$, so benötigen wir die absoluten Häufigkeiten für beliebige Beziehungsmengen $R \subseteq \mathcal{I}$. Da es 2^{13} Teilmengen R gibt, können wir nicht alle diese Beziehungen auszählen. Stehen uns für die Beurteilung einer Verfeinerung nur die absoluten Häufigkeiten der elementaren Beziehungen zur Verfügung, so können wir die Häufigkeit n_R abschätzen (p_R analog):

$$\min\{n_{\{r\}} | r \in R\} \leq n_R \leq \sum_{r \in R} n_{\{r\}} \quad (1)$$

Wie weit obere und untere Grenze auseinander liegen, hängt davon ab, in wievielen Historien Intervalle A und B in *mehreren* Beziehungen stehen. Es ist möglich, dass es in einer Historie mehrere A- und/oder B-Intervalle gibt, so dass diese auch in mehrere Beziehungen zueinander stehen. Die Wahrscheinlichkeit dafür ist abhängig von der Beschaffenheit der Intervallsequenz selbst (anwendungsabhängig) und der Länge der Historien.

Der Rest des Abschnitts gilt für p_R und n_R gleichermaßen, der Einfachheit halber betrachten wir aber nur n_R . Statt n_R direkt auszuzählen können wir äquivalent in m_R die Anzahl der Historien zählen, in denen *jede* der Beziehungen in R beobachtet werden konnte (in n_R zählen wir die Historien, in denen *wenigstens eine* der Beziehungen vorkam). Dann läßt sich bspw. $n_{\{r,s\}}$ wie folgt ermitteln (aus $|A \cup B| = |A| + |B| - |A \cap B|$)

$$n_{\{r,s\}} = n_{\{r\}} + n_{\{s\}} - m_{\{r,s\}} \quad r, s \in \mathcal{I} \quad (2)$$

Da sich die Anzahl der Teilmengen $R \subseteq \mathcal{I}$ nicht verändert hat, brauchen wir ebensoviele m_* -Zähler wie n_* -Zähler. Aber selbst wenn wir m_R nur für zweielementige Beziehungsmengen ermitteln, können wir bereits bessere Abschätzungen für n_R mit $|R| = 3$ geben, als in (1). Für dreielementige Beziehungsmengen machen wir dann nur einen Fehler von $m_{\{r,s,t\}}$, weil wir $n_{\{r,s,t\}}$ bis auf den Summanden $m_{\{r,s,t\}}$ bestimmen können.

$$n_{\{r,s,t\}} = n_{\{r\}} + n_{\{s\}} + n_{\{t\}} - m_{\{r,s\}} - m_{\{r,t\}} - m_{\{s,t\}} + m_{\{r,s,t\}} \quad r, s, t \in \mathcal{I}$$

Bedenken wir, dass $m_{\{r,s,t\}}$ zählt, wie oft A und B in *einer* Historie *sowohl* in der Beziehung s , r als auch t vorkommen, so können wir davon ausgehen, dass die Zähler m_R – und damit auch der Fehler, den wir machen – mit zunehmender Mächtigkeit von R immer kleiner werden. Wir begnügen uns daher mit der Ermittlung von m_R für kleine Mengen R , z.B. $|R| = 2$.

Eine weitere Möglichkeit zur Aufwandsreduktion ergibt sich, wenn die Intervallgrenzen z.B. aus mehreren Zeitreihen automatisch extrahiert werden. Dann sind exakte Übereinstimmungen von Intervallgrenzen eher selten, die Beziehungen *meets*, *starts*, *finishes* etc. treten daher nur sehr selten (und kaum reproduzierbar) auf. Unter solchen Bedingungen macht die detaillierte Betrachtung dieser Beziehungen wenig Sinn, stattdessen könnten die Intervallbeziehungen wie folgt gruppiert werden: $\{b, m\}$, $\{o, f\}$, $\{c, =\}$, $\{s, d, is\}$, $\{io, if\}$, $\{im, a\}$. Diese Beziehungsmengen können dann an die Stelle der bisherig als elementar angesehenen Beziehungen treten. In diesem Fall wären insgesamt nur 21 Zähler erforderlich (6 Zähler n_R für die nunmehr 'elementaren' Beziehungsmengen, 15 Zähler m_R , wobei R sich aus einer paarweisen Kombination ergibt).

3.2 Erweiterung des ITRULE-Regellerners

Wir betrachten exemplarisch die Erweiterung des ITRule-Regellerners [10] um Intervall-Historien. Der Algorithmus lernt die k besten Regeln zur Vorhersage eines Klassenattributs. Dabei wird mit einfachen Regeln begonnen, die solange verfeinert (um Bedingungen in der Prämisse ergänzt) werden, wie es der Regelqualität dienlich ist. Der Algorithmus verwendet das J-Maß zur Bewertung einer Regel sowie eine Abschätzung des maximalen J-Wertes, der durch Spezialisierung einer Regel noch erreicht werden kann. Auf diese Weise wird der Suchraum eingeschränkt, um eine effiziente Berechnung zu ermöglichen.

Betrachten wir also eine Regel 'Wenn $Y=y$, dann $X=x$ ', wobei X und Y Zufallsvariablen für die Erfüllung der Prämisse und Konklusion sind. Das J-Maß betrachtet den mittleren

Informationsgewinn über $X = x$ durch die Erfüllung der Prämisse $Y = y$: es vergleicht die a priori Verteilung von X mit der a posteriori Verteilung von $X|Y$ über die Kullbach-Leibler Distanz:

$$j(X|Y = y) = \sum_{z \in \{x, \bar{x}\}} P(X = z|Y = y) \log_2 \left(\frac{P(X = z|Y = y)}{P(X = z)} \right)$$

Aus $j(X|Y = y)$ erhalten wir die unmittelbare Information, die $Y = y$ über X liefert. Eine Regel ist umso nützlicher, je öfter wir sie anwenden können. Im Mittel nützt uns eine Regel also $J(X|Y = y) = P(Y = y) \cdot j(X|Y = y)$ (J-Maß).

Wie eingangs erwähnt, gibt es im Fall der Intervallsequenzen zwei Arten der Spezialisierung: zum einen kann die Beobachtung eines weiteren Intervalls gefordert werden (ohne Anforderungen an die Lage) und zum anderen kann für zwei bereits in der Regel benutzten Intervalle eine Bedingung an ihre relative Lage eingeführt werden. Für diesen zweiten Fall ist zu klären, welche Intervallbeziehungen diese Bedingung enthalten soll, damit der J-Wert der spezialisierten Regel möglichst optimal wird.

In unserem speziellen Kontext betrachten wir nur die Verfeinerung einer Regel, also ist $P(X = z)$ vor und nach der Verfeinerung gleich. Durch die Spezialisierung ändern sich nur zwei Größen: die Anwendbarkeit der Regel $P(Y = y)$ und die Korrektheit der Regel $P(X = x|Y = y)$. Führen wir die Abkürzungen $P_y = P(Y = y)$, $P_x = P(X = x)$ und $P_{xy} = P(X = x|Y = y)$ ein, so können wir wegen $P(X = \bar{x}) = 1 - P(X = x)$ und $P(X = \bar{x}|Y = y) = 1 - P(X = x|Y = y)$ das J-Maß wie folgt notieren¹:

$$P_y \left(P_{xy} (\log_2(P_{xy}) - \log_2(P_x)) + (1 - P_{xy}) (\log_2(1 - P_{xy}) - \log_2(1 - P_x)) \right)$$

Durch die Spezialisierung ändern sich die Terme $\log_2(P_x)$ und $\log_2(1 - P_x)$ nicht. Die Summe $P_{xy} \log_2(P_{xy}) + (1 - P_{xy}) \log_2(1 - P_{xy})$ ist für $P_{xy} = \frac{1}{2}$ minimal und erreicht bei $P_{xy} = 1$ oder $P_{xy} = 0$ ihr Maximum. Aus Sicht der Regelsemantik (Vorhersage von $X = x$) ist allerdings der Fall $P_{xy} = 1$ anzustreben. Der Wert P_y ist als J-limitierender Faktor ebenfalls zu maximieren.

	$X = x$	$X = \bar{x}$
$Y = y$	a	b
$Y = \bar{y}$	c	d

Bild 4: Kreuztabelle für die Berechnung des J-Maßes.

Welche Einflußmöglichkeiten haben wir bei der Spezialisierung? Bild 4 zeigt die beteiligten Größen in einer Kreuztabelle. Darin ist $P_y = \frac{a+b}{a+b+c+d}$ und $P_{xy} = \frac{a}{a+b}$. Während vor der Spezialisierung die absolute Häufigkeiten a, b, c, d gezählt wurden, treten nach der Spezialisierung an die Stelle von a und b die Zähler p_R und n_R (auch c und d ändern sich, da die Summe aller Werte gleich bleibt). Um P_{xy} zu maximieren, wäre also eine Beziehungsmenge R ideal, bei der $b = n_R$ möglichst klein ist (hohe Konfidenz). Das wäre bspw. auch für $p_R = 1$ und $n_R = 0$ der Fall. Da die Summe aller vier Felder immer konstant bleibt, bedeuten kleine Werte in p_R und n_R aber auch einen kleineren P_y -Wert.

¹Die Zufallsvariable X hat nur zwei mögliche Ausgänge: Entweder die Konklusion gilt ($X = x$) oder nicht ($X = \bar{x}$).

Die Maximierung von P_y erreichen wir, indem wir a und b möglichst groß wählen. Beide Ziele gleichzeitig (Maximierung von P_y und P_{xy}) erreichen wir, wenn a gegenüber b möglichst groß gewählt wird.

Wir gehen daher bei der Spezialisierung wie folgt vor: Ausgehend von der leeren Menge nehmen wir immer eine Intervallbeziehung hinzu. Diese Beziehung wählen wir so aus, dass die Größe $a - b$ aus Bild 4 maximiert wird, wodurch wir die bestmögliche Bewertung nach dem J-Maß für die jeweilige Mächtigkeit von R erhalten. Aus den erhaltenen Beziehungs-Teilmenge wählen wir für die Spezialisierung diejenige aus, die insgesamt den besten J-Wert liefert.

1. Die Regel ohne Spezialisierung liefert J_0 . Setze $R_0 = \emptyset$.
2. Für alle $i = 1..|\mathcal{I}|$: Wähle unter allen elementaren Beziehungen r diejenige aus, für die $p_{R_i} - n_{R_i}$ maximal wird, wobei $R_i = R_{i-1} \cup \{r\}$ (Maximierung von $a - b$). (Zur Berechnung von n_{R_i} siehe Abschnitt 3.1).
3. Bestimme für alle Spezialisierungen R_i die J-Werte. Sei R_* die Beziehungsmenge mit dem besten J-Wert. Wenn $J_* > J_0$, dann führe eine spezialisierte Regel ein, sonst nicht.

Statt den Raum aller möglichen $2^{|\mathcal{I}|}$ Möglichkeiten zu durchsuchen, erweitern wir, einer *gierigen* Strategie folgend, die Beziehungsmenge um die dann jeweils vielversprechendste Ergänzung.

Erste Ergebnisse. Da sich diese Entwicklungen in einem frühen Stadium befinden, können nur erste, vorläufige Ergebnisse berichtet werden. Es wurde eine künstlicher Datensatz erstellt, deren Historien aus Intervallen A , B , C und D bestehen. Das Klassenattribut wurde auf *wahr* gesetzt, wenn die 'Geräte' A , B und C in dieser Reihenfolge nacheinander *eingeschaltet* werden, so dass nach dem Einschalten von C alle Geräte laufen, aber die Reihenfolge des Ausschaltens beliebig ist. Durch die Beliebigkeit der Ausschalt-Reihenfolge reichen elementare Intervallbeziehungen nicht aus, um die Klassen-Bedingung in einer Regel auszudrücken. Statt der 13 Intervallbeziehungen wurde die Partition aus 6 Fällen benutzt, die am Ende von Abschnitt 3.1 vorgestellt wurde. Folgende Beziehungsmatrix beschreibt diesen Sachverhalt:

	A	B	C
A	$\langle = \rangle$	$\langle o, f, c \rangle$	$\langle o, f, c \rangle$
A	$\langle io, if, d \rangle$	$\langle = \rangle$	$\langle o, f, c \rangle$
A	$\langle io, if, d \rangle$	$\langle io, if, d \rangle$	$\langle = \rangle$

Dabei ist die Beziehungsmatrix redundant, weil sich die Beziehungen unterhalb der Diagonalen eindeutig aus den Beziehungen oberhalb der Diagonalen ergeben. In einer ersten Version wurden daher mehrere gleichwertige Regeln ermittelt, die z.T. Beziehungen aus dem unteren, z.T. Beziehungen aus dem oberen Teil der Matrix gefordert haben, insgesamt aber dasselbe Muster beschreiben. Der verschwendete Aufwand zur Entwicklung solcher Regel-Duplikate kann aber leicht vermieden werden, indem für die Intervall-Label L eine beliebige Ordnung etabliert wird und dann nur noch Beziehungen $l_1 \langle R \rangle l_2$ verfeinert werden, bei denen $l_1 \in L$ in der Ordnung vor $l_2 \in L$ kommt.

4 Fazit

Bei der Argumentation über zeitliche Abläufe bedient sich der Mensch eines Vokabulars, das dem der Allen'schen Intervallalgebra sehr nahe kommt. Einfachere Repräsentationen zeitlicher Abläufe, wie etwa die Darstellung in Form von Ereignisfolgen, erfordern zusätzliche Konzepte, um gleichwertige Zusammenhänge abbilden zu können. Intervallsequenzen sind daher eine gute Repräsentation für zeitliche Entwicklungen (Kundenhistorien, Patientenakten, Aktienentwicklung, etc.).

Eine Schwierigkeit bei der Analyse von Intervallsequenzen liegt in der immensen Anzahl von Möglichkeiten, die sich bei der Formulierung von Mustern in Intervallsequenzen bieten. Frühere Ansätze basieren daher meistens auf Techniken des *Association Rule Mining*, um effizient alle Muster aufzuzählen. In dieser Arbeit wurde hingegen ein klassischer Regellerner erweitert, der nun Muster in Intervallsequenzen als Prämisse entwickeln kann. Gegenüber den auf Assoziationsregeln basierenden Ansätzen wird der Anspruch auf vollständige Aufzählung aller möglichen Regeln aufgegeben, aber dadurch eine noch flexiblere Formulierung von Intervallmustern erschlossen. Erste Tests liefern vielversprechende Ergebnisse.

Literatur

- [1] Berthold, M.; Hand, D. J. (Hg.): *Intelligent Data Analysis*. Springer. 1999.
- [2] Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; Verkamo, A. I.: Fast Discovery of Association Rules. In: [11], Kap. 12, S. 307–328. MIT Press. 1996.
- [3] Mannila, H.; Toivonen, H.; Verkamo, A. I.: Discovering frequent episodes in sequences. In: *Proc. of the 1st ACM SIGKDD Int. Conf. on Knowl. Discovery and Data Mining*, S. 210–215. Menlo Park, Calif. 1995.
- [4] Allen, J. F.; Kautz, H. A.; Pelavin, R. N.; Tenenbergs, J. D.: *Reasoning about Plans*. Morgan Kaufmann Publishers. 1991.
- [5] Kam, P.-S.; Fu, A. W.-C.: Discovering Temporal Patterns for Interval-based Events. In: *Proc. of the 2nd Int. Conf. on Data Warehousing and Knowl. Discovery*, Bd. 1874 von LNCS, S. 317–326. Springer. 2000.
- [6] Villafane, R.; Hua, K. A.; Tran, D.; Maulik, B.: Knowledge Discovery from Series of Interval Events. *Journal of Intelligent Information Systems* 15 (2000) 1, S. 71–89.
- [7] Höppner, F.; Klawonn, F.: Finding Informative Rules in Interval Sequences. *Intelligent Data Analysis – An International Journal* 6 (2002) 3, S. 237–256.
- [8] Höppner, F.: Handling Feature Ambiguity in Knowledge Discovery from Time Series. In: *Proc. of 5th Int. Conf. on Discovery Science*, Nr. 2534 in LNCS, S. 398–405. Lübeck, Germany: Springer. 2002.
- [9] Quinlan, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers. 1993.

- [10] Smyth, P.; Goodman, R. M.: Rule Induction Using Information Theory. In: *Knowledge Discovery in Databases*, Kap. 9, S. 159–176. MIT Press. 1991.
- [11] Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; Uthurusamy, R. (Hg.): *Advances in Knowledge Discovery and Data Mining*. MIT Press. 1996.

Classification via Fuzzy Preference Learning

Eyke Hüllermeier and Klaus Brinker

Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg
Tel.: (0391) 6718665
Fax: (0391) 6712020
E-Mail: {huellerm, brinker}@iti.cs.uni-magdeburg.de

Abstract

This paper introduces a new approach to classification which combines pairwise decomposition techniques with ideas and tools from fuzzy preference modeling. More specifically, our approach first decomposes a polychotomous classification problem involving m classes into an ensemble of binary problems, one for each ordered pair of classes. The corresponding classifiers are trained on the relevant subsets of the original training data. Given a new instance to be classified, this instance is submitted to every binary learner, the output of which is assumed to be a score in the unit interval. The latter is interpreted as a fuzzy degree of preference for the first in comparison with the second class. By combining the outputs of all classifiers, one thus obtains a fuzzy preference matrix which is taken as a point of departure for the final classification decision. This way, the problem of classification has been reduced to a problem of decision making based on a fuzzy preference relation. Corresponding techniques, which have been investigated quite intensively in the field of fuzzy set theory, hence become amenable to the task of classification. In particular, by decomposing a preference matrix into a strict preference, an indifference, and an incomparability relation, this approach allows for quantifying different types of uncertainty in classification.

1 Introduction

As one of the standard problems of supervised learning, the performance task of classification has been studied intensively in the field of machine learning. In classification, the prediction to be made consists of a discrete class label. Thus, the problem is to learn a model that establishes an $\mathcal{X} \rightarrow \mathcal{L}$ mapping from an instance space \mathcal{X} to a finite set \mathcal{L} of class labels.

The simplest type of classification problems are *dichotomous (binary, two-class)* problems for which $|\mathcal{L}| = 2$. For such problems, a multitude of efficient and theoretically well-founded classification methods exists. In fact, the representation of models is often geared toward the binary case, and sometimes even restricted to this problem class. For example, several popular machine learning techniques, such as support vector machines [1], learn a decision boundary that can only divide the instance space into two parts, one for each class. Similarly, *concept learning*, i.e., the task of learning a description of a target concept from examples and counter-examples of the concept, may be considered as a two-class classification task. Most rule learning algorithms operate in this framework.

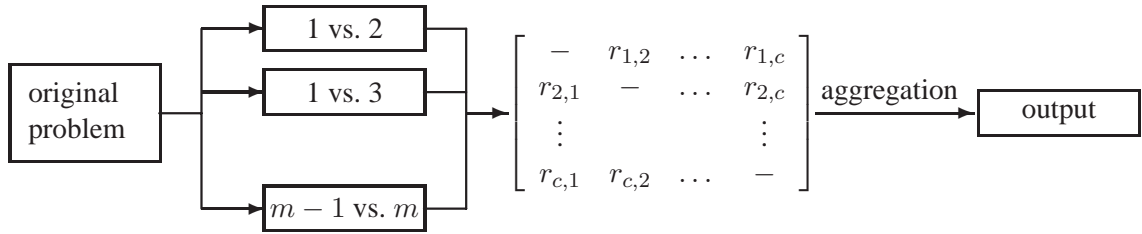


Figure 1: Basic structure of the learning by pairwise comparison approach.

Needless to say, practically relevant problems are rarely restricted to the binary case. One approach for tackling *polychotomous* (*multi-class*) problems is to develop classifiers that are able to represent an $\mathcal{X} \rightarrow \mathcal{L}$ mapping for $|\mathcal{L}| > 2$ directly, such as classification trees. An alternative strategy to tackle such problems is to transform the original problem into several binary problems via a *class binarization* technique. The most popular class binarization technique is the unordered or one-against-rest binarization, where one takes each class in turn and learns a binary concept that discriminates this class from all other classes. At prediction time, each binary classifier predicts whether the query input belongs to its concept or not. Tie breaking techniques (typically based on confidence estimates for the individual predictions) are used in case of conflicts, which may arise when more than one concept is predicted or all classifiers abstain.

The key idea of the *learning by pairwise comparison* (LPC) approach (aka pairwise classification, round robin learning, one-vs-one) is to transform an m -class problem into $m(m-1)/2$ binary problems, one for each pair of classes (Fig. 1). At classification time, a query instance is submitted to all binary models, and the predictions of these models are combined into an overall classification. In [2, 3], it was shown that pairwise classification is not only more accurate than the one-against-rest technique but that, despite the fact that the number of models that have to be learned is quadratic in the number of classes, pairwise classification is also more efficient than one-against-rest classification. The reason is that the binary decision problems not only contain fewer training examples (because all examples that do not belong to either of the classes are ignored), but that the decision boundaries of each binary problem may also be considerably simpler than for the problems generated by the one-against-rest transformation (which in turn can further increase computational efficiency in the evaluation phase).

This paper elaborates on another interesting aspect of the LPC approach: Assuming that every binary learner outputs a score in the unit interval, and interpreting this score as a “fuzzy degree of preference” for the first in comparison with the second class, the ensemble of pairwise learners produces a *fuzzy preference matrix*. The final classification decision can then be made on the basis of this matrix. This way, the problem of classification has been reduced to a problem of decision making based on a fuzzy preference relation. Corresponding techniques, which have been investigated quite intensively in the field of fuzzy set theory [4], hence become amenable to the task of classification.

The remainder of the paper is organized as follows. Section 2 details the LPC approach to classification. The idea of classification based on fuzzy preference relations is outlined in section 3. Section 4 elaborates on an important element of this approach, namely the learning of weak preferences between class labels. First empirical results are presented in

2 Learning by Pairwise Comparison

As mentioned before, learning by pairwise comparison (LPC) transforms a multi-class classification problem, i.e., a problem involving $m > 2$ classes (labels) $\mathcal{L} = \{\lambda_1 \dots \lambda_m\}$, into a number of *binary* problems. To this end, a separate model (base learner) $\mathcal{M}_{i,j}$ is trained for each *pair* of labels $(\lambda_i, \lambda_j) \in \mathcal{L}$. $\mathcal{M}_{i,j}$ is intended to separate the objects with label λ_i from those having label λ_j . If $(x, \lambda_a) \in \mathcal{X} \times \mathcal{L}$ is an original training example, then x is considered as a *positive* example for all learners $\mathcal{M}_{a,j}$ and as a *negative* example for the learners $\mathcal{M}_{j,a}$ ($j \neq a$); those models $\mathcal{M}_{i,j}$ with $a \notin \{i, j\}$ simply ignore this example.

At classification time, a query x is submitted to all learners, and each prediction $\mathcal{M}_{i,j}(x)$ is interpreted as a vote for a label. In particular, if $\mathcal{M}_{i,j}$ is a $\{0, 1\}$ -valued classifier, $\mathcal{M}_{i,j}(x) = 1$ is counted as a vote for λ_i , while $\mathcal{M}_{i,j}(x) = 0$ would be considered as a vote for λ_j . Given these outputs, the simplest classification strategy is to predict the class label with the highest number of votes. A straightforward extension of the above voting scheme to the case of $[0, 1]$ -valued (scoring) classifiers yields a *weighted voting procedure*: The score for label λ_i is computed by

$$r_i \stackrel{\text{df}}{=} \sum_{1 \leq j \neq i \leq m} r_{i,j}, \quad (1)$$

where $r_{i,j} = \mathcal{M}_{i,j}(x)$, and again the label with the highest score is predicted.

The votes $r_{i,j}$ in (1) and, hence, the learners $\mathcal{M}_{i,j}$ are usually assumed to be (additively) *reciprocal*, that is,

$$r_{j,i} = 1 - r_{i,j} \quad (2)$$

and correspondingly $\mathcal{M}_{i,j}(x) \equiv 1 - \mathcal{M}_{j,i}(x)$. Practically, this means that only one half of the $m(m-1)$ classifiers $\mathcal{M}_{i,j}$ needs to be trained, for example those for $i < j$. As will be explained in more detail below, this restriction is not very useful in our approach. Therefore, we will train the whole set of classifiers $\mathcal{M}_{i,j}$, $1 \leq i \neq j \leq m$, which means that no particular relation between $r_{i,j}$ and $r_{j,i}$ will be guaranteed.

3 Classifying on the Basis of Fuzzy Preference Relations

Considering the classification problem as a *decision problem*, namely a problem of deciding on a class label for a query input x , an output $r_{i,j} = \mathcal{M}_{i,j}(x)$ can be interpreted as a *preference* of label λ_i in comparison with label λ_j : The higher $r_{i,j}$, the more preferred is λ_i as a classification, i.e., the more likely λ_i appears in comparison with label λ_j . Correspondingly, the matrix

$$\mathcal{R} = \begin{bmatrix} - & r_{12} & \dots & r_{1m} \\ r_{21} & - & \dots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \dots & - \end{bmatrix} \quad (3)$$

obtained by collecting the outputs of the whole classifier ensemble can be interpreted as a *valued preference relation*.

A classification decision can then be made on the basis of the relation (3). To this end, one can resort to corresponding techniques that have been developed in fuzzy preference modeling and decision making [4]. In principle, the simple voting scheme outlined in section 2 can be seen as a special case of such a decision making technique.

What is even more interesting, however, is the application of techniques for decomposing the relation \mathcal{R} into three associated relations with different meaning. Suppose that \mathcal{R} is considered as a *weak preference relation*, which means that $r_{i,j} = \mathcal{R}(\lambda_i, \lambda_j)$ is interpreted as $\lambda_i \succeq \lambda_j$, that is, “label λ_i is at least as likely as label λ_j ”. In this case, \mathcal{R} is reflexive, i.e., the missing diagonal elements in (3) should be 1. In the non-fuzzy case where $r_{i,j} \in \{0, 1\}$, a weak preference relation induces a *strict preference relation* \mathcal{P} , an *indifference relation* \mathcal{I} and an *incomparability relation* \mathcal{J} in a straightforward way. Denoting strict preference by \succ , indifference by \sim , and incomparability by \perp ,

$$\begin{aligned}\lambda_i \succ \lambda_j &\Leftrightarrow (\lambda_i \succeq \lambda_j) \wedge (\lambda_i \not\preceq \lambda_j) \\ \lambda_i \sim \lambda_j &\Leftrightarrow (\lambda_i \succeq \lambda_j) \wedge (\lambda_j \succeq \lambda_i) \\ \lambda_i \perp \lambda_j &\Leftrightarrow (\lambda_i \not\preceq \lambda_j) \wedge (\lambda_j \not\preceq \lambda_i)\end{aligned}$$

Generally, a triple $(\mathcal{P}, \mathcal{I}, \mathcal{J})$ with an asymmetric relation \mathcal{P} , a reflexive and symmetric relation \mathcal{I} and a symmetric relation \mathcal{J} is referred to as a *preference structure*. Formally, $(\mathcal{P}, \mathcal{I}, \mathcal{J})$ must also satisfy the following: $\mathcal{P} \cap \mathcal{I} = \emptyset$, $\mathcal{P} \cap \mathcal{J} = \emptyset$, $\mathcal{I} \cap \mathcal{J} = \emptyset$.

A *fuzzy preference structure* is a triple $(\mathcal{P}, \mathcal{I}, \mathcal{J})$ of fuzzy relations having certain mathematical properties which are generalizations of the above properties required in the non-fuzzy case.¹ Fuzzy preference structures, especially their axiomatic construction, have been studied extensively in literature (e.g. [4, 5, 6, 7, 8, 9]). The same is true for the question of how to decompose a weak (valued) preference relation $\mathcal{R} \in [0, 1]^{m \times m}$ so as to assure certain properties of the triple $(\mathcal{P}, \mathcal{I}, \mathcal{J})$. Without going into technical detail, we only give an example of a commonly employed decomposition scheme (again denoting $r_{i,j} = \mathcal{R}(\lambda_i, \lambda_j)$):

$$\begin{aligned}\mathcal{P}(\lambda_i, \lambda_j) &= \min(r_{i,j}, 1 - r_{j,i}) \\ \mathcal{I}(\lambda_i, \lambda_j) &= \max(r_{i,j} + r_{j,i} - 1, 0) \\ \mathcal{J}(\lambda_i, \lambda_j) &= \max(1 - r_{i,j} - r_{j,i}, 0)\end{aligned}\tag{4}$$

The crucial point is that, in the context of classification, the relations \mathcal{I} and \mathcal{J} do have a very interesting meaning: Indifference corresponds to the *ambiguity* of a classification decision, while incompatibility reflects the corresponding degree of *ignorance*.

To illustrate what we mean, respectively, by ambiguity and ignorance, consider the classification scenario shown in Fig. 2: Given observations from two classes black and white, three new instances marked by a cross need to be classified. Obviously, given the current observations, the upper left instance can quite safely be classified as white. The case of

¹The question of how to generalize these properties is non-trivial. Especially, there are different options which are formally not equivalent.

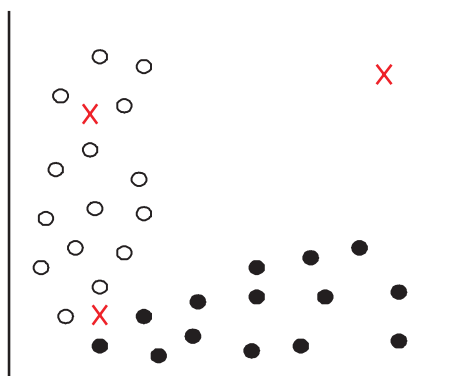


Figure 2: Classification scenario: Observations from two classes (points) and new query instances (crosses).

the lower left instance, however, involves a high level of ambiguity, since both classes, black and white, appear plausible. The third situation is an example of ignorance: The upper right instance is located in a region of the instance space in which no observations have been made so far. Consequently, there is neither evidence in favor of class black nor in favor of class white.

Of course, uncertainty of a classification can also be represented by a standard probabilistic classifier, or even by a more general type of scoring classifier: A classification appears uncertain as soon as at least one other class has a probability (score) which is almost as high as the probability (score) of the predicted class. It should be noted, however, that probability cannot distinguish between ambiguity and ignorance, mainly due to the constraint that probability degrees must always add up to 1. In particular, complete ignorance is usually modeled by the uniform probability distribution, as suggested by the principle of insufficient reason. In the third case of our example, this means giving a probability of $1/2$ to both classes. The adequacy of the uniform distribution as a representation of ignorance has been called into question [10]. In particular, one cannot distinguish between complete ignorance and the situation where one can be quite sure that the class labels are indeed equi-probable, since the uniform distribution has been derived from a large enough number of similar instances (case 2 in our example).

From a knowledge representational point of view, a distinction between ambiguity and ignorance clearly seems worthwhile. For example, telling a patient that your experience does not allow any statement concerning his prospect of survival is very different from telling him that his chance is fifty-fifty! But separating ambiguity from ignorance can also be useful from a decision making (classification) perspective, at least if there is an option to *abstain* from a decision. In fact, in a situation of high ignorance, it might be reasonable to refuse an immediate decision, and instead to gather additional information. For example, consider a physician who is looking for a diagnosis. If the current knowledge about the patient at hand is not sufficient for a reliable diagnosis, he will gather additional information, e.g., by making complementary medical tests or by searching the literature for similar cases. In machine learning terms, the first type of information procurement corresponds to adding features (attributes) to the instance representation, while the second one corresponds to making additional observations.

While abstaining from an immediate decision may be reasonable in a situation of igno-

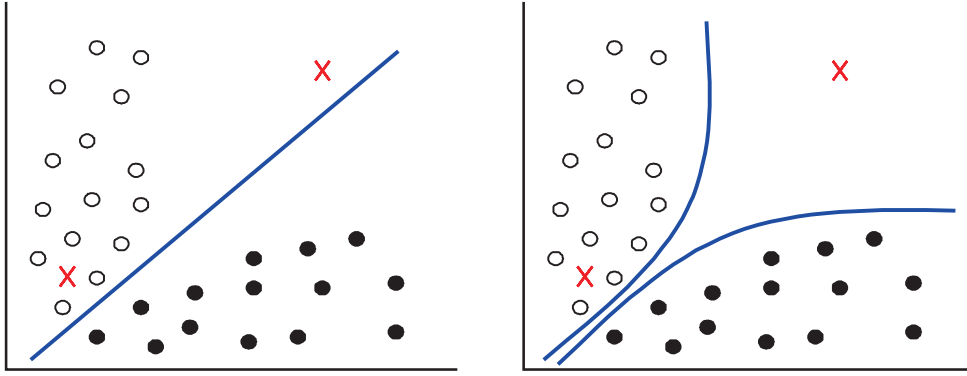


Figure 3: Left: decision boundary induced by a discriminative learner. Right: boundaries that characterize the respective classes.

rance, the same is not necessarily true in cases of ambiguity. In the second case of our example in Fig. 2, for instance, additional observations will hardly be useful, i.e., corresponding examples will hardly resolve the ambiguity. As opposed to this, additional examples could be very helpful in the third case, where they may indeed reduce the level of ambiguity.

The general idea of our method is to learn the weak preference relation (3), using an LPC approach, and to decompose this relation into a preference structure $(\mathcal{P}, \mathcal{I}, \mathcal{J})$, that is, into three corresponding relations

$$\begin{bmatrix} 0 & p_{1,2} & \cdots & p_{1,m} \\ p_{2,1} & 0 & \cdots & p_{2,m} \\ \vdots & & & \vdots \\ p_{m,1} & p_{m,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} 1 & \iota_{1,2} & \cdots & \iota_{1,m} \\ \iota_{2,1} & 1 & \cdots & \iota_{2,m} \\ \vdots & & & \vdots \\ \iota_{m,1} & \iota_{m,2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} 0 & \mathcal{J}_{1,2} & \cdots & \mathcal{J}_{1,m} \\ \mathcal{J}_{2,1} & 0 & \cdots & \mathcal{J}_{2,m} \\ \vdots & & & \vdots \\ \mathcal{J}_{m,1} & \mathcal{J}_{m,2} & \cdots & 0 \end{bmatrix}$$

These relations can then be taken as a point of departure for sophisticated decision strategies. This approach becomes especially interesting in extended classification scenarios, for example, if several class labels instead of single one may be predicted in cases of ambiguity, or an immediate decision may be deferred in cases of ignorance. The discussion of concrete decision strategies is beyond the scope of this paper.

4 Learning Weak Preference Relations

Since the relations \mathcal{P} , \mathcal{I} , and \mathcal{J} are derived from the weak preference relation \mathcal{R} , the first step of our method consists of learning this relation. A first important observation is that standard *discriminative* learning methods are not suitable for this purpose. In fact, such methods produce decision boundaries which do not reflect the degree of ignorance in an adequate way.

To illustrate this problem, Fig. 3 (left) shows a separating hyperplane as it would be derived, e.g., by a support vector machine. Given a new instance to be classified, the score of a class is typically defined as a function of the distance of the instance from the

hyperplane. Defining the weak preferences $\mathcal{R}(\text{black}, \text{white})$, $\mathcal{R}(\text{white}, \text{black})$ in terms of this score, ambiguity and ignorance will, according to (4), only depend on this distance. In particular, the same degrees of ambiguity and ignorance will be predicted for the two instances marked by a cross, which is of course not reasonable.

As can be seen, finding a decision boundary that optimally discriminates between the two classes is not appropriate in the context of our application. Instead, it is more reasonable to characterize both classes separately, as shown in the right picture in Fig. 3. The upper decision boundary characterizes the class `white` while the same is accomplished for class `black` by the lower boundary. Correspondingly, the weak preference $\mathcal{R}(\text{white}, \text{black})$ is a function of the upper boundary, while $\mathcal{R}(\text{black}, \text{white})$ depends on the lower boundary. For example, for the instance located at the lower left cross, $\mathcal{R}(\text{white}, \text{black})$ will be high and $\mathcal{R}(\text{black}, \text{white})$ will be small. In contrast, for the second instance located at the upper right cross, both $\mathcal{R}(\text{white}, \text{black})$ and $\mathcal{R}(\text{black}, \text{white})$ will be small. As desired, the degree of ignorance will therefore be quite high for this instance.

The above comments explain why we do not make the assumption (2) of reciprocal votes and instead train models $\mathcal{M}_{i,j}$ and $\mathcal{M}_{j,i}$ separately. In both cases, the two classes λ_i and λ_j are treated in an asymmetric way: $\mathcal{M}_{i,j}$ is intended to characterize class λ_i while $\mathcal{M}_{j,i}$ should characterize λ_j . Note that inducing a characterizing model is typically accomplished by methods for concept learning. For example, rule induction methods like covering algorithms typically learn this type of model [11].

5 Experimental Results

In this section, we present some first empirical results providing evidence for the usefulness of our approach. To this end, we have considered an extended classification scenario in which one is not forced to decide in favor of a single class label. Instead, a subset of potential candidate labels may be predicted. However, such ambiguous predictions are punished in terms of a suitable loss function:

$$\text{loss}(L, \lambda) = \begin{cases} \frac{|L|-1}{m-1} & \lambda \in L \\ 1 & \lambda \notin L \end{cases}$$

where $L \subseteq \mathcal{L}$ is a set-valued prediction and λ denotes the true class label. This setting obviously focuses on the aspect of ambiguity in classification, that is, the relation \mathcal{I} .

For this study, the following data sets have been taken from the Statlog and UCI repositories: `segment` (2310 instances, 7 classes), `letter` (15000 instances, 26 classes), `satimage` (4435 instances, 6 classes), `shuttle` (43500 instances, 5 classes). As training and testing data, 1000 examples have been chosen at random, respectively, from the original data (except for the `segment` data set, which has been split into two sets of equal size).

As binary models $\mathcal{M}_{i,j}$ we used cost-sensitive support vector machines (with linear kernels) as offered by the `libsvm`-library [12]. As an effect of decreasing the weight (misclassification cost) of a class, the corresponding decision boundary moves closer to the representative examples of this class. This way, it becomes possible to train non-reciprocal models $\mathcal{M}_{i,j}$ and $\mathcal{M}_{j,i}$ as outlined in the previous section.

To make use of the indifference relation \mathcal{I} , we have employed the following decision strategy: First, the prediction set L is initialized by a single best label λ^* which is chosen

c	t	loss \pm std	$ L $	T	loss \pm std	$ L $
1	-	0,068 \pm 0,253	1,000	0	0,068 \pm 0,253	1,000
2	0,2	0,050 \pm 0,183	1,113	0	0,068 \pm 0,253	1,000
4	0,3	0,048 \pm 0,157	1,166	0	0,068 \pm 0,253	1,000
8	0,4	0,056 \pm 0,178	1,171	0	0,068 \pm 0,253	1,000
16	0,4	0,054 \pm 0,149	1,239	0	0,068 \pm 0,253	1,000
1	-	0,299 \pm 0,458	1,000	2	0,189 \pm 0,331	2,398
2	0,1	0,189 \pm 0,270	4,239	3	0,185 \pm 0,274	3,365
4	0,2	0,177 \pm 0,288	3,612	2	0,183 \pm 0,324	2,385
8	0,2	0,193 \pm 0,259	4,925	2	0,182 \pm 0,322	2,383
16	0,2	0,194 \pm 0,245	5,211	3	0,189 \pm 0,279	3,363
1	-	0,146 \pm 0,353	1,000	0	0,144 \pm 0,351	1,000
2	0,2	0,118 \pm 0,283	1,178	0	0,145 \pm 0,352	1,000
4	0,3	0,110 \pm 0,251	1,261	0	0,146 \pm 0,353	1,000
8	0,3	0,117 \pm 0,216	1,425	0	0,143 \pm 0,350	1,000
16	0,3	0,132 \pm 0,226	1,493	0	0,145 \pm 0,352	1,000
1	-	0,067 \pm 0,250	1,000	0	0,067 \pm 0,250	1,000
2	0,3	0,033 \pm 0,127	1,083	0	0,067 \pm 0,250	1,000
4	0,3	0,044 \pm 0,148	1,109	0	0,067 \pm 0,250	1,000
8	0,3	0,040 \pm 0,103	1,146	0	0,067 \pm 0,250	1,000
16	0,3	0,049 \pm 0,114	1,181	0	0,067 \pm 0,250	1,000

Table 1: Results for the data sets segment, letter , satimage, shuttle (from top to bottom): class weight used by the SVM learner (c), indifference threshold, average and standard deviation of the loss function, average size of prediction set L (left: our approach, right: baseline).

from the strict preference matrix \mathcal{P} using a simple weighted voting scheme. Then, L is extended by including all labels λ such that $\lambda \sim \lambda^*$. Since \mathcal{I} is a fuzzy relation, the prediction thus defined is actually a fuzzy set. To simplify the evaluation, we have turned this set into a crisp one by thresholding \mathcal{I} . More specifically, we defined $(\lambda \sim \lambda^*) \Leftrightarrow (\mathcal{I}(\lambda, \lambda^*) \geq t)$, where $t \in \{0.1, 0.2 \dots 1\}$ is a suitable threshold. The latter has been optimized on the training data using 10-fold cross validation.

For comparison, we used simple weighted voting on the original relation \mathcal{R} as a baseline. To allow for an analogous form of set-valued prediction, L was defined by all labels the total score (sum of votes) of which differs from the score of the best label by at most T ; again, T is a suitable threshold which is optimized on the training data (candidate values for T are $\{0, 1 \dots 5\}$).

The results of the experiments are documented in Table 1. As can be seen, our approach performs better than the baseline method throughout, even though the improvement in the average loss is statistically not significant. Nevertheless, we take the results as first evidence in favor of the hypothesis that our idea of basing classification decisions on (fuzzy) preference structures may improve performance in extended classification scenarios.

6 Conclusions

In this paper, we have introduced a new approach to classification which refers to the concept of fuzzy preference structures. This approach is intimately related with learning by pairwise comparison (LPC), in machine learning a well-known technique for reducing multi-class to binary problems: LPC is used in order to learn a weak preference relation among the potential class labels, which is then decomposed into a strict preference, indifference, and incomparability relation.

We have argued that, in the classification context, indifference can be interpreted as the ambiguity of a prediction while indifference represents the level of ignorance. These concepts can be extremely useful, especially in extended classification scenarios which go beyond the prediction of a single label. Besides, the aspect of ignorance, which has not received much attention in machine learning so far, seems to be especially interesting in connection with so-called *reliable* classification [13]. Investigating the potential of our approach in this context is an important aspect of future work.

First empirical studies have focused on using the indifference relation in order to improve classification performance in a set-valued classification setting where several labels can be predicted in cases of ambiguity. The results are quite promising, even though they should clearly be considered as preliminary. Conducting more extensive experimental studies is another major point on the agenda. Besides, we plan to consider other classification scenarios, especially those in which the aspect of ignorance and, hence, the idea of abstaining from or deferring a decision play an important role. In this connection, it will also be necessary to develop new decision policies which produce reasonable classification decisions on the basis of a fuzzy preference structure.

Acknowledgements. This research was supported by the German Research Foundation (DFG).

References

- [1] Schölkopf, B.; Smola, A.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press. 2001.
- [2] Fürnkranz, J.: Round robin rule learning. In: *ICML-2001, Proc. 18th International Conference on Machine Learning*, S. 146–153. Williamstown, MA. 2001.
- [3] Fürnkranz, J.: Round robin classification. *Journal of Machine Learning Research* 2 (2002), S. 721–747.
- [4] Fodor, J.; Roubens, M.: *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer. 1994.
- [5] Baets, B. D.; de Walle, B. V.; Kerre, E.: Fuzzy preference structures and their characterization. *The Journal Fuzzy Mathematics* 3 (1995), S. 373–381.
- [6] Fodor, J.; Roubens, M.: Valued preference structures. *European Journal of Operational Research* 79 (1994), S. 277–286.

- [7] Orlovski, S.: Decision-making with a fuzzy preference relation. *Fuzzy Sets and Systems* 1 (1978), S. 155–167.
- [8] Ovchinnikov, S.; Ozernoy, V.: Using fuzzy binary relations for identifying noninferior decision alternatives. *Fuzzy Sets and Systems* 25 (1988), S. 21–32.
- [9] Perny, P.; Roy, B.: The use of fuzzy outranking relations in preference modelling. *Fuzzy Sets and Systems* 49 (1992), S. 33–53.
- [10] Dubois, D.; Prade, H.; Smets, P.: Representing partial ignorance 26 (1996) 3, S. 361–377.
- [11] Fürnkranz, J.: Separate-and-Conquer Rule Learning. *Artificial Intelligence Review* 13 (1999) 1, S. 3–54.
- [12] Chang, C.-C.; Lin, C.-J.: *LIBSVM: a library for support vector machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 2001.
- [13] Kukar, M.; Kononenko, I.: Reliable Classifications with Machine Learning. In: *Proc. European Conference on Machine Learning, ECML*, S. 219–231. 2002.

Granulierungseffekte in unscharf definierten Medien bei Strukturbildungsprozessen mit Bottom-Up-Verfahren

Ralf Rettberg, Hanns Sommer

Universität Kassel Fb.15
34109 Kassel Mönchebergstr. 7
Tel.: (0561) 804 3261
Fax: (0561) 804 7768
E-Mail: sommer@rts-maschinenbau.uni-kassel.de

Zusammenfassung

Wir untersuchen den Strukturbildungsprozess bei Kettenmolekülen in einer Lösung mittels Monte Carlo Methoden. Durch die Verwendung von Aggregationsoperatoren wird es ermöglicht, Granulierungseffekte zu berücksichtigen .

1 Einleitung

In der Nanotechnologie stoßen die gängigen Top-Down Verfahren (z.B. Ätzmethoden) zur Herstellung von Nanostrukturen, mit dem Kleinerwerden dieser Gebilde, immer mehr an technologische Grenzen. Als Alternative bieten sich Bottom-Up Verfahren an, bei denen speziell gewünschte Mikrostrukturen aus makroskopischen Umgebungen emergieren. In Lösungen von Kettenmolekülen können sich bei entsprechenden Konzentrationen und Temperaturen strukturierte Mikrogebilde (sogenannte 'scaffolds') herausbilden, die dann als Grundbausteine für Nanomaschinen Verwendung finden. Das Verstehen und die Regelung solcher Bildungsprozesse ist für die Nanotechnologie von großer Bedeutung.

Im zweiten Abschnitt werden wir auf die Prinzipien zur Modellbildung solcher Prozesse eingehen. Wir beschreiben eine Lösung mit Kettenmolekülen durch einen Metropolis Algorithmus. D.h. wir nehmen an, die Kettenmoleküle seien in einem Raumgebiet zunächst zufallsverteilt und führen dann zufällige Bewegungen aus. Die nach einer Zufallsbewegung entstehende neue Konfiguration wird bewertet und behalten, falls sie besser abschneidet als die Ausgangskonfiguration. Andernfalls wird diese behalten.

In der richtigen, den physikalischen Gesetzen entsprechenden Bewertung, liegt das eigentliche Problem einer solchen Modellbildung. Es zeigt sich, dass Methoden der klassischen Kontinuumsmechanik hierzu nicht ausreichen. Deren rein lokale Verfahren sind nicht in der Lage, den Einfluss von Granulierungseffekten zu beschreiben. Granuli sind Sinneinheiten, die Zadeh folgendermaßen zu präzisieren versucht [1]:

” In g-precisation, with g standing for granular, [an approximatively known entity] *a is treated as a granule, with a granule being a clumb of values [or structures] drawn together by indistinguishability, equivalence or proximity.”

Eines der Anliegen Zadehs, bei der Einführung der Fuzzy Logik war es, zum Verständnis solcher Sinneinheiten beizutragen. Zadeh bedauert ausdrücklich, dass dieses Ziel bisher nicht in ausreichender Weise erreicht wurde.

Im dritten Abschnitt werden Aggregationsoperatoren nach Yager [2] eingeführt und es wird gezeigt, wie mit diesen globale Strukturbildungen im Evolutionsprozess einer Lösung

von Kettenmolekülen berücksichtigt werden können. Insbesondere die Kontextabhängigkeit der Zeitentwicklung der Lösung, zur Berücksichtigung quantenmechanischer Phänomene kann dadurch im Modell erfasst werden.

Im vierten Abschnitt werden Simulationsergebnisse vorgestellt, in denen sich der Einfluss der verschiedenen Aggregationsoperatoren zeigt. Wir weisen nach, dass das vorgelegte Modellbildungsverfahren die Möglichkeiten der klassischen Mechanik übersteigt und somit zur Modellierung von Phänomenen in Frage kommt, die im Rahmen der klassischen Physik nicht verstehbar sind. Die vorgelegten Ergebnisse sind ein erster Schritt zum Verständnis der Strukturbildung in nichtklassischen Prozessen. Unser Fernziel ist es, diese Strukturbildungsprozesse, durch geeignete Eingriffe mittels der globalen Parameter, zu regeln.

2 Emergenz von Strukturen aus Lösungen

2.1 Problemstellung

Ein in der Nanotechnologie verwendetes Verfahren zur Erzeugung von Strukturen besteht darin, Kettenmoleküle (z.B. DNA-Ketten) in eine Lösung zu geben. Da die Ketten bestimmte Präferenzen haben, sich aneinander anzulagern, entstehen nach einiger Zeit in der Lösung charakteristische Strukturen, die zum Bau von Nano-Geräten verwendet werden können [3]. Aufgabe des Ingenieurs ist es, die Prozesse so steuerbar zu machen, dass genau die Strukturen entstehen, die später verwendet werden können. Diese Aufgabe stellt die Ingenieure jedoch vor erhebliche Probleme, da die Strukturbildungsprozesse nicht mit den Differentialgleichungen der klassischen Mechanik beschreibbar sind.

Die wesentlichen Phänomene, die diese Strukturbildungsprozesse so schwer beschreibbar machen, sind die folgenden:

(A) Führung des Prozesses durch stabile Zwischenkonfigurationen:

Die Chemiker haben seit langem erkannt, dass chemische Reaktionen meist nicht direkt, sondern über Zwischenkonfigurationen verlaufen [4]. Durch diese Zwischenkonfigurationen, die auch mittels Katalysatoren ermöglicht werden können, werden die Strukturen so 'vorgeordnet', dass die Wahrscheinlichkeit für die Entstehung des Endprodukts, gegenüber einem rein zufälligen Zusammentreffen der Teile, erheblich erhöht wird. Zum Verstehen eines Reaktionsverlaufs ist es für den Chemiker notwendig, diese Zwischenprodukte zu kennen. Für uns jedoch, die wir eine neue Reaktion entwerfen wollen, sind diese Strukturen nicht vorhersehbar und damit ist der Strukturbildungsprozess nicht exakt prognostizierbar. Wir sind bei der Modellbildung auf Monte Carlo Methoden angewiesen.

(B) Quantenmechanische Einflüsse:

Da die Strukturbildung auf atomarer Ebene abläuft, müssen die Gesetze der Quantenmechanik und deren charakteristische Phänomene berücksichtigt werden. Wesentliche nicht klassische Phänomene sind, z.B. der Tunneleffekt, auf Grund dessen Teilchen Potentialbarrieren durchtunneln können, und die Notwendigkeit, Fermi- bzw. Bose-Statistiken anstelle der klassischen Boltzmann-Statistik zu verwenden.

(C) Klassische Randbedingungen:

Der Strukturbildungsprozess kann aber auch nicht rein quantenmechanisch beschrieben werden. Durch Dekohärenzeffekte (vgl. Doppelspaltexperiment mit Fullerenen), bilden

sich in einem System aus vielen Teilen notwendigerweise Strukturen heraus, die klassisch beschrieben werden müssen. Chemiker gehen dann häufig so vor, dass sie die klassischen Strukturen als Randbedingungen auffassen zu den Differentialgleichungen, die die quantenmechanischen Effekte beschreiben [5]. In gemischten quanten-klassischen Systemen werden die schweren Moleküle als klassisches System beschrieben, das die Randbedingungen liefert zum quantenmechanischen System der leichten Elektronen, Protonen und Photonen.

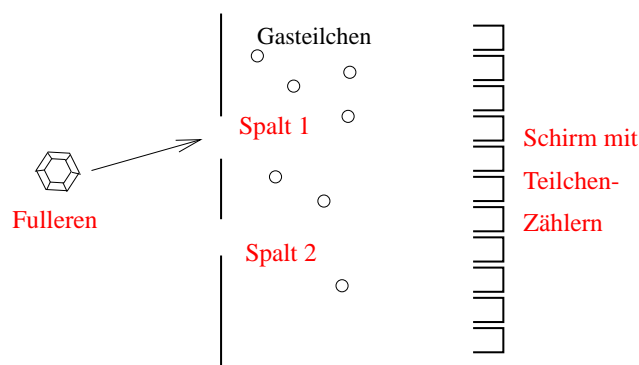
Das Problem dieses Ansatzes besteht jedoch in der apriori vorausgesetzten Trennung zwischen leichten und schweren Teilen. Üblicherweise bestehen ja die schweren Teile, die Moleküle, aus den leichten Teilen. Wenn wir die Trennung dieser Strukturen voraussetzen, so können wir prinzipiell Bildungsprozesse der Entstehung schwerer Strukturen, nicht untersuchen. Wir würden damit unser Ziel, Strukturbildungsprozesse zu untersuchen, im Ansatz verfehlen. Das Problem der Trennung von quantenmechanischer und klassischer Beschreibung, das eines der Grundprobleme der Nanotechnologie ist, wird von Sen und Sengupta verdeutlicht [6]:

”Es sind zwei verschiedene Realitäten, die bezüglich der Skalierung der Physik zu existieren scheinen und deren Komplexität im Grenzbereich zwischen Quantenmechanik und klassischer Mechanik stark zum Ausdruck kommt. Dieser Grenzbereich, in dem Elemente beider Mechaniken für ein vollständiges Verständnis notwendig erscheinen, ist ziemlich undurchsichtig.

Weder ein vollständiger mathematischer Formalismus (zum Verständnis dieses Grenzbereichs), noch eine konzeptuelle Struktur einer Theorie kann von einer der beiden Mechaniken abgeleitet werden.”

Um nanotechnologische Strukturbildungsprozesse prinzipiell regelbar zu machen, müssen wir daher Methoden zur Modellbildung verwenden, die eine Kombination quantenmechanischer Grundprinzipien und gleichzeitig den Übergang zu klassischen Strukturen erlauben.

Doppelspaltexperiment mit fluorierten Fullerenen [7]:



Flourierte Fullerene sind Moleküle aus 60 C- und 48 F-Atomen. Flourierte Fullerene sind Teilchen, die wie Photonen, beim Doppelspaltexperiment am Schirm Interferenzerscheinungen zeigen. Für sie gilt also ebenfalls das Superpositionsprinzip der Quantenmechanik. Durch ein leichtes Gas im Raum zwischen den Spalten und dem Schirm wird jedoch die Interferenz zum Verschwinden gebracht. Nicht wegen der Stöße, die die Gasteilchen auf die Fullerene ausüben, dafür sind diese zu leicht, sondern durch die Herstellung eines 'Gesamtsystems' aus Fullerenen und Umgebung, das die Gasteilchen bewirken.

2.2 Lösungsansatz

Das Modell zur Emergenz von Strukturen aus Lösungen von Kettenmolekülen soll zwei Prinzipien realisieren:

Kontextabhängigkeit und Übergang zur klassischen Beschreibung mittels Dekohärenz.

Das erste Prinzip, die Kontextabhängigkeit, ist das Grundprinzip, das quantenmechanische Vorgänge von klassischen Prozessen unterscheidet. Die Kontextabhängigkeit quantenmechanischer Prozesse wurde von Kochen und Specker nachgewiesen, und es kann gezeigt werden, dass dieser Effekt die anderen typischen quantenmechanischen Phänomene zur Folge hat [8].

Kontextabhängigkeit bedeutet, dass der Prozessverlauf in einer Teilregion nicht nur durch die Verhältnisse in dieser selbst, und dem über deren Ränder hereinkommenden Einfluss der Außenwelt bestimmt ist, sondern dass Strukturen (Granuli) im System in ihrer Gesamterscheinung Einfluss nehmen auf alle Regionen, in denen Teile der Struktur präsent sind. Ein kontextabhängiges System konkretisiert sich im Bezug auf alle seine Teile. Diese Konkretisierung muss aber nicht eindeutig sein, so dass quantenmechanische Systeme in Superpositionen aus mehreren Konkretisierungen existieren können. Die Umkehrfunktion zu dieser mehrdeutigen Existenz (der Superposition der Quantenzustände), ist die Dekohärenz. H.D. Zeh schreibt [9]:

” Die Idee der Quantisierung kann allgemein verstanden werden als Umkehrung der Dekohärenz: die Wiederherstellung der Superpositionen welche durch die Wechselwirkung mit der Umgebung unterdrückt werden und die damit die klassische Beschreibung möglich machen.”

Wegen dem Auftreten der Dekohärenz kann ein System aus sehr vielen miteinander wechselwirkenden Teilen nicht im Zustand der Superposition aus mehreren Konkretisierungen verharren. In solchen wechselwirkenden Systemen sehr vieler Teile überwiegt immer die Wahrscheinlichkeit für eine Konkretisierung, so dass es zum klassischen System wird. In großen Systemen sehr vieler Teile wird eine gleichgewichtige Wahrscheinlichkeitsverteilung für verschiedene Konkretisierungen extrem unwahrscheinlich. Dies ist eine Folge des Gesetzes der großen Zahl, die aus Percolationsexperimenten bekannt ist [10].

Bilden sich also Granuli in der Lösung der Kettenmoleküle, so muss im Modell sichergestellt sein, dass diese Granuli sich gegeneinander nach klassischen Gesetzen bewegen, wohingegen im Einflussbereich eines Granuli liegende Teilgebiete sich nach den Gesetzen für kontextabhängige Systeme entwickeln. Z.B. wird der Tunneleffekt in kontextabhängigen Systemen dadurch bewirkt, dass globale Beziehungen ein lokales Teilgebiet zu einer Entwicklung zwingen, die den dort lokal herrschenden Energiebedingungen zuwider läuft.

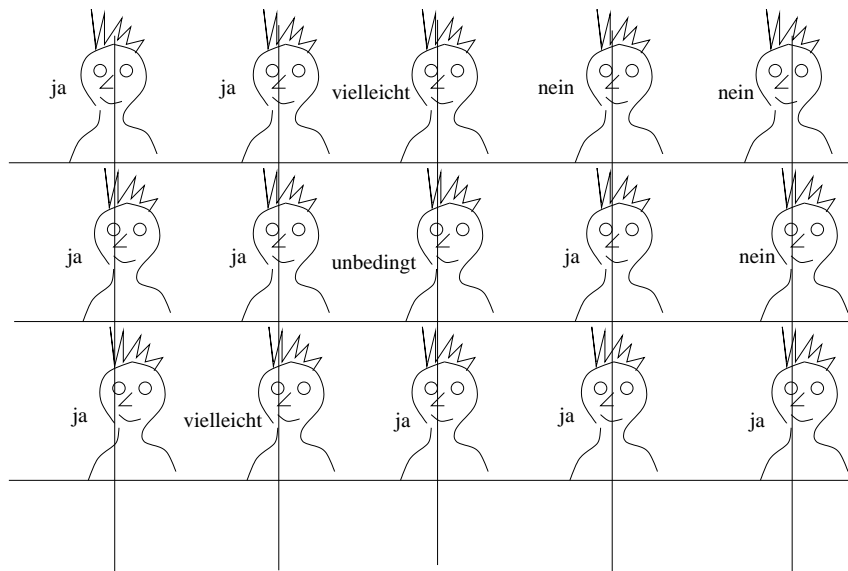
Das wesentliche Hilfsmittel, das wir zur Erweiterung der Modellbildungsmöglichkeiten einsetzen, sind die Aggregationsoperatoren zur Bewertung der Lage der Kettenmoleküle. Sie erlauben es, den Begriff der Richtigkeit zu erweitern. Nach klassischer Auffassung überträgt sich die Richtigkeit von den Teilen auf das Ganze, d.h. nur was in allen seinen Teilen richtig ist, kann auch als Ganzes richtig sein. Der Richtigkeitsbegriff, der sich mit Aggregationsoperatoren ergibt, kann jedoch die globale Gesamtstruktur stärker berücksichtigen, als die Details und somit kann eine lokale Entwicklung auch in die falsche Richtung laufen (entgegen dem Energiegradienten), wenn nur der ”Gesamttrend” stimmt.

3 Aggregation von Gesamtbewertungen

Da wir die Gesetze unter denen Strukturen in Lösungen emergieren nicht kennen, so müssen wir das Aggregationsproblem, das sich am Ende des zweiten Abschnitts stellte, in größtmöglicher Allgemeinheit formulieren:

Aggregationsproblem:

Bei einem System aus vielen Teilen, liege für jeden der Teile eine Bewertung vor. Wie kann aus diesen Teilbewertungen eine Gesamtbewertung erzeugt werden?



Gitterpunkte, in denen die Agenten die richtigkeit der lokalen Lösungsbedingungen prüfen

Die allgemeinste Definition von Aggregationsoperatoren (R.R. Yager):

Bezeichnungen: $U^{[0,1]}$:= $\{ \langle u_1, u_2, \dots, u_n \rangle \text{ ungeordnete } n\text{-Tupel} \text{ mit Wiederholung von Elementen } u_i \in [0, 1] \}$

$\langle u_1, u_2, \dots, u_n \rangle \oplus \langle v_1, v_2, \dots, v_m \rangle := \langle u_1, \dots, u_n, v_1, \dots, v_m \rangle$

$\langle u_1, u_2, \dots, u_n \rangle \leq \langle v_1, v_2, \dots, v_m \rangle \Leftrightarrow n = m$ und es gibt eine Umordnung $\Pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ so dass für $i = 1, \dots, n$ gilt: $u_i \leq v_{\Pi(i)}$.

Definition: $A : U^{[0,1]} \rightarrow [0, 1]$ heißt **Aggregationsoperator**: \Leftrightarrow

Für alle $\langle u_1, u_2, \dots, u_n \rangle, \langle v_1, v_2, \dots, v_m \rangle \in U^{[0,1]}$ gilt:

- $\langle u_1, u_2, \dots, u_n \rangle \leq \langle v_1, v_2, \dots, v_m \rangle \Rightarrow A(\langle u_1, u_2, \dots, u_n \rangle) \leq A(\langle v_1, v_2, \dots, v_m \rangle)$
- Es gibt ein $u_0 \in [0, 1]$ (die Stimmenthaltung) mit:
 $A(\langle u_1, u_2, \dots, u_n \rangle \oplus \langle u_0 \rangle) = A(\langle u_1, u_2, \dots, u_n \rangle)$

Definition: Ein Aggregationsoperator A heißt

(a) **und-Operator**, (b) **oder-Operator**, bzw. (c) **assoziativ** falls

zusätzlich für alle $\langle u_1, \dots, u_n \rangle, \langle v_1, \dots, v_m \rangle, \langle w_1, \dots, w_m \rangle \in U^{[0,1]}$ gilt:

- (a) $A(\langle u_1, \dots, u_n \rangle) \geq A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle)$ bzw.

- (b) $A(\langle u_1, \dots, u_n \rangle) \leq A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle)$
- (c) $A(\langle A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle) \rangle \oplus \langle w_1, \dots, w_m \rangle) =$
 $A(\langle u_1, \dots, u_n \rangle \oplus \langle A(\langle v_1, \dots, v_m \rangle \oplus \langle w_1, \dots, w_m \rangle) \rangle) =$
 $A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle \oplus \langle w_1, \dots, w_m \rangle)$

Aggregationsoperatoren sind auf Grunde der folgenden Sätze mittels der Fuzzy Logik darstellbar (Die Beweise finden sich in [11]):

SATZ (a): Ein **assoziativer und-Aggregationsoperator** kann nach entsprechender Umskalierung des Bewertungsintervalls $[0, 1]$ mit einer bijektiven, bistetigen, monoton steigenden Skalierungsfunktion $\Psi_\wedge : [0, 1] \rightarrow [0, 1]$ dargestellt werden mittels dem **Fuzzy-und-Operator** \wedge der für $a, b \in [0, 1]$ definiert ist durch: $a \wedge b := a \cdot b$ (bzw. $a \wedge b := \min(a, b)$):

$$A(\langle u_1, u_2, \dots, u_n \rangle) := u_1 \wedge u_2 \wedge \dots \wedge u_n$$

(a): Ein **assoziativer oder-Aggregationsoperator** kann nach entsprechender Umskalierung des Bewertungsintervalls $[0, 1]$ mit einer bijektiven, bistetigen, monoton steigenden Skalierungsfunktion $\Psi_\vee : [0, 1] \rightarrow [0, 1]$ dargestellt werden mittels dem **Fuzzy-oder-Operator** \vee der für $a, b \in [0, 1]$ definiert ist durch: $a \vee b := 1 - (1 - a) \cdot (1 - b)$ (bzw. $a \vee b := \max(a, b)$):

$$A(\langle u_1, u_2, \dots, u_n \rangle) := u_1 \vee u_2 \vee \dots \vee u_n$$

Definition: Ein Aggregationsoperator heißt **verstärkend**, falls eine Bestätigung mit $u > u_0$ (bzw. eine Verneinung mit $v < u_0$) durch mehrere Agenten stärker wirkt als nur diejenige eines einzelnen Agenten:

$$A(\langle u, u \rangle) > A(\langle u \rangle) > A(\langle u_0 \rangle) \quad (\text{bzw. } A(\langle v, v \rangle) < A(\langle v \rangle))$$

Der wesentliche Satz zur Darstellung von Aggregationsoperatoren geht auf eine Idee von Kiendl [12] zurück, der erkannte dass Bejahung und Veto verschiedene Konzepte sind. (Kiendl bewies, dass Fuzzy-Regler mit Bejahung und Verneinung allgemeiner sind als Mamdani-Fuzzy-Regler.)

SATZ: Ein **assoziativer, verstärkender Aggregationsoperator** kann nach entsprechender Umskalierung des Bewertungsintervalls $[0, 1]$ mit bijektiven, bistetigen, monoton steigenden Skalierungsfunktionen $\Psi_\wedge : [0, u_0] \rightarrow [0, 1]$ und $\Psi_\vee : [u_0, 1] \rightarrow [0, 1]$ dargestellt werden mittels den **Fuzzy-Operatoren** \wedge und \vee , die für $a, b \in [0, 1]$ definiert sind durch: $a \wedge b := a \cdot b$, bzw. $a \vee b := 1 - (1 - a) \cdot (1 - b)$.

Dabei beschreibt \vee einen bejahenden - und \wedge einen verneinenden Aggregationsoperator, deren Bewertung nach den folgenden Prinzipien zu einer Bewertung zusammengefasst werden kann:

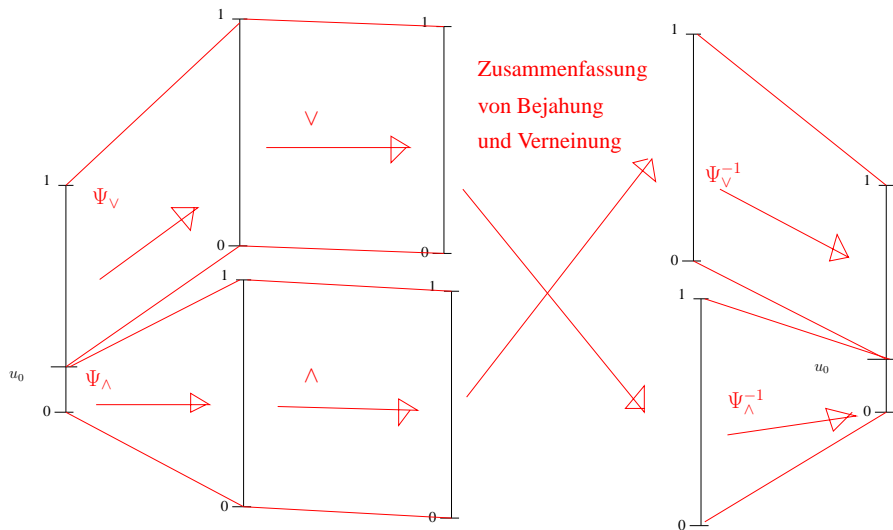
- **Starkes Veto:** Wird eine Verneinung ausgesprochen (eine Bewertung $< u_0$), so werden alle Bejahungen vernachlässigt.
- **Starkes Ja:** Wird eine Bejahung ausgesprochen (eine Bewertung $> u_0$), so werden alle Verneinungen vernachlässigt.

- **Ausgewogene Entscheidung:** Seien $u_i, i = 1, \dots, n$ Bejahungen oder Verneinungen mit:

$$m_i := \begin{cases} \ln(u_i) & \text{für eine Verneinung} \\ -\ln(1 - u_i) & \text{für eine Bejahung} \end{cases}$$

$$m_{\text{aggregiert}} := \sum_{i=1}^n m_i$$

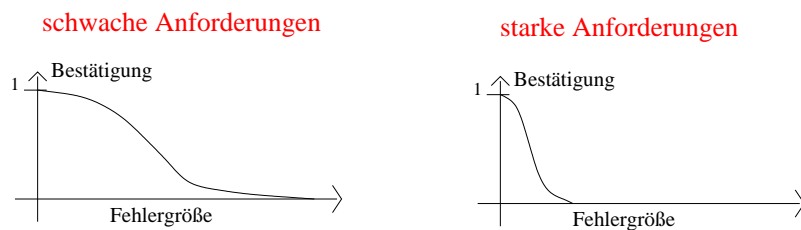
$$u_{\text{aggregiert}} := \begin{cases} \exp(m_{\text{aggregiert}}) & \text{für } m_{\text{aggregiert}} < 0 \\ 1 - \exp(-m_{\text{aggregiert}}) & \text{für } m_{\text{aggregiert}} \geq 0 \end{cases}$$



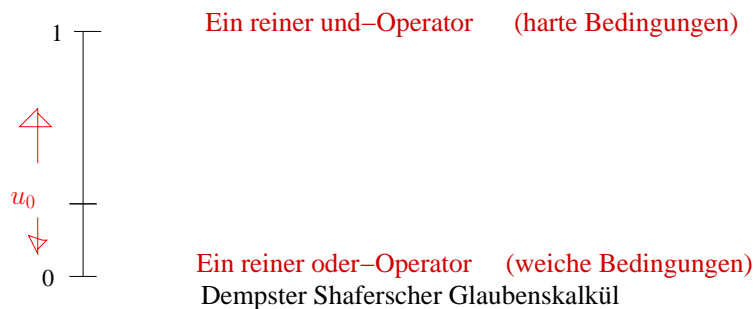
Als Folgerung aus dem letzten Satz erhalten wir eine Zusammenstellung aller **Freiheitsgrade zur Definition von Aggregationsoperatoren**.

Schwache Anforderungen und weiche Bedingungen führen zur Verstärkung der Gesamtbewertung durch die Aggregation der Teilbewertungen, wohingegen starke Anforderungen und harte Bedingungen die aggregierte Gesamtbewertung klein halten.

(I) Skalierungen Ψ_v und $\Psi_^$:



(II) Typenbestimmung des Aggregationsoperators durch Wahl von u_0 :



(III) Zusatzbestätigung beim Auftreten von Granuli:

Gebiete (und deren Umgebung), die durch die Molekülketten verbunden sind, werden mit stark verstärkenden Aggregationsoperatoren bewertet (z.B. oder-Aggregationsoperatoren), die Bewertung der Zwischengebiete in denen keine Molekülketten liegen (und diejenige nichtverbundener Komplexe) könnte dagegen mit einem wenig verstärkenden Aggregationsoperator, (z.B. dem Summenoperator '+') zusammengefasst werden .

4 Erste Tests mit dem neuen Modellbildungsverfahren

4.1 Modellbeschreibung

Die Molekülketten werden dargestellt durch l direkt benachbarte Punkte (i, j) in einem $n \times n$ -Gitter. Um Randeffekte zu vermeiden wird das Gitter zum Torus gefaltet..

Jeder Punkt (i, j) einer Molekülkette hat eine Ladung $e(i, j) \in \{-1, 0, 1\}$.

Da sich verschiedene Ladungen anziehen und gleiche Ladungen abstoßen, sind Strukturen in denen verschiedene Ladungen benachbart sind, stabiler als andere. Die lokale Stabilität in einem Gitterpunkt (i, j) wird beschrieben durch den Ausdruck:

$$w(i, j) := 0.8 \cdot \frac{A_{n-} - A_{n+}}{8} + 0.2 \cdot \frac{A_{un-} - A_{un+}}{12}$$

Mit $A_{n-} :=$ Anzahl der nächsten Nachbarn von (i, j) mit verschiedenem Vorzeichen,

$A_{n+} :=$ Anzahl der nächsten Nachbarn von (i, j) mit gleichem Vorzeichen,

$A_{un-} :=$ Anzahl der übernächsten Nachbarn von (i, j) mit verschiedenem Vorzeichen,

$A_{un+} :=$ Anzahl der übernächsten Nachbarn von (i, j) mit gleichem Vorzeichen.

Die lokalen Bewertungen $w(i, j)$ aller Gitterpunkte werden mit einem Aggregationsoperator zur Gesamtbewertung vereinigt. Ein Metropolis-Algorithmus ändert in zufälliger Weise die Lagen der Ketten im Gitter und behält diejenigen für die sich die Gesamtbewertung nicht verschlechtert hat.

Initialisierung:

Zu Anfang wird das Gebiet der Größe 97×97 mit geraden Ketten im Abstand von 3 Gitterlängen aufgefüllt. Jede Kette hat die Länge 4 und die Ladungsbelegung $(1, 0, -1, 0)$.

Testergebnisse:

In der Bildfolge wird das Endergebnis nach langer Zeitentwicklung dargestellt für die angegebenen Aggregationsoperatoren (vgl. nächste Seite).

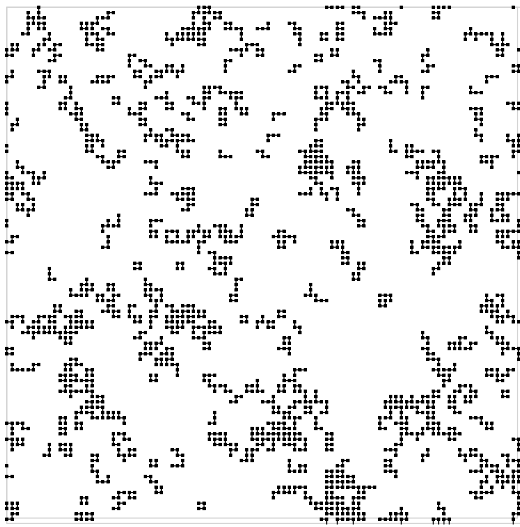
4.2 Diskussion und offene Fragen

Durch die vorgelegten Testergebnisse wird nachgewiesen, dass das Endergebnis des Metropolis Algorithmus vom verwendeten Aggregationsverfahren abhängig ist. Die Auswahl der Aggregationsoperatoren gibt daher zusätzliche Freiheitsgrade durch die das System an die empirischen Beobachtungen angepasst werden kann.

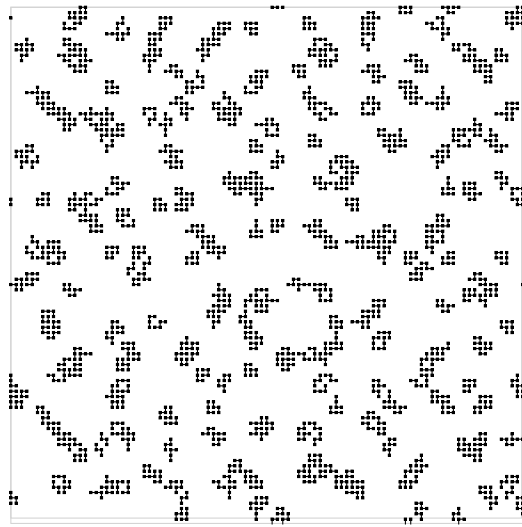
Wir haben bisher noch keine Tests mit der Zusatzbestätigung für Granuli (III) durchgeführt, da hier Probleme mit der Rechenzeit auftreten. Kontextabhängigkeit bedeutet Nichtseparabilität der Eigenschaften in verschiedenen Raumpunkten und ist damit ein Kennzeichen

für die Komplexität eines Systems. Dies spiegelt sich auch in der Simulation wider. Durchaus nicht alle komplex erscheinenden Systeme sind jedoch kontextabhängig. Von Vogelschwärmen dachte man lange Zeit, auf Grund ihres beeindruckenden Erscheinungsbildes, dass es sich hierbei um kontextabhängige Systeme handelt, bis Simulationsergebnisse nachwiesen, dass ihr Verhalten auch mit lokalen Modellen simulierbar ist. Wir wissen bisher nicht ob es möglich ist, auf Grunde des Erscheinungsbildes der Simulation zwischen kontextabhängigen und kontextunabhängigen Systemen zu unterscheiden. Wir kennen keine Phänomene, die nur in einer der Klassen vorkommen. Weitere Tests mit dem vorgestellten Simulationsprogramm sollen dieser Frage nachgehen.

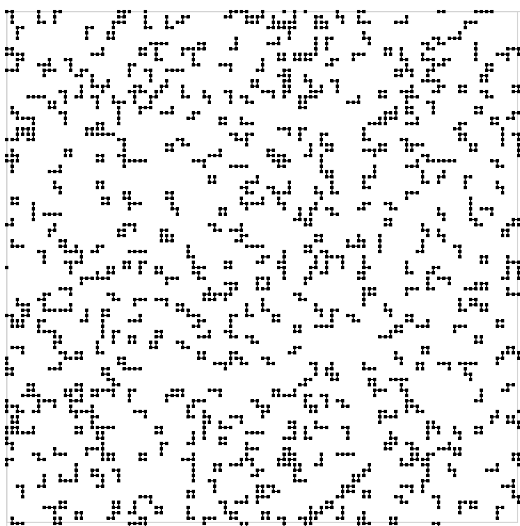
Wir wollen deutlich betonen, dass die im Grenzbereich zwischen Quantenmechanik und klassischer Physik auftretenden Fragen für die Physiker noch offen sind, und auch von uns nicht gelöst wurden. Wir glauben jedoch, dass das vorgestellte Programmsystem die Möglichkeit bietet, mit Simulationsexperimenten diese Fragen zu untersuchen.



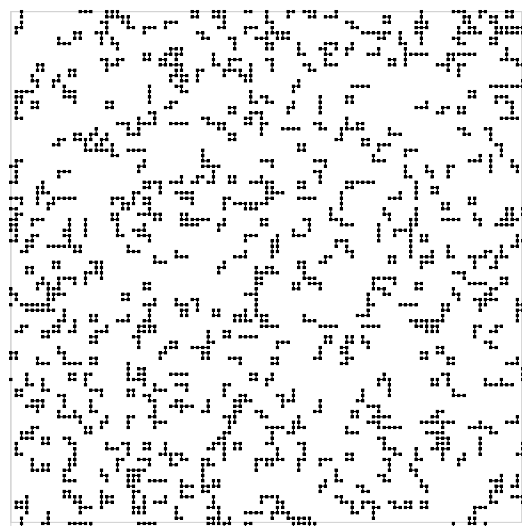
Max-Aggregationsoperator



Summen-Aggregationsoperator



Min-Aggregationsoperator



Allgemeiner Aggregationsoperator

4.3 Schlussbemerkungen

Die Nanotechnologie kann nicht auf einer bereits wohlbekannten physikalischen Grundtheorie aufbauen. Holistische Effekte und Emergenzphänomene werden hier wesentlich. Damit steht sie jedoch nicht allein. Auch in der Mikrobiologie und der Medizin häufen sich die Bedenken gegen eine rein reduktionistische Verstehensweise. (Vgl. [13] für Beispiele aus der Krebsforschung.) Granulierungsphänomene bestimmen neue Effekte. Unser Beitrag sollte aufzeigen, wie die Fuzzy Logik zum Verstehen dieser Phänomene beitragen kann.

Literatur

- [1] Zadeh, L.: *From imprecise to granular probabilities*. Fuzzy, Sets and Systems, 154, S.370-374, 2005.
- [2] Yager, R. R.: *QWA Aggregation Over a Continuous Interval Argument With Application to Decision Making*. IEEE Trans. Man and Cybernetics, B-34,5 S.1952-2158, 2004.
- [3] Ha Park, S.; Yan, H.; Reif, J. H.; LaBean, T. H.; Finkelstein, G.: *Electronic nanostructures templated on self-assembled DNA scaffolds*. Nanotechnology 15, S.525-527, 2004.
- [4] Friesner, R. A.; Guallar, V.: *Ab Initio Quantum Chemical and Mixed Quantum Mechanics/Molecular Mechanics Methods for Studying Enzymatic Catalysis*. Annu. Rev. Phys. Chem. 56, S.389-427, 2005.
- [5] Kapral, R.: *Progress in the Theory of Mixed Quantum-Classical Dynamics*. Annu. Rev. Phys. Chem. 57, S.129-157, 2006.
- [6] Sen, D.; Sengupta, S.: *A Critique of the Classical Limit Problem of Quantum Mechanics*. Foundations Phys. Letters. 57, S.129-157, 2006.
- [7] Arndt, M.; Hackermüller, L.; Hornberger, K.: *Wann wird ein Quantenobjekt klassisch?*. Phys. Unserer Zeit, 1,37, S.24-29, 2006.
- [8] Sommer, H. J.: *Zur Modellbildung kontextabhängiger Systeme*. In: *Proc., 15. Workshop Computational Intelligence* (Mikut, R.; Reischl, M., Hg.), S. 172-181. Universitätsverlag Karlsruhe. 2005.
- [9] Zeh, H. D.: *Where has all the information gone?*. Physics Letters A. 347, S.1-7, 2005.
- [10] Efros, A.: *Fisica y Geometria del Desorden*. Editorial MIR Moscu. 1987.
- [11] Sommer, H. J.: *Zur einfachen Darstellung eines allgemeinen Ansatzes der Wissensverarbeitung mittels Fuzzy Logik*. In: *Proc., 5. Workshop Fuzzy Control* (Kiendl, H., Hg.), S. 1-13. Forschungsbericht 0295,5 Universität Dortmund. 1995.
- [12] Kiendl, H.: *Fuzzy Control methodenorientiert*. München: Oldenbourg. 1997.
- [13] Soto, M. A.; Sonnenschein, C.: *Emergentism by Default: A View From the Bench*. Synthese 151, S.361-376, 2006.



Optimierung der Parameter für die Vagusnervstimulation mit KNN zur Behandlung refraktärer Epilepsie bei Kindern (OptiVaNeS)

A. Lodwich¹, B. Zaaimi², R. Grebe², J. Krone¹,
U. Lehmann¹, F. Wallois²

1. Fachhochschule Südwestfalen
Frauenstuhlweg 31
58644 Iserlohn
Tel. (02371) 566 465
Fax (02371) 566 270
E-Mail: lodwich@fh-swf.de

2. Université de Picardie Jules Verne
3 rue des louvels
80000 Amiens
Tel. 03 22 82 79 20
Fax 03 22 82 79 63
E-Mail: boubker.zaaimi@u-picardie.fr

1 Kurzbeschreibung

Refraktäre Epilepsien bei Kindern sind mit Medikamenten nicht erfolgreich behandelbar und können schwere Formen annehmen. Oft kann aber die Belastung der betroffenen Familien mit Hilfe eines Vagusnervstimulators (VNS) gemildert werden. Die Elektrode eines solchen Stimulators wird so in unmittelbarer Nähe des Vagusnervs (VN) eingepflanzt, dass die regelmäßig abgegebenen elektrischen Impulse eine Funktionsunterbrechung bzw. gezielte Reizung großer Teile des Nerven verursachen. Die Anordnung ist mit einem Herzschrittmacher vergleichbar. Der Vagusnerv selbst besteht aus verschiedenen Fasern, die unterschiedliche Funktionen erfüllen. Daher kann der Stimulator durch Veränderung der Impulsparameter, wie Frequenz, Stromstärke, etc. unterschiedliche Wirkungen erzielen. Optimal wäre hier eine gezielte Reizung der einzelnen Nervenfasern. Die explizite Ansteuerung von ganz bestimmten Faserarten setzt jedoch, bis dato klinisch nicht realisierbare, mikrochirurgische Eingriffe voraus [1].

Im Rahmen des OptiVaNeS Projekts wird daher systematisch die Wirksamkeit der Stimulationsparameter untersucht, die in der Praxis bereits erfolgreich zur Reduzierung der Anfallshäufigkeit eingesetzt werden. Dabei sollen die zahlreichen vom Arzt

wählbaren Einstellkombinationen aus Pulsstärke, Pulsdauer, Pulsweite, Anwendungszeit und Pause auf eine kleine Zahl häufig wirksamer Parameterkombinationen reduziert werden.

Die Verwendung des Stimulators erfolgt dabei unter hohen Auflagen und großer Vorsicht. Daher sind klinische Studien an Menschen sehr schwierig und langwierig. Tiermodelle (biologisches Modell für den Menschen) ermöglichen einen größeren Handlungsspielraum und schnellere Ergebnisse. Tiermodelle sind jedoch ebenfalls nicht ideal, da sie sich verändern und die Wirkung der VNS-Behandlung nur mittelbar anhand verschiedener Biosignale erfasst werden kann, die wiederum von Umwelt- und Befindlichkeitsfaktoren abhängen. Bis heute existiert kein allgemein akzeptiertes Erklärungsmodell für den Zusammenhang zwischen der Wirksamkeit von VNS und den aufgezeichneten Signalen. Das Projekt basiert hier auf der langjährigen Behandlungserfahrung der Ärzte.

Um schnelle Fortschritte zu erzielen und (un)systematische Fehler der verschiedenen Einflussfaktoren auf die Tiermodelle zu vermeiden, setzt das OptiVaNeS-Projekt auf die Hilfe des Computers und mathematischer Modelle auf Basis von Computational Intelligence (CI). Verschiedene Tools werden zum Data Mining und zum Training von Künstlich Neuronalen Netzen (KNN) mit vorliegenden Messreihen verwendet. Mit Hilfe der KNN kann ein Standardtiermodell für Ratten trainiert werden, das sich dann einer großen Zahl von Parametern aussetzen lässt. Aus den Ergebnissen lassen sich wiederum Behandlungsentscheidungsbäume erzeugen, die den Arzt bei der klinischen Überprüfung seiner Behandlung unterstützen. Für jede Behandlungs- und Zusammenhangsvermutung muss ein neuer Entscheidungsbaum erzeugt werden.

Das OptiVaNeS Projekt will die erfolgreichsten Parameterkombinationen für die Behandlung von Kindern finden und gleichzeitig die Verifikation oder Falsifikation der vermuteten Zusammenhänge erbringen. Sollten die Schritte bis zu diesem Stand ermutigend und erfolgreich verlaufen, können neue Typen von intelligenten Stimulatoren gebaut werden, die anhand der Beobachtung von Biosignalen jeweils die optimale Behandlung selbstständig und rechtzeitig wählen. Dazu könnten z.B. CI-Regler [2] verwendet werden.

2 Kooperation

2.1 Allgemein

Das Projekt OptiVaNeS ist ein kooperatives anwendungsorientiertes Forschungsprojekt zwischen der Universität Jules Verne aus Amiens (Frankreich) und der Fachhochschule Südwestfalen, Standort Iserlohn (Deutschland) mit finanzieller Unterstützung im Rahmen des europäischen PROCOPE. Die Kooperation wird zusätzlich durch die durch das Land NRW geförderte Kompetenzplattform *KOPF Computer Vision based on Computational Intelligence* gefördert. An der Fachhochschule Südwestfalen wird ein gleichnamiges Masterstudium angeboten.

2.2 Ausgangslage

Vagusnervstimulation (Vagus Nerve Stimulation, VNS) ist die einzige zugelassene elektrische Behandlung im Zusammenhang mit Epilepsie. Verwandte Methoden sind ECT (Electro Convulsive Therapy), MST (Magnetic Stimulation Therapy), TMS (Transcranial Magnetic Stimulation), und DBS (Deep Brain Stimulation). [3]

Der Einfluss der fünf Parameter der Stimulation auf den Verlauf der Krankheit ist unbekannt. Experimentelles Anwenden der gleichen Therapie führt manchmal zu fast vollständigem Verschwinden der Symptome und in anderen Fällen verbleibt sie ohne Wirkung. Man vermutet, dass bestimmte Parameterkombinationen durch Einwirkung auf die Atmung einen wechselnden Sauerstoffpegel im Blut hervorrufen und damit einen neuroprotektiven Mechanismus im Gehirn auslösen. Die Stimulation des Vagusnervs kann allerdings zu Herzrhythmusstörungen führen, was vermieden werden soll.

2.3 Ziele

Das Ziel der Kooperation ist es die wirksamsten Parameter der Vagusnervstimulation anhand eines Computermodells herauszufiltern. Aus einer großen kombinatorisch erzeugten Menge wurden 81 Parameterkombinationen für das Projekt ausgewählt. In der Praxis können Parameteränderungen am Menschen nur alle drei bis vierundzwanzig Wochen durchgeführt werden. Aufgrund der Gesamtdauer lässt sich eine Analyse am Menschen nicht durchführen.

Durch Modellierung eines Tiermodells (z.B. einer Ratte) kann eine erheblich höhere Anzahl von Stimulationsparametern zeitnah ausprobiert werden. Tiermodelle sind Tiere, die einen bestimmten biologischen Aspekt mit dem Mensch gemeinsam haben. Ein Künstlich Neuronales Netz (KNN) erfordert keine detaillierte Modellierung dieses Tieres, sondern bildet es ausreichend genau anhand gemessener Biosignale ab. Über Kennlinienanalyse des KNN können bedingte Aussagen über die Wirkung der VNS gemacht werden, die sich in einem Entscheidungsbaum niederschlagen.

Langfristig will man statische Stimulatoren durch dynamische Geräte ersetzen. Solche Geräte können dann automatisch die passende Parameterkombination wählen, wenn der gewünschte Effekt zu verschwinden droht.

3 Refraktäre Epilepsie

3.1 Epilepsie

Epilepsie wird festgestellt, wenn mindestens zwei epileptische Anfälle beobachtet werden konnten, die durch paroxysmale (lähmende) synchrone Entladungen von Neuronengruppen hervorgerufen werden. Diese führen zu plötzlichen für die Epilepsievariante stereotypen Verhaltens- oder Befindungsstörungen. [4][5][6][7]

3.2 Epileptische Anfälle

Epileptische Anfälle treten plötzlich auf. Manche Patienten können ein baldiges Auftreten wegen einem vorlaufenden lokalen Anfall spüren. In solchen Fällen kann die gezielte Vagusnervstimulation verwendet werden. In aller Regel wird ein epileptischer Anfall durch den Ausfall von Bewusstsein begleitet und kann bis zu 60-mal am Tag auftreten.

Dabei werden die Anfälle in drei Arten unterschieden:

- Konvulsiv
 - Sichtbares Zittern am ganzen Körper (generalisiert), Stürze
- Nonkonvulsiv
 - z.B. Abscencen
- Myoklonisch
 - Unregelmäßiges Zucken oder Verdrehen einzelner Körperteile

3.3 Epileptische Krankheitsbilder

Epilepsien können entweder nach Symptomen oder nach Ursachen klassifiziert werden. Es ist sehr schwierig sie richtig zu klassifizieren, wie die nachstehende Tabelle belegt:

**Klassifikation von Epilepsiesyndromen auf klinischer Basis,
mit EEG, und Neuroimaging [14]**

<i>Anfallstyp</i>	<i>Klinisch ohne Apparate, %</i>	<i>Klinisch + EEG, %</i>	<i>Klinisch + EEG + Kernspin, %</i>
Generalisiert	8	23	23
Partial	39	54	58
Nicht klassifiziert	53	23	19

3.4 Heilungsaussichten

Es bestehen unterschiedliche Aussichten in Abhängigkeit davon wie man die Epilepsie erworben hat. Juvenile (jugendliche) Epilepsien müssen selten behandelt werden und ihre Symptome verschwinden oft schon nach einem einzigen Anfall. Idiopathische Epilepsien sind dagegen genetisch bedingt und sind daher am schwierigsten zu behandeln. Epilepsien werden entweder durch verschiedene Antikonvulsiva, Reizvermeidung, chirurgische Eingriffe, ketogene Diäten und VNS behandelt. Epilepsievarianten, die sich nicht durch Medikamente behandeln lassen sind meistens unprovoked generalisierte Varianten, die man refraktär nennt. Für refraktäre Epilepsien bestehen bis jetzt keine wirklich wirksamen Behandlungsmethoden.

4 Vagusnervstimulation

4.1 Geschichte

Die Vagusnervstimulation wurde 1988 nach einem dreijährigen Test an Tieren zum ersten Mal am Menschen ausprobiert. 1992 hat man die anfallhemmende Wirkung an Säugetieren demonstrieren können. [8] 1997 wurde die Methode in Kombination mit Medikamenten zur Behandlung von Epilepsie von der U.S. Food and Drug Administration (FDA) genehmigt. [13]

Langzeituntersuchungen haben 1994ff ergeben, dass während der ersten 12 Monate die Effizienz der Behandlung zunimmt und dass eine breite Palette refraktärer Epilepsien damit behandelt werden können. [9][10][11][12] Neue Anwendungsgebiete eröffnen sich in der Behandlung refraktärer Depressionen. [15]

4.2 Anwendung

Der Vagusnervstimulator besteht aus einem Gehäuse mit Batterie, Steuerelektronik und einer Elektrode, die um den Vagusnerv gewickelt wird. Das Gerät wird unter der Haut implantiert und ähnelt dem Wirkungsprinzip und Einsatz eines Herzschrittmachers.

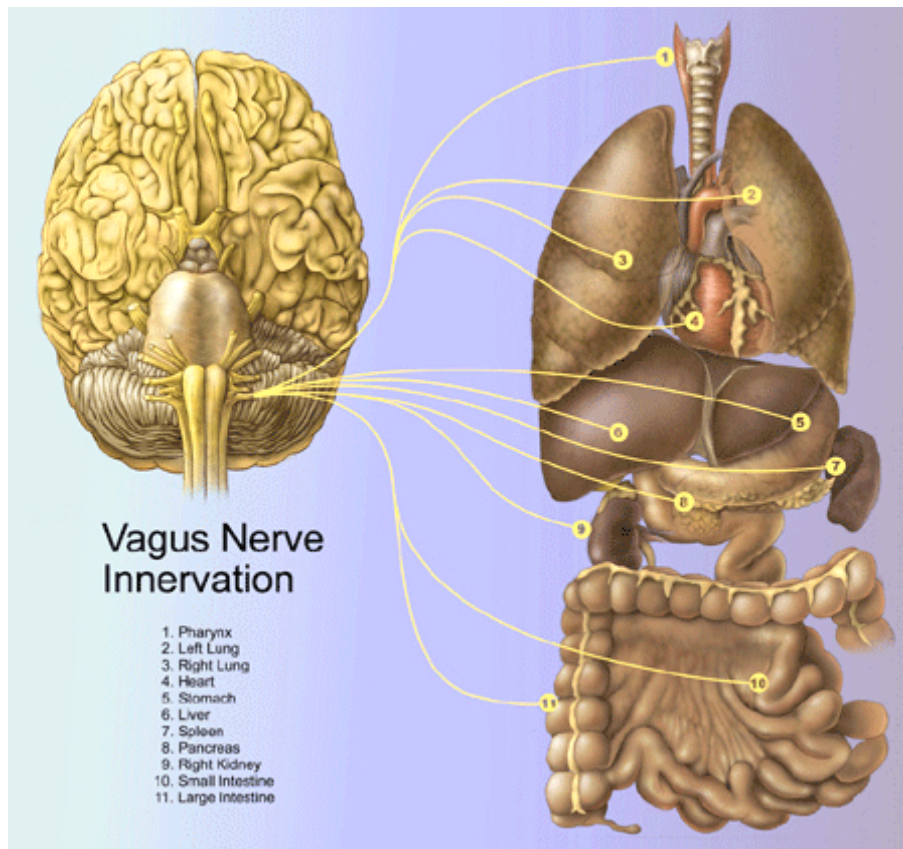


Abbildung 1: Der Vagusnerv steuert innere Organe [15]

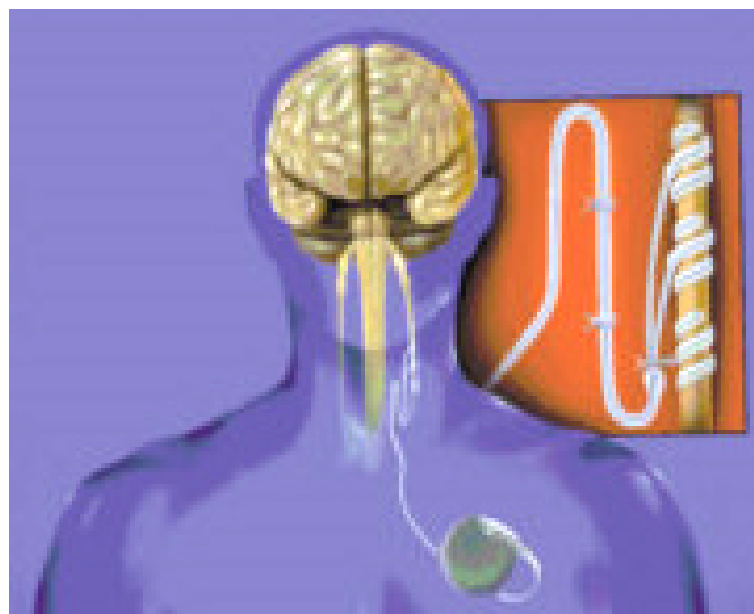


Abbildung 2: Implantierter Vagusnervstimulator [13]



Abbildung 3: Stimulator der Firma Cyberonics, Inc. Modell *pulse 102* mit Steuergerät und Antenne.

Die Einstellung der Parameter erfolgt über eine Antenne, die an ein Steuergerät angeschlossen wird. Die Parameter können aus einer Liste ausgewählt werden. Es ergeben sich einige hundert Möglichkeiten den Stimulator einzustellen.

5 Daten

5.1 Quellen

Das KNN wird mit Daten aus Messungen an gesunden Ratten trainiert. Im Rahmen des Projekts werden ca. zwei Dutzend Ratten aufgezeichnet, um eine statistische Sicherheit zu erhalten. Die Messungen beinhalten Elektrokardiographie (ECG), Plethysmographie (Barometrie, PG) und die Aktivitäten des Stimulators (VNS Artefakt), da die Stimulation mit den Auswirkungen zeitlich korreliert werden soll. Die Daten müssen von Hand vorgefiltert werden, da die Störungen vielseitig sind. Unerkannte Wachaktivität, Rauschen, Signalschwinden und starkes Übersprechen zwischen den Kanälen sind die wichtigsten parasitären Effekte, die einer guten Aufzeichnung entgegenstehen.

Nachdem die Daten manuell in einzelne Kanäle getrennt worden sind, übernimmt ein Matlab-Programm die weitere Verarbeitung. Die Originaldaten bestehen aus drei Volumen zu je drei Kanälen mit jeweils 10 Millionen Datensätzen bei einer Samplingrate von ca. 1 kHz und einigen Metadaten-Dateien für jede Ratte und jedes Volumen. In den Metadaten werden irrelevante Datenbereiche markiert und Stimulationsparameter zeitlich zu den Aufnahmen zugeordnet. Irrelevante Datenbereiche entstehen z.B. durch Wachaktivitäten der Ratten oder durch das Öffnen des barometrischen Käfigs (PG). Die erwartete Summe der Datensätze zur Auswertung

wird ca. 600 Millionen betragen. In kompakter Form sind das 1,8 Milliarden Samples zu je 4 Byte.

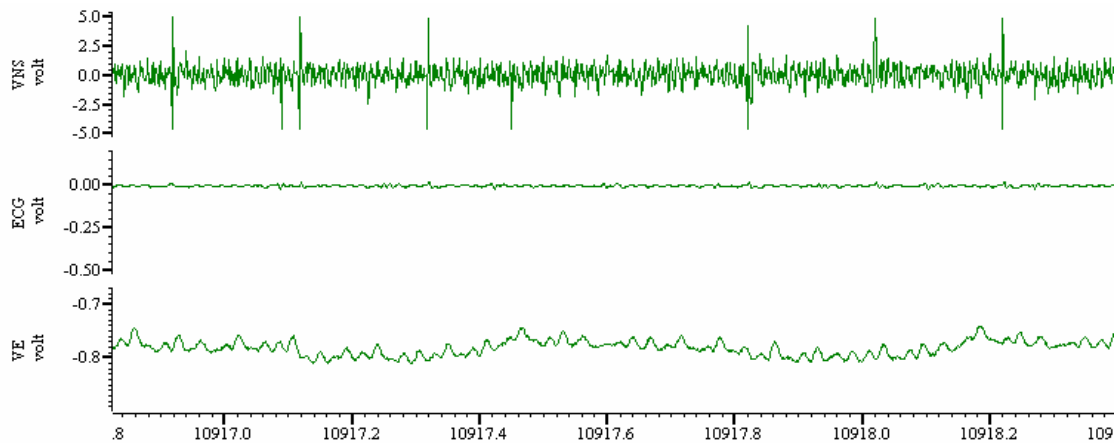


Abbildung 4: Ungefilterte Daten offenbaren die schwierige Aufzeichnungsrealität. Sporadisches Auftreten des VNS, fast abwesendes ECG auf entsprechendem Kanal und starkes ECG Artefakt auf dem PG Kanal.

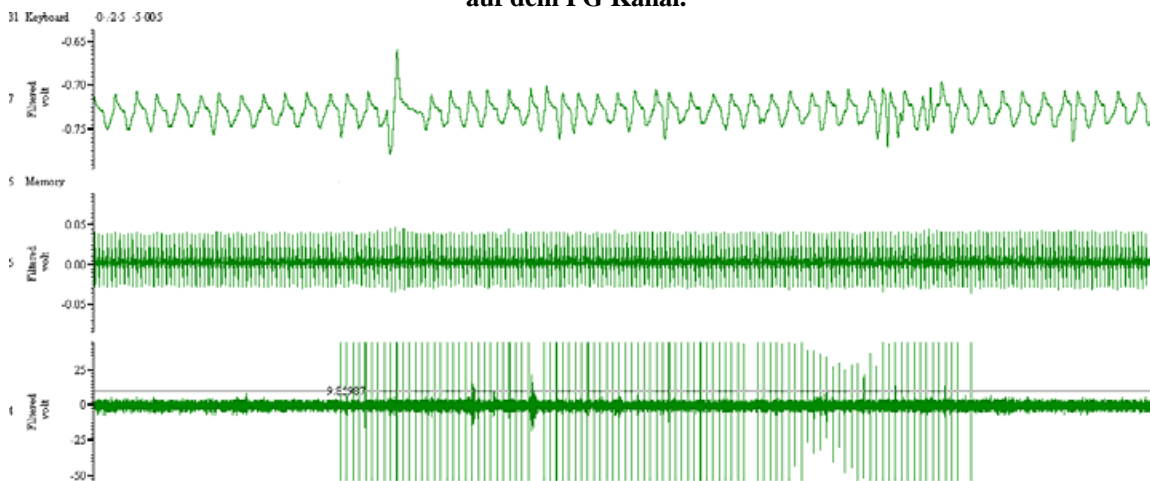


Abbildung 5: Gefilterte Aufzeichnung von ca. 1 Minute. Von oben nach unten: PG, ECG und der VNS Artefakt

5.2 Enthaltene Informationen

Die drei Informationskanäle werden zunächst um fünf weitere aus den Metadaten erweitert. Diese beinhalten die korrespondierende Pulsbreite, Stromstärke, Pulsfrequenz sowie die An- und Auszeit. PG, ECG und VNS werden analysiert und in neue Kanäle zerlegt: VR (Ventilation Rate), VA (Ventilation Amplitude), VS (Ventilation Slope), VI (Inhalation Time), VX (Exhalation Time), VT (Minute Ventilation), HR (Heart Rate), t1 (Zeit ab Stimulationsbeginn), t2 (Zeit ab Stimulationsende). Die Originalkanäle werden verworfen. Damit bleibt ein Datenbestand von ca. 8,4 Milliarden Samples die mindestens 4 Byte breit sind.

5.3 Vorverarbeitung

Die Zahl der Datensätze ist für ein sinnvolles Training zu groß. Mehrere Schritte sind notwendig, um die Menge zu verringern. Im ersten Schritt werden Fehlerbereiche ausgeschnitten, die ca. 10% bis 20% der Daten ausmachen. Danach wird eine Reduktion der Samplerate um Faktor 8 durchgeführt. Somit ist jeder Herzschlag mit mindestens einer Herzfrequenzangabe vertreten.

Die Daten müssen geglättet werden weil die Messung der Ereignisse zu scharf ist. Das Herz ist relativ schwer von seiner Frequenz abzulenken und wird während der Stimulation nur bei einzelnen Herzschlägen verzögert. Für die Atmung gilt ähnliches. Damit man Tendenzen im Verlauf der Signale erkennt, werden die Daten durch einen Tiefpassfilter (Glättung) gefiltert. Anschließend wird für jede Parameterkombination ein statistischer Musterverlauf aus ca. 100 Zyklen erstellt. Zuletzt werden die Kanäle um Faktor 100 in ihrer Samplingrate reduziert, weil durch Glättung massive Redundanzen entstanden sind und kein Fokus mehr auf den Einzelereignissen liegt. Der gemessene Fehler durch die Reduktion beträgt $\pm 2,93\%$. Die so erzeugten Daten werden zu einer Datei zusammengelegt und bilden die Grundlage für das Training der KNN.

5.4 Qualität der Daten

Die Qualität der Daten ist für jede Stimulation unterschiedlich gut. Man muss mit widersprüchlichen Effekten bei gleichen Einstellungen rechnen, da sie von verschiedenen Ratten und unterschiedlichen Zeitpunkten stammen. Die Daten eignen sich nicht für ein Training von rekurrenten Netzen, da die Datensätze nicht zwingend im zeitlichen Zusammenhang stehen. Abhilfe leisten die Parameter T1 und T2, die einem zeitunabhängigen Netz bei der zeitlichen Konzeptionalisierung der Effekte helfen.

6 Konventionelle Analyse

6.1 Verbesserung der konventionellen Analyse

Das Projekt hat bereits die Verbesserung der konventionellen Analyse ermöglicht. Das Konstruieren der für den manuellen Vergleich erforderlichen Graphen erfolgt ohne Aufsicht. Als Resultat liegen rund 240 Graphen in Form eines Durchschnittsverlaufs und eines Graphen mit der Erhebungsbasis. Vorher musste der statistische Vorgang von Hand für jede Parameterkombination einzeln durchgeführt werden.

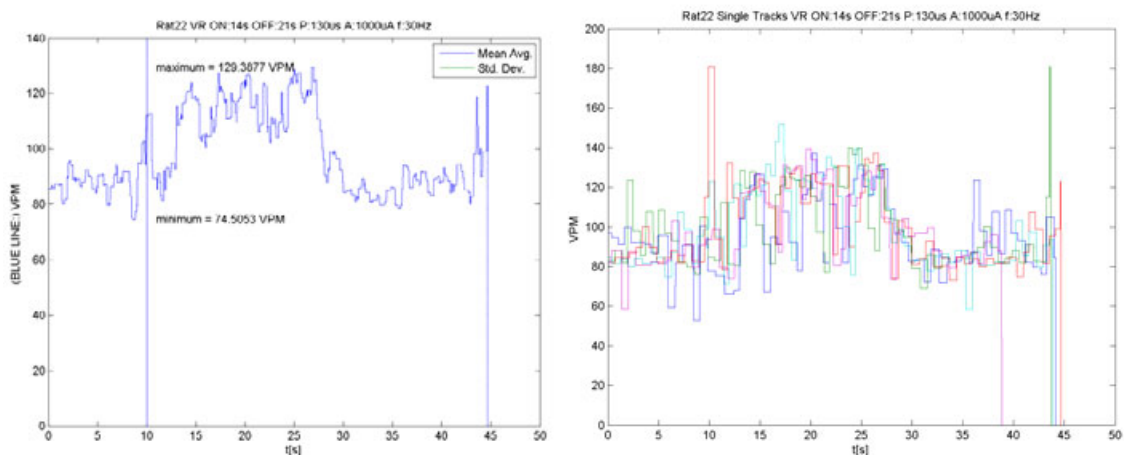


Abbildung 6: Musterverlauf (links) und seine Datenbasis (rechts)

Zusätzlich entstehen Tabellendokumente mit detaillierten statistischen Informationen.

6.2 Direkter Vergleich

Die Auswahl eines Parameters erfolgte bisher immer durch direkten Vergleich der Graphen. Der Vergleich ist zeitlich aufwendig weil für jede Parameterkombination rund 20 Graphen auf ihre Grundtendenz verglichen werden müssen.

Ohne direkten Vergleich lassen sich auch schon Entscheidungsbäume für die Therapie mit Hilfe eines Data Mining Tools wie dem InfoMiner [16] von der Otto-von-Guericke-Universität in Magdeburg erstellen. Allerdings erhält man damit nur Punktwissen und es ist schwierig Transzendenzen und Optima zu erkennen.

7 Neuronale Netze

7.1 Gründe für KNN

Künstlich Neuronale Netze können das Problem des Punktwissens überwinden. Aus den konfliktbehafteten Datensätzen können durch Training die wahren Tendenzen bei den Parametern abgebildet werden.

Darüber hinaus umgeht man das Problem der Modellierung der Wirkungsweise eines Vagusnervs, seines Aufbaus und der Probleme der Ladungskonvektion.

Aus einem KNN können ganz leicht tendenzielle Verlaufskurven in einer Qualität generiert werden, wie man sie durch Glättung nicht erhält.

Durch Kennlinienanalyse können Einflussfaktoren auf die Befindlichkeit des Tiermodells untersucht werden. Man erhofft sich Aussagen über Bedingungen, unter denen andere Parameter an Bedeutung gewinnen oder verlieren. Optima außerhalb des Trainingsbereichs können gefunden werden.

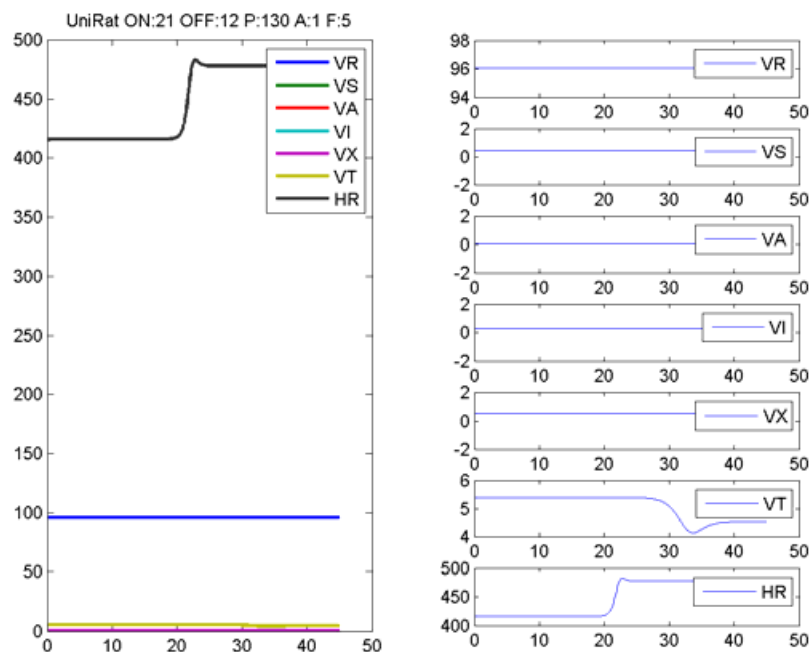


Abbildung 7: Ein aus dem trainierten KNN generierter Verlauf für die Parameter ON:21s OFF:12s P:130us A:1mA F:5Hz. Man erkennt massiven Einfluss des Parameters auf das Herz und einen geringen bis keinen Einfluss auf die Atmung. Die Auswirkungen zeigen sich sprunghaft nach dem Ende der Stimulation.

7.2 Architektur

Da die Daten ein Training mit rekurrenten Netzen nicht ohne weiteres zulassen, können ausschließlich vorwärtsgerichtete Netze verwendet werden. Das Gesamtnetz besteht aus sieben Einzelnetzen für die jeweiligen Ausgänge, die alle 7 Eingänge besitzen (ON, OFF, P, A, F, T1, T2). Dadurch sind die Targetneuronen relativ stark entkoppelt.

Die Einzelnetze sind alle gleich und werden von einer Vorlage zu einem Array zusammengelegt.

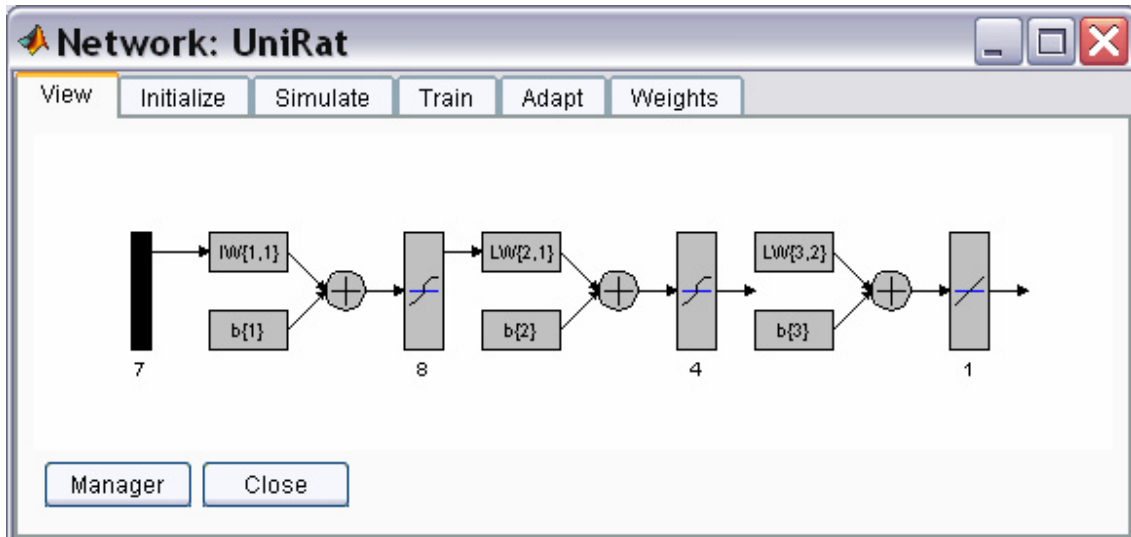


Abbildung 8: KNN Template

Das Ausgangsneuron enthält immer eine lineare Übertragungsfunktion, während die restlichen Neuronen durch eine sigmoide Übertragungsfunktion gekennzeichnet sind. Das KNN ist im Vergleich zu der Zahl der Datensätze sehr klein. Aussagen über die Qualität der Größenwahl werden noch im Verlauf des Projekts geklärt werden, wenn andere KNN-Vorlagen ausprobiert werden.

7.3 Training

Das Training erfolgt mit der Neural Network Toolbox für Matlab und wird vom Programm als vorletzter Programmschritt durchgeführt. Die Daten werden zwar von Matlab für die Eingänge normiert, aber nicht für die Ausgänge. Sicherheitshalber führt das Steuerprogramm für den Prozess eine eigene Normierung für alle Daten aus.

Es werden keine Test oder Validierungsdatensätze bereitgestellt, da der Prozess des Trainings erfahrungsgemäß nach 25 Epochen mit dem Resilient-Propagation keine relevanten Änderungen mehr ins KNN hineinbringt. Abbruch des Trainings erfolgt grundsätzlich nach 100 Epochen. Bei dem großen Verhältnis zwischen Datensätzen und Verbindungen besteht keine Gefahr des Auswendiglernens.

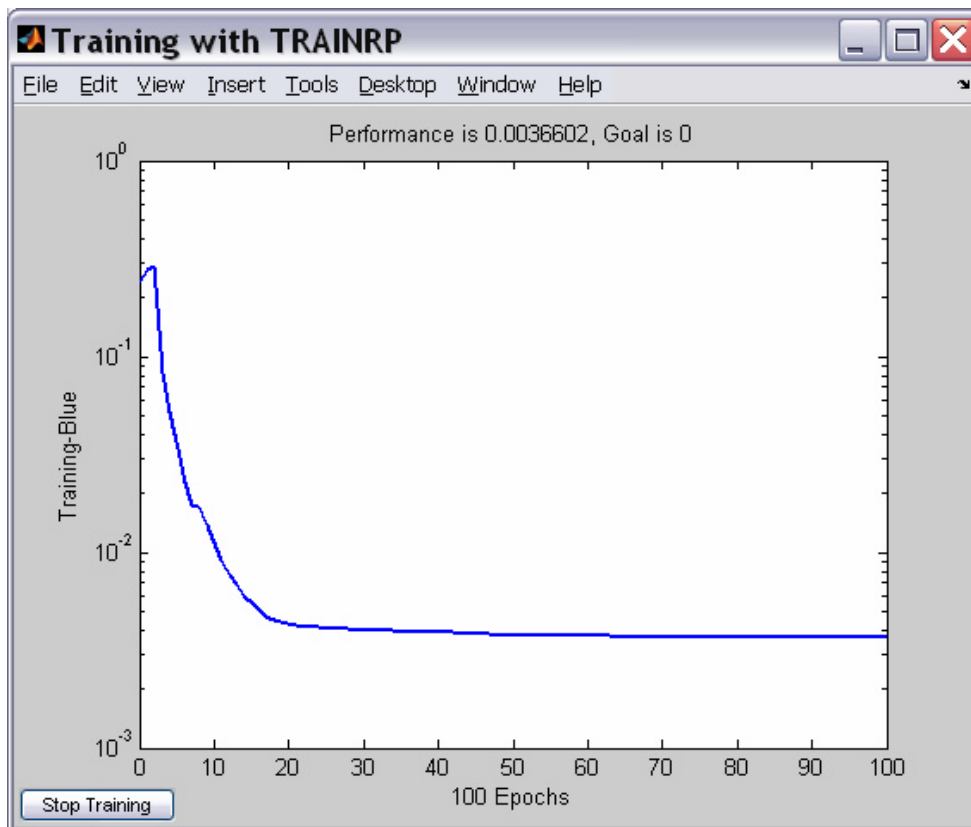


Abbildung 9: Typischer Verlauf des Trainingvorgangs (Resilient Propagation).

8 Entscheidungsbäume mit InfoMiner 1.2 [16]

8.1 Unklassifizierte Daten

Mit dem InfoMiner 1.2, einem frei verfügbaren Werkzeug, können Häufungen bestimmter Eigenschaften in Abhängigkeit einiger Eingangsparameter bestimmt werden. Dazu müssen sie nicht speziell vorverarbeitet werden.

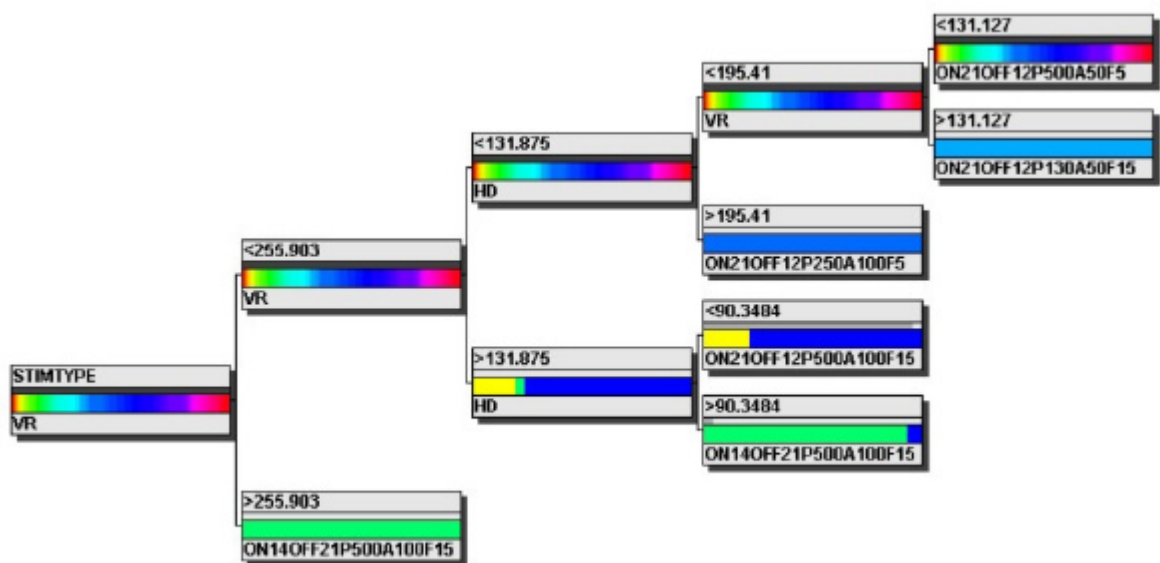


Abbildung 10: Ein ganz kleiner Entscheidungsbaum zeigt die nach einander zu treffenden Entscheidungen. Die Blätter des Baums zeigen die dazugehörige Parameterkombination an.

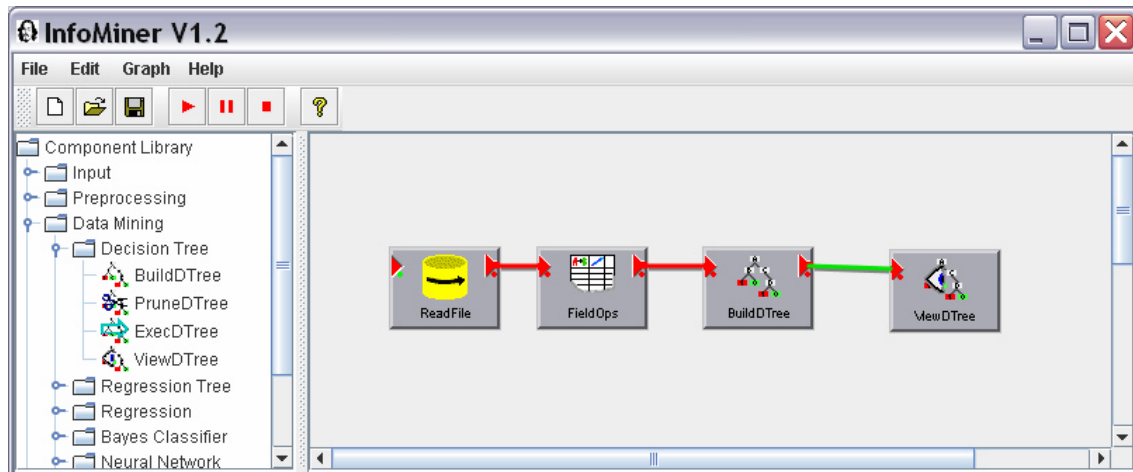


Abbildung 11: Der InfoMiner kann verschiedene Algorithmen mit einander kombinieren. Entscheidungsbäume, Künstliche Neuronale Netze, Bayes Filter uvm. In diesem Beispiel wird aus einer Datei gelesen und diese Daten dann zur Entwicklung eines Entscheidungsbaumes verwendet.

8.2 Klassifizierte Daten

Dank des trainierten KNN können Verlaufstendenzen für die jeweiligen Parameterkombinationen einfacher analysiert und daher klarer in begrifflicher Form ausgedrückt werden. So können Entscheidungsbäume nach Kriterien wie „Starke Erhöhung“ oder „Keine Wirkung“ iteriert werden und nicht durch Vergleichen intransparenter Zahlen.

9 Klinische Untersuchung

9.1 Hintergrund

Die grundsätzlich geringen Stimulationsenergien führen nicht unmittelbar zur Beeinflussung des Krankheitsverlaufs. Epileptische Anfälle werden nicht abgebrochen. Die Stimulation erfolgt rein prophylaktisch, aber die Begründung für ihre Wirkung ist nicht genau bekannt. Man vermutet, dass ein neuroprotektiver Mechanismus durch den schwankenden Sauerstoffgehalt im Blut aktiviert wird. Daher werden Stimulationsantworten in Form verringerter Atmungsaktivität und normaler Herzaktivität als besonders prädestiniert für die Behandlung angesehen.

Ein festes Einkreisen der häufigsten Parameter für eine bestimmte Art der Herz- und Lungenfunktionsantwort kann nicht gänzlich durch Computersimulation als Phantomergebnis ausgeschlossen werden, da die Positionierung und Wirkung der VNS-Elektrode in keiner Weise modelliert wird.

Abhilfe bringt nur die prototypische Behandlung von epileptischen Ratten. Nur wenn die Anzahl der Anfälle in einer klinischen Untersuchung besonders niedrig ausfällt, ist man am unmittelbaren Ziel angekommen und kann dann im nächsten Schritt nach Randbedingungen für ein Optimum suchen.

Wenn die gewählten Parameter in der Praxis nicht hinreichen, dann muss man entweder die Datenbasis erhöhen oder andere Körperparameter aufzeichnen, z.B. Elektroenzephalographie (EEG).

10 Literatur

- [1] A. Lodwich, B. Zaamini, R. Grebe, J. Krone, U. Lehmann, F. Wallois
Optimization of Vagus Nerve Stimulation for Epileptic Children with ANN
Vortrag, Iserlohn, 16. Januar 2006
- [2] U. Lehmann, J. Krone, C. Beckert, S. Dormeier, T. Mund, R. Schamne,
M. Pearaudin :
Computational Intelligence Controller für nichtlineare Regelstrecken.
Forschungsland NRW, Informationsbroschüre zur Hannover Messe Industrie.
Hannover April 2003.
- [3] Mark S. George, M.D. *New Methods of Brain Stimulation (ECT, MST, TMS,
VNS and DBS) Are Improving OCD Research and Therapy*
<http://www.musc.edu/fnrd/dbscns.htm>, CNS Spectrum, 2000
- [4] Hartmut Siemes, Blaise F. D. Bourgeois: *Anfälle und Epilepsien bei Kindern
und Jugendlichen.* Thieme, Stuttgart-New York 2001, ISBN 3-13-127031-4
- [5] Rolf Degen: *Epilepsien und epileptische Syndrome im Kindes- und
Erwachsenenalter. Elektroenzephalographie.* Blackwell, Berlin-Wien 1999,
ISBN 3-89412-366-4
- [6] Ansgar Matthes, Hansjörg Schneble: *Epilepsien - Diagnostik und Therapie für
Klinik und Praxis.* Thieme, Stuttgart-New York 1999, ISBN 3-13-454806-2
- [7] http://de.wikipedia.org/wiki/Epilepsie#Epilepsie_bei_Haustieren
- [8] Zabara J. *Inhibition of experimental seizures in canines by repetitive vagal
stimulation.* Epilepsia 1992; 33:1005-1012
- [9] Ben-Menachem E, et al. *Vagus nerve stimulation for treatment of partial
seizures: A controlled study of effect on seizures.* First International Vagus
Nerve Stimulation Study Group. Epilepsia. 1994 May-Jun;35(3):616-26. [PMID:
8026408]
- [10] Handforth A, et al. *Vagus nerve stimulation therapy for partial-onset seizures: a
randomized active-control trial.* Neurology 1998 Jul;51(1):48-55. [PMID:
9674777]
- [11] George R, et al. *Vagus nerve stimulation for treatment of partial seizures: 3.
Long-term follow-up on first 67 patients exiting a controlled study.* First
International Vagus Nerv Stimulation Study Group. Epilepsia. 1994 May-
Jun;35(3):637-43. [PMID: 8026410]
- [12] DeGiorgio CM, et al. *Prospective long-term study of vagus nerve stimulation for
the treatment of refractory seizures.* Epilepsia 2000 Sep;41(9):1195-200.
[PMID: 10999559]
- [13] <http://www.epilepsy.com>
- [14] http://www.neuro24.de/epilepsie_syndrome.htm
- [15] <http://pni.unibe.ch/vns.html>
- [16] <http://fuzzy.cs.uni-magdeburg.de/>

Adaptive Klassifikation von Zellbildern

Nicolas Cebron und Michael R. Berthold

Universität Konstanz

Box M712, 78457 Konstanz

Tel.: (07531) 88-4719

Fax: (07531) 88-5132

E-Mail: {cebron, berthold}@inf.uni-konstanz.de

1 Einleitung

In vielen unterschiedlichen Anwendungsbereichen führt der rasante Anstieg von Datenmengen zu Problemen bei der Vorklassifizierung ausreichend vieler, repräsentativer Beispieldaten. In dieser Arbeit widmen wir uns dem Bereich der Bioinformatik, bei dem die Entwicklung von Hochgeschwindigkeitskameras zur Analyse von Zellbildern neue Einsatzmöglichkeiten im Bereich der Wirkstoffanalyse eröffnet. Diese Geräte sind in der Lage, hunderttausende Zell-Assay Bilder innerhalb eines einzigen Tages zu produzieren. Die Analyse jedes einzelnen Bildes durch einen Biologen ist dabei kaum noch möglich. Daher werden bisher in aufwändiger Kleinarbeit spezielle, an das Problem angepasste Skripte zur Klassifikation solcher Bilder verwendet.

Das Ziel unserer Arbeit ist es, den Prozess der Zellbild-Klassifikation zu automatisieren. Dazu werden eine Handvoll interessanter Beispiele in den Bilddaten selektiert und dem Experten präsentiert. Basierend auf diesen wenigen, hand-klassifizierten Beispielen soll ein Modell gelernt werden, das anschließend in der Lage ist, den Rest der Bilddaten zu klassifizieren.

Da wir zu Beginn des Prozesses keine (klassifizierten) Trainingsdaten haben, benötigen wir ein Modell, welches Brücken schlägt zwischen dem Konzept des unüberwachten Lernens und dem Konzept des überwachten Lernens. Zu diesem Zweck stellen wir in dieser Arbeit ein neues aktives Cluster- und Klassifikationsschema vor, welches basierend auf einem initialen Fuzzy-c-means clustering die Daten gruppiert und erste Muster zur manuellen Klassifikation auswählt. Anschließend werden die gefundenen Prototypen mit Hilfe des Verfahrens der Lernenden Vektor Quantisierung (LVQ) an das Modell des Biologen angepasst. Inspiriert durch das Konzept des aktiven Lernens wird hierbei versucht, möglichst nützliche und sinnvolle Beispiele zu selektieren, um so die Überwachung durch den Biologen und damit den Aufwand möglichst gering zu halten.

Desweiteren präsentieren wir ein Framework zur Analyse von Zellbildern und vergleichen die Klassifikationsergebnisse auf einem Zelldatensatz mit anderen aktiven Lernansätzen.

2 Mining von Zellbildern

Um ein Zell-Assay-Bild im Ganzen zu klassifizieren, sind einige Vorverarbeitungsschritte notwendig. Abbildung 1 stellt den Ablauf schematisch dar. Die einzelnen Schritte werden in den nächsten Abschnitten genauer erläutert:

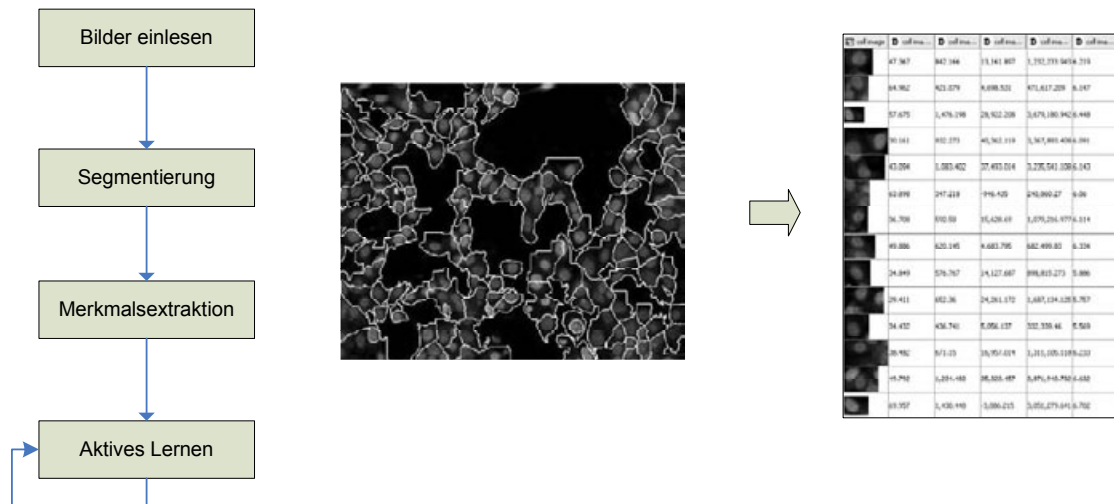


Bild 1: Arbeitsablauf bei der Klassifikation von Zell-Assays.

Segmentierung: Da die Reaktionen der einzelnen Zellen auf den Wirkstoff im Bild unterschiedlich ausfallen können, muss jede einzelne Zelle im Bild gesondert betrachtet werden. Die Gesamtklassifikation des Bildes erfolgt durch eine Mehrheitsentscheidung über die Einzelbilder. Bei Tausenden von Originalbildern, welche jeweils in hunderte einzelner Zellbilder zerlegt werden, erreichen wir eine Größenordnung von Millionen Bildern. Bei den bisherigen Bildern sind die markierten Zellkerne die Ausgangspunkte für ein *Region-Growing* Verfahren [5]. Dieses Verfahren setzt ein Distanzmaß ein, das sowohl lokale Features im Bild als auch die euklidische Distanz berücksichtigt.

Merkmalsextraktion: Um die einzelnen Zellbilder aus verschiedenen Sichten zu beschreiben, stehen verschiedene Module zur Merkmalsextraktion zur Verfügung. So lassen sich numerische Daten bezüglich des Histogramms eines Bildes erzeugen, die die Helligkeit und ihre Verteilung im Bild beschreiben. Zur Beschreibung der Textur einer Zelle gibt es ein Modul, das die 14 Haralick-Features [18] berechnet. Ein weiteres Feature-Modul berechnet die Verteilung der Grauwerte entlang eines Vektors durch das Bild. Um die Form einer Zelle zu beschreiben, können auch die Zernike-Momente [19] berechnet werden.

Aktives Lernen Der zu klassifizierende Datensatz besteht aus Millionen einzelner kleiner Zellbilder, wobei jedes Zellbild durch seine numerischen Features beschrieben wird. Das Ziel unserer Arbeit ist es, jedes einzelne Bild einer Zelle zu klassifizieren, um dann alle Gesamtbilder des Zell-Assays klassifizieren zu können. Das Problem hierbei ist, dass wir zu Beginn nicht über klassifizierte Trainingsdaten verfügen, so dass eine Initialisierung recht schwierig ist. Wir haben jedoch die Möglichkeit, einem Biologen ein Zellbild zu zeigen um anschließend einen Klassennamen für diese Zelle zu erhalten. Da die Anzahl der nicht-klassifizierten Daten sehr groß und die Kapazität des Biologen nicht unbeschränkt ist, benötigen wir eine Strategie, um möglichst sinnvolle Beispiele aus dem Datensatz auszuwählen. Das hierfür entwickelte adaptive Klassifikationsschema stellen wir in Kapitel 4 vor.

3 Aktives Lernen

Bei vielen Klassifikationsaufgaben in industriellen Anwendungen stößt man auf das Problem, dass eine große Anzahl von nicht klassifizierten Daten U zur Verfügung steht. Gleichzeitig sind die Kosten, um die Klasse von einem Beispiel zu bestimmen, sehr hoch. Das Konzept des aktiven Lernens [11] versucht, dieses Problem zu lösen, indem es einen Lernalgorithmus in die Lage versetzt spezifische Beispiele aus den unklassifizierten Daten auszuwählen. Diese werden dann von einem "Orakel" klassifiziert.

Gegeben sei die Menge der nicht klassifizierten Daten U , eine kleine Menge mit klassifizierten Trainingsdaten L und eine Menge von möglichen Klassen C . Ein aktiver Lerner wird dann als Tupel (f, q) beschrieben, wobei $f : L \mapsto C$ der Klassifizierer ist, der auf den klassifizierten Trainingsdaten gelernt wurde. Die Auswahlfunktion q dient dazu, weitere Beispiele aus den unklassifizierten Daten U auszuwählen. Nachdem eine feste Anzahl von Beispielen zum Klassifizieren ausgewählt worden sind, wird ein neuer Lerner f' trainiert.

Für verschiedene Lernalgorithmen existieren bereits Verfahren zum aktiven Lernen. So wird in der Arbeit von [11] ein Ansatz zum aktiven Lernen mit Neuronalen Netzen vorgestellt. Sehr populär sind auch Ansätze zum aktiven Lernen mit Support Vector Machines (SVM). Der teure Trainingsprozess der SVM wird durch das Training mit nur ausgewählten Beispielen reduziert. In der Arbeit von [12] werden Beispiele ausgewählt, die nahe an der Hyperebene liegen. Im Bereich des *Image Retrieval* [13] [14] oder speziell im Bereich von bildgestützter Arzneimittelforschung [15] sind solche Ansätze oft vertreten.

Um gleichzeitig auch die Verteilung der Daten beim aktiven Lernen zu berücksichtigen, haben wir einen Ansatz entwickelt, der die Daten zunächst clustert. Arbeiten, die das aktive Lernen und Clustering miteinander verbinden sind bisher noch selten.

In der Arbeit von [16] werden die Daten geclustert, indem der Datensatz zunächst mit einem *Farthest-First-Traversal* exploriert wird. Anschließend stabilisieren zufällig gewählte Beispiele die Klassifikation.

In der Arbeit von [17] wird ein aktives halb-überwachtes Clustering durchgeführt, indem ein Kostenfaktor in die Zielfunktion des Fuzzy c -means Algorithmus eingerechnet wird, der die Klassifikation für ausgewählte Beispiele mit berücksichtigt. Der Cluster mit dem größten Fuzzy-Hypervolumen wird hierbei zur Selektion von neuen Beispielen herangezogen.

Das von uns entwickelte adaptive Klassifikationsschema unterscheidet sich von den bisherigen Arbeiten in der Art, dass wir zu Beginn nicht über einen klassifizierten Trainingsdatensatz verfügen. Daher werden die Daten zuerst geclustert, bevor ein Benutzer die Klassifikationsgenauigkeit durch Überwachung verbessert.

4 Adaptive Klassifikation

In diesem Kapitel stellen wir das adaptive aktive Klassifikationsschema vor, welches zur Analyse der Zellbilder eingesetzt wird. Wir beginnen mit einem initialen Fuzzy c -means Clustering mit Erkennung von Rauschen in Abschnitt 4.1, danach wiederholen wir die LVQ-Technik in Abschnitt 4.2 um die initiale Klassifikation anzupassen. Eine aktive Strategie, um neue Beispiele für den LVQ-Algorithmus auszuwählen wird in Abschnitt 4.3 vorgestellt. In Abschnitt 4.4 werden dann alle Einzeltechniken zu einem adaptiven aktiven Klassifikationsschema zusammengefügt.

4.1 Fuzzy c -means mit Erkennung von Rauschen

Der Fuzzy c -means (FCM) Algorithmus [1] ist eine bekannte Technik zur Erkennung der zugrunde liegenden Strukturen in den Daten. Um die Daten gruppieren zu können, ist ein Abstandsmaß notwendig, wobei üblicherweise die euklidische Distanz benutzt wird. Fuzzy Clustering zeichnet sich dadurch aus, dass die Cluster überlappen dürfen. Jeder Datenpunkt besitzt einen Zugehörigkeitsgrad zu jedem Cluster-Prototypen. In unserer Arbeit verwenden wir eine erweiterte Version mit Erkennung von Rauschen in den Daten [2].

Sei $T = \vec{x}_i, i = 1, \dots, m$ eine Menge von Feature-Vektoren und $W = \vec{w}_k, k = 1, \dots, c$ die Menge der c Prototypen. V sei die Matrix mit den Koeffizienten $v_{i,k}$, welche die Zugehörigkeit des Datenpunktes \vec{x}_i zu cluster k bezeichnen. Der fuzzy c -means Algorithmus mit Noise-Erkennung minimiert schrittweise die folgende Zielfunktion:

$$J_m = \sum_{i=1}^{|T|} \sum_{k=1}^c v_{i,k}^m d(\vec{w}_k, \vec{x}_i)^2 + \delta^2 \sum_{i=1}^{|T|} \left(1 - \sum_{k=1}^c v_{i,k}\right)^2 \quad (1)$$

$m \in (1, \infty)$ ist der Fuzzifizierungs-Parameter, der bestimmt in welchem Ausmaß die Cluster überlappen dürfen. Der erste Term entspricht der Zielfunktion im normalen Fuzzy c -means Algorithmus, der zweite Term ergibt sich aus der Noise-Erkennung. Hierbei wird ein weiterer Noise-Cluster eingeführt. δ ist die Distanz, die jeder Datenpunkt zu diesem Noise-Cluster hat. Diese Distanz kann entweder fix sein, oder in jeder Iteration angepasst werden. Die Idee beim Noise-Clustering ist, dass Datenpunkte, welche sich nicht den gegebenen Clustern zuordnen lassen, dem Noise Cluster zugeordnet werden. J_m wird unter folgender Randbedingung minimiert:

$$\forall i : 0 \leq \sum_{k=1}^{c-1} v_{i,k} \leq 1 \quad (2)$$

Der fuzzy c -means Algorithmus wird oft benutzt, wenn keine a-priori Informationen verfügbar sind. Basierend auf den Prototypen kann man einen Datensatz klassifizieren, indem man jedem Prototyp einen Klassennamen zuordnet. Jeder Datenpunkt wird dann dem nächstgelegenen Prototyp zugeordnet und bekommt dessen Name. Diese Kategorisierung ist meistens nicht sehr genau, da die Struktur des Clusterings nicht unbedingt der Klassenverteilung im Datensatz entsprechen muss. Daher möchte man das Clustering bzw. die Prototypen anpassen, um eine bessere Klassifikation zu erreichen. Da der FCM diese Möglichkeit nicht direkt bietet, benützen wir den LVQ-Algorithmus um unsere Prototypen zu verschieben. Im nächsten Abschnitt wird diese Technik vorgestellt.

4.2 Lernende Vektor Quantisierung

Lernende Vektor Quantisierung (LVQ) [3] ist eine so genannte konkurrierende Lernmethode. Eine genaue Beschreibung des Algorithmus findet sich in Algorithmus 1. Der Algorithmus arbeitet folgendermaßen: Für jedes Trainingsbeispiel wird der nächstgelegene Prototyp bestimmt und dann aktualisiert. Besitzen der Prototyp und das Trainingsbeispiel den gleichen Klassennamen, wird der Prototyp ein kleines Stück in Richtung des Beispiels bewegt. Falls Prototyp und Beispiel unterschiedliche Namen haben, wird der Prototyp ein kleines Stück vom Beispiel wegbewegt. Dabei wird die Lernrate ϵ in jeder Iteration verringert, so dass sich die Prototypen immer weniger bewegen. Diese Technik ist auch als

- 1: Wähle R initiale Prototypen für jede Klasse:
 $m_1(k), m_2(k), \dots, m_R(k), k = 1, 2, \dots, K$.
- 2: **repeat**
- 3: Wähle zufällig ein Trainingsbeispiel \vec{x}_i . Sei $m_j(k)$ der nächste Prototyp zu \vec{x}_i . Sei g_i der Klassenname von \vec{x}_i und g_j der Klassenname vom Prototyp.
- 4: **if** $g_i = g_j$ **then** // gleiche Klasse
- 5: Bewege den Prototyp in Richtung des Trainingsbeispiels:
 $m_j(k) \leftarrow m_j(k) + \epsilon(\vec{x}_i - m_j(k))$, wobei ϵ die Lernrate ist.
- 6: **end if**
- 7: **if** $g_i \neq g_j$ **then** // verschiedene Klassen
- 8: Bewege den Prototyp in entgegengesetzter Richtung des Trainingsbeispiels:
 $m_j(k) \leftarrow m_j(k) - \epsilon(\vec{x}_i - m_j(k))$
- 9: **end if**
- 10: Verringere die Lernrate ϵ
- 11: **until** Maximale Anzahl Iterationen erreicht oder keine signifikante Änderung der Prototypen.

Algorithmus 1: LVQ

Simulated Annealing [4] bekannt. Eine wichtige Anforderung im LVQ-Algorithmus ist die Verfügbarkeit des Klassennamens für jedes zufällig gewählte Trainingsbeispiel. In unserer Applikation hingegen sind alle Trainingsdaten ohne Klassennamen. Jedoch steht uns ein Experte zur Verfügung, der für wenige ausgewählte Beispiele die Klasse bestimmen kann. Da der Experte nur eine beschränkte Kapazität hat, benötigen wir eine Strategie, um möglichst sinnvolle Beispiele in den Daten zu finden. Auf diese Strategie wird im nächsten Abschnitt genauer eingegangen.

4.3 Auswahl von Trainingsbeispielen

Die Auswahl der Trainingsbeispiele spielt eine besondere Rolle beim LVQ-Algorithmus. Diese Beispiele stellen das Modell der Daten für den Algorithmus dar, und müssen deshalb sorgfältig ausgewählt werden. Dabei müssen zwei Aspekte betrachtet werden: 1. Exploration der Daten, d.h. Verifikation der bestehenden Hypothese und 2. Ausnutzung (Exploitation) der Daten, in dem man bestehende Entscheidungsgrenzen verfeinert. In unserer Anwendung, in der die Klassennamen von einem Experten zugewiesen werden, ist es sehr wichtig, dass wir mit wenigen Beispielen soviel Information wie möglich gewinnen. Die Idee hinter der Selektionsstrategie ist es, die Verteilung der Daten zu berücksichtigen. Dies wurde auch bereits in der Arbeit von [6] festgestellt. Dort wird vorgeschlagen, den erwarteten Fehler des Lernalgorithmus zu minimieren:

$$\int_x E [(\hat{y}(x; D) - y(x))^2 | x] P(x) dx \quad (3)$$

E_T bezeichnet die Erwartung über $P(y|x)$ und $\hat{y}(x; D)$ ist die Ausgabe des Lernalgorithmus bei Eingabe x aus der Trainingsmenge D . Indem wir die vom Fuzzy c -means gefundene Cluster als Modellierung für die Verteilung der Daten $P(x)$ ansehen, können wir weitere "interessante" Beispiele finden, indem wir uns Datenpunkte anschauen, die zwischen den Clustern liegen.

Wir verfolgen also zwei Ziele: Verifikation der Cluster (Exploration) und die Anpassung

von Klassifikationsgrenzen. Beide werden in den folgenden Abschnitten genauer erläutert.

4.3.1 Exploration

Um allgemeine Informationen über den Klassennamen des Clusters zu erhalten - welcher unsere gegenwärtige Hypothese darstellt - verwenden wir die *Cluster Mean Selection*-Technik von [7]. Hierbei wird ein Cluster erneut geclustert und die nächsten Nachbarn der Prototypen dieser Subcluster werden dann klassifiziert. Wir führen diese Technik rekursiv aus solange die Namen innerhalb eines Clusters inhomogen sind, wird der Cluster erneut aufgespalten. Dies wird solange rekursiv durchgeführt, bis die Namen in einem Cluster gleich sind oder eine gegebene Rekursionstiefe erreicht wird.

4.3.2 Anpassung der Klassifikation

Von Datenpunkten, die zwischen zwei Clustern unterschiedlicher Klasse liegen wird der größte Informationsgewinn erwartet. Dies erfolgt in Übereinstimmung mit den Ergebnissen in [7] und [8].

An dieser Stelle wollen wir ein Verfahren vorstellen, das geeignet ist, Datenpunkte zwischen zwei Clustern zu finden und welches sich besonders einfach beim Fuzzy Clustering einsetzen lässt. Wir konzentrieren uns darauf, nicht nur ein Beispiel pro Iteration zu finden, sondern gleich eine Menge von N Trainingsbeispielen auszuwählen.

Bei der Definition der Zielfunktion des Fuzzy c -means Algorithmus bezeichnet $v_{i,k}$ die Zugehörigkeit von \vec{x}_i zum Cluster k . Wenn wir also nach Datenpunkten zwischen zwei Clustern suchen, so müssen diese eine fast gleiche Zugehörigkeit zu zwei Clustern haben. Wir berechnen für jeden Datenpunkt seinen Rang:

$$\text{Rang}(\vec{x}_i) = 1 - (\min |v_{i,k} - v_{i,l}|) \quad \forall k, l = 1, \dots, c \quad k \neq l \quad (4)$$

Voraussetzung für die Berechnung des Rangs zwischen zwei Clustern ist, dass die Cluster unterschiedlichen Klassen zugeordnet sind. Bei der Rangordnung alleine besteht die Gefahr, dass nur Beispiele ausgewählt werden, die sehr ähnlich zueinander sind. Um dieses Problem zu umgehen führen wir zusätzlich noch eine Selektion aus, die uns eine gewisse Vielfalt gewährleistet. Hierzu selektieren wir eine Menge mit hohem Rang und wählen unter diesen Beispielen eine Anzahl von möglichst unterschiedlichen Beispielen. Hierbei sei auf das *Farthest-First-Traversal*-Verfahren von [9] verwiesen. Hierbei werden die unterschiedlichsten Beispiele selektiert, indem das erste Beispiel zufällig gewählt wird und jedes weitere Beispiel das am weitesten entfernte von der Menge der bisher selektierten Beispiele ist. Die dabei verwendete *min-max-Distanz* von einem Datenpunkt x zur Menge S ist definiert als $d(S, x) = \min_{y \in S} d(x, y)$.

4.4 Adaptive aktive Klassifikation

In diesem Abschnitt fassen wir die bisher erwähnten Techniken zu einem adaptiven aktiven Klassifikationsschema zusammen. Die einzelnen Schritte sind in Algorithmus 2

zusammengefasst. Den Ausgangspunkt für eine initiale Klassifikation bildet das FCM Clustering. Wir unterstellen den Strukturen in den Daten eine gewisse Semantik für die Klassifikation, d.h. Beispiele, die im Datenraum nahe beieinander liegen, sind auch sehr wahrscheinlich in derselben Klasse. Das Fuzzy Clustering fasst die Daten zusammen und verhindert so unnötige Abfragen.

Insbesondere die Variante mit der Erkennung von Rauschen hilft dabei, seltene Beispiele zu erkennen und herauszufiltern, die für die Gesamtklassifikation keine Relevanz besitzen. Dennoch kann man so dem biologischen Experten seltene und damit potentiell interessante Beispiele anzeigen.

Die erste Phase umfasst die Klassifikation der initialen Cluster-Prototypen. Der Datensatz wird geclustert, und die Prototypen zur Klassifikation angezeigt. Anschliessend wird überprüft, ob die Klassennamen unterschiedlich sind. Solange die Verteilung der Klassennamen im aktuellen Cluster nicht homogen ist, wird der Cluster erneut aufgespaltet. Dabei kann der Cluster analog zur Tiefensuche nach der Evaluation gleich aufgeteilt werden, oder er wird analog zur Breitensuche erst nach Betrachtung aller aktuellen Cluster aufgespaltet. Dies wird solange durchgeführt, bis eine gewünschte Rekursionstiefe erreicht ist. Dabei ist jedoch zu beachten, dass die Anzahl der Beispiele pro Ebene exponentiell steigen kann, und diese Phase nur zur Initialisierung dient.

Nachdem wir unsere initialen Prototypen erhalten haben, versuchen wir die Klassifikation zu verbessern, indem wir die Prototypen analog zum LVQ-Verfahren weiter verschieben. Dazu benötigen wir weitere Trainingsbeispiele, die nach dem in Abschnitt 4.3 dargestellten Verfahren selektiert werden. Alle selektierten und mit einem Klassennamen versehenen Beispiele stehen dann dem LVQ-Algorithmus zum Trainieren zur Verfügung. Der Vorgang kann beliebig oft wiederholt werden, wobei der Benutzer ihn abbrechen kann, wenn eine akzeptable Klassifikationsgüte erreicht worden ist. Im Folgenden wollen wir noch verschiedene Kriterien zur Evaluation dieser Klassifikationsgüte vorstellen:

Clustergütemaße geben uns Informationen über die Qualität des Clusterings. Wir setzen hier Gütemasse für die Variation innerhalb der Cluster und zwischen den Clustern ein [10]. Diese Indikatoren erweisen sich als nützlich, um zu Beginn die richtigen Attribute im Datensatz zu wählen, so dass die Struktur der Daten die Semantik der Klassifikation widerspiegelt. Mit zunehmender Anzahl von Klassifikationsschritten nimmt die Bedeutung dieser Deskriptoren naturgemäß leicht ab.

Gradient Die im bisherigen Klassifikationsprozess erhaltenen klassifizierten Beispiele können zum Testen verwendet werden. Nachdem die Klassen für die Beispiele in und zwischen den Clustern bestimmt worden sind, werden die Prototypen angepasst. Dadurch ergibt sich eine neue Klassifikation der Daten. Das Verhältnis von korrekt klassifizierten Beispielen (auf Basis der bisher klassifizierten Beispiele) zu falsch klassifizierten Beispielen kann als Indikator verwendet werden.

Cluster-Visualisierung Speziell in unserem Anwendungsfall, bei dem die numerischen Feature-Vektoren mit Bildern von Zellen assoziiert sind, können wir diese Bilder dem biologischen Experten präsentieren. Wir selektieren hierfür eine feste Anzahl von Zellbildern aus jedem Cluster. Eine Entscheidung darüber, ob die Klassifikationsgenauigkeit ausreichend ist oder nicht, kann dann vom Experten basierend auf diesen Bildern gefällt werden.

```

1:  $L \leftarrow 0$ 
2: while Unterschiedliche Klassen im aktuellen Cluster do
3:   Teile den Cluster in  $l$  SubCluster mit dem Fuzzy  $c$ -means Algorithmus auf.
4:   Frage nach den Klassennamen der Cluster Prototypen.
5:   Füge die klassifzierten Prototypen zur Menge der Trainingsbeispiele  $L$  hinzu.
6: end while
7: while Lernprozess durch Evaluation nicht beendet do
8:    $T \leftarrow$  Wähle  $m$  Trainingsbeispiele an den Cluster Grenzen.
9:   Wähle  $n$  voneinander verschiedene Beispiele aus  $T$ , füge diese zu den Trainings-
   daten  $L$  hinzu.
10:  Unterteile jeden Cluster in Subcluster, wähle  $n$  Beispiele aus mit Hilfe der Subclu-
   ster Prototypen.
11:  Frage nach den Klassennamen für diese Beispiele und füge sie zur Trainingsmenge
    $L$  hinzu.
12:  Trainiere die Prototypen mit Hilfe der Trainingsdaten  $L$ .
13:  Erniedrige die Lernrate  $\epsilon$ .
14: end while

```

Algorithmus 2: Adaptive Klassifikation

5 Ergebnisse

Da die von uns verwendeten Daten aus den Zell-Assays vertraulich sind, werden die Ergebnisse auf einem frei erhältlichen Datensatz überprüft. Dabei handelt es sich um einen Zellbilddatensatz vom NISIS pap-smear Wettbewerb [20]. Die Aufgabe hierbei ist es, verschiedene Vorstufen von Gebärmutterhalskrebs zu klassifizieren, bevor sich diese zu einem invasiven Karzinom entwickeln. Der Datensatz besteht aus 917 Bildern, welche von medizinisch-technischen Assistenten und Doktoren klassifiziert wurden. Jedes Bild wird mit 20 numerischen Attributen beschrieben, insgesamt existieren sieben verschiedene Klassen. Eine grundlegende Datenanalyse gibt einen Überblick und Informationen über untere Schranken für die Klassifikationsgenauigkeit mit einem linearen Klassifizierer.

Zum Vergleich haben wir den gleichen Datensatz auch mit einem anderen Verfahren klassifiziert: Aktives Lernen mit Support Vektor Machines [12], erwähnt in Kapitel 3. Abbildung 2 zeigt die Fehlerrate von 5 Testläufen mit unserem adaptiven Klassifikationsverfahren. Zu Beginn sinkt die Fehlerrate durch die Exploration sehr stark; sobald alle Klassen gefunden worden sind, sinkt die Fehlerrate durch die Anpassung der Prototypen noch um weitere 10 Prozent.

Der Ansatz von [12] kennt keine Explorationsphase. Hier geht man so vor, dass Beispiele zufällig gezogen werden, bis eine feste Anzahl von Beispielen für jede Klasse existieren. Dies erklärt die starken Schwankungen beim Klassifikationsfehler zu Beginn. Mit zunehmender Anzahl von Iterationen stabilisiert sich der Fehler. Nach ca. 200 bis 300 Trainingsbeispielen ist der Fehler der Support Vector Machine um ca. 5% niedriger als bei unserem Ansatz.

Zusammenfassend lässt sich sagen, dass die Performanz unseres adaptiven Klassifikationsschemas leicht unter dem Niveau einer Support Vector Machine liegt. Der Vorteil unseres Verfahrens besteht jedoch in einer stabileren und besseren Klassifikation zu Beginn. Je nach Auswahl eines Kernels für die SVM kann das Training sehr lange dauern. Der Trainingsprozess in unserem Verfahren ist deutlich schneller: nach 200 Beispielen

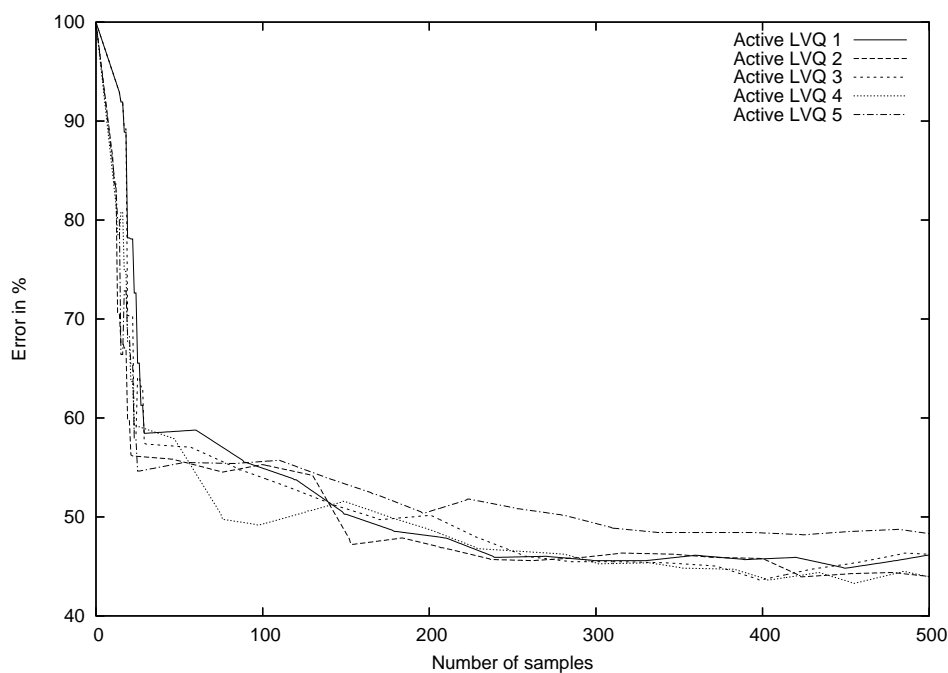


Bild 2: Active LVQ

hat unser Verfahren ca. 8,2 Sekunden reine Trainingszeit benötigt, die SVM dagegen 175 Sekunden.

6 Zusammenfassung und Ausblick

In dieser Arbeit haben wir uns mit dem Problem beschäftigt, eine große Anzahl von Daten unter der Randbedingung zu klassifizieren, dass wir zu Beginn über keine klassifizierten Trainingsbeispiele verfügen.

Zu diesem Zweck haben wir ein neues Klassifikationsschema entwickelt, welches den Datensatz durch ein initiales Clustering mit Erkennung von Rauschen kompakt zusammenfasst. Eine initiale Klassifikation wird dadurch gewonnen, indem man jeden Cluster in seine Subcluster unterteilt, und dann nach den Namen für die Prototypen fragt. Sobald die Klassen innerhalb eines Clusters homogen sind, stoppt der Prozess im aktuellen Cluster. Jeder Datenpunkt bekommt die Klasse des Prototypen, der die geringste Distanz zu ihm aufweist. Basierend auf dieser initialen Klassifikation wird dann versucht, das durch die Prototypen erhaltene Konzept noch näher an die Semantik des Benutzers anzupassen. Hierfür werden die Prototypen mit Hilfe des LVQ-Verfahrens trainiert, basierend auf wenigen klassifizierten Trainingsbeispielen. Die hier vorgestellte Strategie, die möglichst sinnvolle Beispiele aus der Menge der nicht klassifizierten Daten auswählt ermöglicht eine schnellere Adaption als sie z. B. bei einer Zufallsauswahl möglich wäre.

An einem konkreten Anwendungsbeispiel in der Biologie haben wir den Nutzen solch eines Klassifikationsschemas aufgezeigt und mit Daten aus dem Bereich der Zellanalyse dessen Leistungsfähigkeit demonstriert.

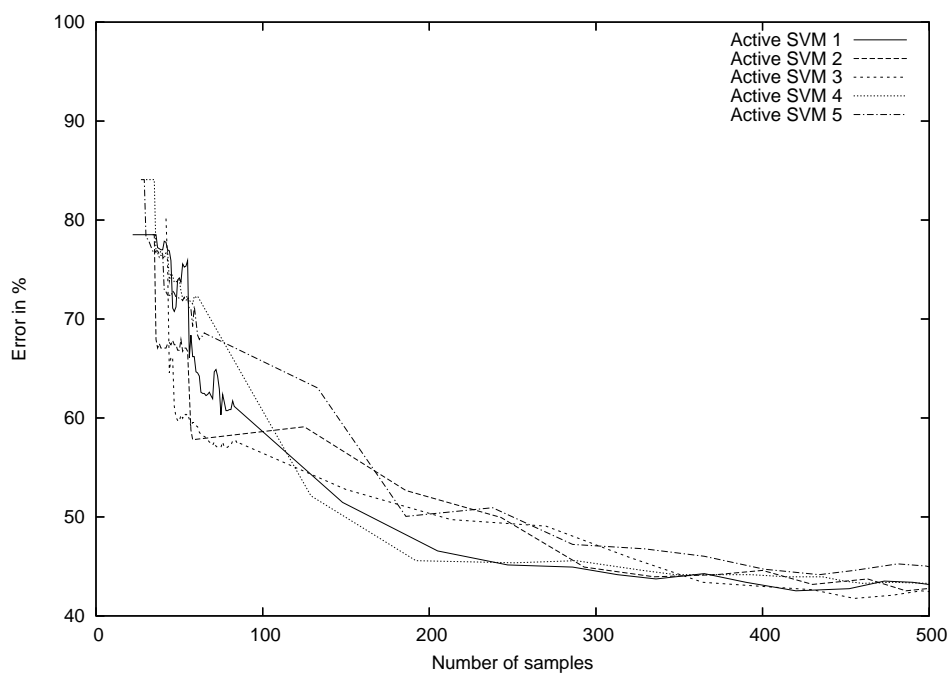


Bild 3: Active SVM

Literatur

- [1] Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press. 1981.
- [2] Dave, R.N.: *Characterization and detection of noise in clustering*. New York: Elsevier Science Inc. 1991.
- [3] Kohonen, T.: *Self-Organizing Maps*. Heidelberg: Springer Verlag 1995.
- [4] Kirkpatrick, S. and Gelatt, C.D. Jr. and Vecchi, M.P.: *Optimization by Simulated Annealing*. Science 1983.
- [5] Jones, T. and Carpenter, A. and Golland, P.: *Voronoi-Based Segmentation of Cells on Image Manifolds*. Proc. Computer Vision for Biomedical Image Applications, Beijing, China, pp. 535-543 2005.
- [6] D. Cohn and Z. Ghahramani and M. Jordan: *Active Learning with Statistical Models*. Cambridge, MA: MIT Press 1995.
- [7] B. Gabrys and L. Petrakieva: *Combining labelled and unlabelled data in the design of pattern classification systems*. New York : Elsevier 2004.
- [8] H.T. Nguyen and A. Smeulders: *Active Learning Using Pre-Clustering*. New York : ACM Press 2004.
- [9] Hochbaum and Shmoys: *A best possible heuristic for the k-center problem*. Mathematics of Operations Research, pp. 180-184. 1985.
- [10] M.P. Windham: *Cluster validity for fuzzy clustering algorithms*. Fuzzy Sets and Systems 5, pp. 177-185 1981.

- [11] D. A. Cohn and L. Atlas and R. E. Ladner: *Improving Generalization with Active Learning*. Machine Learning Vol. 15 Nr. 2, pp. 201-221 1994.
- [12] G. Schohn and D. Cohn: *Less is more: Active learning with support vector machines*. Proc. 17th International Conference on Machine Learning, pp. 839-846 2000.
- [13] T. Luo et. al.: *Active Learning to Recognize Multiple Types of Plankton*. Journal of Machine Learning Research, pp. 589-613 2005.
- [14] L. Wang and K. L. Chan and Z. h. Zhang: *Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval*. Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 629-634 2003.
- [15] M. K. Warmuth et. al.: *Support vector machines for active learning in the drug discovery process*. Journal of Chemical Information Sciences, pp. 667-673 2003.
- [16] S. Basu and A. Banerjee and R. J. Mooney: *Active Semi-Supervision for Pairwise Constrained Clustering*. Proc. SIAM International Conference on Data Mining 2004.
- [17] N. Grira and M. Crucianu and N. Boujemaa: *Active Semi-Supervised Clustering for Image Database Categorization*. Proc. Conference Content-Based Multimedia Indexing, Riga, Latvia 2005.
- [18] R.M. Haralick and K. Shanmugam and I. Dinstein: *Textural Features for Image Classification*. IEEE Transactions on systems, man and cybernetics 1973.
- [19] F. Zernike: *Diffraction theory of the cut procedure and its improved form, the phase contrast method*. Physica 1934.
- [20] J.Jantzen et. al.: *Pap-smear benchmark data for pattern classification*. <http://fuzzy.iau.dtu.dk/downloads/smear2005/> 2005.

Analyse von Zeitreihen in der Medizin: Informationsgehalt, Klassifikation und Unsicherheit

Ole Burmeister

Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik,
D-76021 Karlsruhe, Postfach 3640, Telefon: (07247) 82-5749, Fax: (07247) 82-5786,
E-Mail: ole.burmeister@iai.fzk.de

1 Einleitung

Durch den Trend zur elektronischen Datenerfassung übersteigen die erfassten Datenmengen die Fähigkeiten der menschlichen Nutzer, Wissen aus diesen Daten zu extrahieren [1]. Eine (teil-)automatisierte Analyse der Daten ist notwendig und erlaubt eine objektive Auswertung auch sehr großer Datenmengen („Data-Mining“). Aufgaben im Data-Mining sind z.B. die Detektion von Abhängigkeiten, die Identifikation und Beschreibung von Klassen, sowie die Detektion von Ausreißern [2]. Ziele des Einsatzes von Data-Mining Systemen sind nicht nur Systeme zur Entscheidungsunterstützung, sondern auch der Wissensgewinn [1].

Data-Mining-Anwendungen im Bereich der Medizin müssen ihre Informationen häufig aus Zeitreihen extrahieren und haben große Anforderungen bezüglich der Sicherheit von Entscheidungen der Klassifikatoren. Vorhandene Vorbehalte seitens der Mediziner [3] können durch Nachvollziehbarkeit und Sicherheit von Klassifikationsentscheidungen abgebaut werden.

In [4] werden zwei Varianten der Zeitreihenklassifikation eingeführt:

- Typ 1: Die Zeitreihe gehört zu genau einer Klasse: $\mathbf{Z} \rightarrow \hat{y}$
- Typ 2: Jeder Abtastpunkt einer Zeitreihe gehört zu einer Klasse, somit gehört die Zeitreihe zu einer Sequenz von Klassen: $\mathbf{Z}[k] \rightarrow \hat{y}[k]$

Für Typ 1 erfolgt die Klassifikation häufig nach einem Schema, das der Klassifikation von Einzelmerkmalen entspricht (Bild 1).

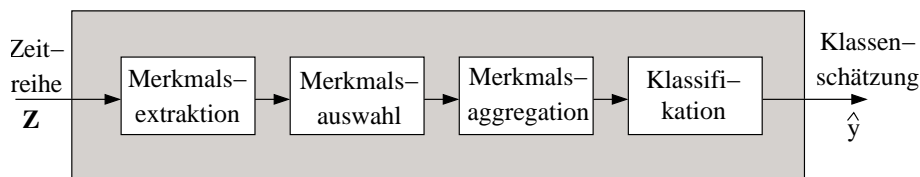


Bild 1: Klassifikationsschema – Nach [5, 6]

Abtastpunkte innerhalb eines definierten Intervalls werden zu einzelnen Merkmalen verrechnet und für die Klassifikation verwendet (siehe z.B. [7]). Bei Problemen nach Typ 2 enthalten die Zeitreihen tendenziell zeitvariante Unterschiede, die von Klassifikatoren erkannt werden müssen. Dass Muster, die eine Klasse definieren, im Laufe einer Zeitreihe variieren können, wurde z.B. in [8] implizit vorgestellt. In [9, 10] wurden die Probleme formalisiert und unterschiedliche Lösungsansätze in Form zeitvarianter Klassifikatoren verglichen. Diese Arbeit erweitert die Anwendung der zeitvarianten Klassifikatoren auf

die Bestimmung und Visualisierung des Informationsgehalts und der Betrachtung der Unsicherheit von Klassifikationsentscheidungen.

Die Bestimmung der Unsicherheit von Klassifikationsentscheidungen wird bereits seit langer Zeit durchgeführt (siehe z.B. [11]), um Entscheidungssysteme zu entwerfen [12, 13]. Diese Arbeit präsentiert Methoden, die sowohl auf herkömmliche Probleme als auch auf die kontinuierliche Klassifikation von Zeitreihen anwendbar sind. Als Beispiele werden eine kontinuierliche Klassifikation von Zeitreihen und ein neuer Ansatz für ein System zur Entscheidungsunterstützung in der Therapieplanung mit Hilfe der instrumentellen Ganganalyse verwendet.

Im folgenden Kapitel 2 werden die Datensätze der zwei Anwendungsbeispiele vorgestellt. Anschließend wird in Kapitel 3 auf die Bestimmung und Visualisierung des Informationsgehalts der Zeitreihen und in Kapitel 4 auf die Unsicherheit von Klassifikationsentscheidungen eingegangen. Die Arbeit schließt mit einer Zusammenfassung in Kapitel 5.

2 Anwendungsbeispiele

2.1 Brain Machine Interface (BMI)

Eine Anwendung der zeitvarianten Klassifikatoren auf Brain Machine Interfaces wurde bereits in [9, 10] vorgestellt. Daher wird hier auf eine Definition und Einleitung verzichtet.

Es wird ein Datensatz verwendet, der eine ECoG-Aufzeichnung [14] (ECoG: Elektrokortikogramm) aus der internationalen BCI Competition III¹ enthält. Ein ECoG ist eine invasive Aufzeichnungstechnik, bei der die Elektroden nicht auf der Schädel-, sondern der Hirnoberfläche aufgelegt werden. Für diesen Datensatz wurde ein 8x8-Gitter aus Platinelektroden auf dem rechten motorischen Kortex platziert. Die 64 Kanäle wurden mit einer Abtastrate von 1000 Hz aufgezeichnet und zur weiteren Verarbeitung auf 250 Hz reduziert. Die Aufgabe des Probanden war, sich zwei verschiedene Bewegungen vorzustellen (Zunge oder linker kleiner Finger). Vor jedem Versuch wurde dem Probanden per visuellem Hinweis die Bewegung vorgegeben. Die Aufnahme der Daten wurde jeweils 0.5 Sekunden nach dem visuellen Hinweis gestartet, um Artefakte durch Augenbewegungen zu vermeiden. Die Aufnahmezeit betrug pro Versuch 3 Sekunden. Der Lerndatensatz besteht aus $N_{Lern} = 278$ Versuchen, im Testdatensatz sind $N_{Test} = 100$ Versuche enthalten.

Zwischen der Aufnahme der Lerndaten und der Testdaten lag ein Zeitraum von einer Woche. Eine derart lange Zeit zwischen zwei Aufnahmesitzungen erhöht die Gefahr, dass der Proband keine reproduzierbaren Zeitreihen generieren kann. Wie alle Klassifikatoren haben auch die zeitvarianten Klassifikatoren Schwierigkeiten, wenn sich die generierten Muster in den Lern- und Testdaten deutlich unterscheiden. Häufig sind die Probleme durch Änderungen der Amplituden in den generierten Mustern gekennzeichnet, zum Teil auch durch zeitliche Verschiebungen.

Aus den Originalzeitreihen wurden weitere Zeitreihen extrahiert, durch die das Problem der zeitlich getrennten Aufnahmen abgeschwächt wird. Neben einer Referenzierung der Originalzeitreihen [15] wurden die Zeitreihen in einem breiten Frequenzband mittels

¹Daten unter http://ida.first.fhg.de/projects/bci/competition_iii/

Butterworth-Filtern 5. Ordnung gefiltert und anschließend so genannte „Common Spatial Patterns“ (CSP) berechnet [15]. Eine Transformation der originalen Zeitreihen mit Hilfe der CSP führt zu Varianzen in den Zeitreihen, die eine möglichst optimale Trennung der Klassen ermöglicht. Die Berechnung dieser Varianzen in Fenstern einer definierten Breite führt zu neuen Zeitreihen.

Die Online-Klassifikation von Datensätzen für die Steuerung von BMIs entspricht dem Typ 2, haben allerdings längere Phasen mit gleicher Klassenzugehörigkeit. Dieser Datensatz liegt bereits in Form von einzelnen Versuchen für die Offline-Klassifikation vor und muss daher zum Typ 1 gezählt werden.

2.2 Instrumentelle Ganganalyse

Die instrumentelle Ganganalyse ist ein Verfahren zur Untersuchung des menschlichen Gangs [16]. Für die Anwendung der Methoden lag ein Datensatz der Orthopädischen Universitätsklinik Heidelberg vor. Es werden die Daten beider Beine von 108 Patienten (216 Datentupel) mit Infantiler Zerebralparese (ICP) vor einem chirurgischen Multi-Level-Eingriff verwendet. Die Zeitreihen enthalten Gelenkwinkelverläufe verschiedener Gelenke (Knie, Fußgelenk, Becken, Hüfte). Als Einzelmerkmale liegen Daten zu Patienteninformationen (Alter zum Zeitpunkt der Messung, Gewicht, ...) und so genannte Raum-Zeit-Parameter (Schrittlänge, Gehgeschwindigkeit, ...) vor. Weiterhin sind in diesem Datensatz die durchgeführten Eingriffe jedes Patienten enthalten. Die Eingriffe wurden nach Ort, Art des Gewebes und einer Nummer für die OP-Technik codiert (z.B. KFe9: Femur suprakondyläre Derotations-Verkürzungs-Osteotomie). Da einige der im Datensatz codierten Operationen nur bei einer sehr kleinen Anzahl an Patienten durchgeführt worden sind und z.T. ähnliche Voraussetzungen und Wirkungen haben, wurden die chirurgischen Eingriffe nur nach Gewebeart und Ort des Eingriffes kategorisiert, die eigentliche Technik wurde vernachlässigt. Dieses Vorgehen ist konsistent mit der klinischen Praxis, bei der die spezielle OP-Technik erst unmittelbar vor dem Eingriff festgelegt wird.

Der Datensatz erlaubt verschiedene Klassifikationsprobleme, u.a.: die Trennung von Patienten und Normprobanden sowie die Entscheidungsunterstützung in der Therapieplanung [17]. Dieser Datensatz repräsentiert Klassifikationsprobleme, bei denen die vektorialen Zeitreihen zu einer einzelnen Klasse gehören (Typ 1). Die Generierung von Merkmalen aus Zeitreihen in der instrumentellen Ganganalyse ist ausführlich in [16, 18] beschrieben.

3 Informationsgehalt und dessen Visualisierung

3.1 Einleitung

Die Visualisierung des Informationsgehalts von Zeitreihen kann zum Wissensgewinn der Nutzer beitragen. In medizinischen Anwendungen steht teilweise weniger das Ergebnis von Klassifikatoren im Vordergrund als der Entscheidungsweg oder die verwendeten Merkmale. Zum anderen kann durch die Betrachtung des Informationsgehalts eine Entscheidung für oder gegen einen zeitvarianten Klassifikator getroffen werden.

Der Informationsgehalt eines Merkmals ist ein relativ abstraktes Maß, das angibt, wie gut es zur Lösung eines bestimmten Problems geeignet ist. Die Werte werden meist auf den Bereich $[0, \dots, 1]$ normiert.

Zwei grundsätzlich verschiedene Ansätze sind der *filter* bzw. *wrapper approach* [19, 20]. Im ersten Fall wird mittels einer Bewertungsfunktion die Qualität eines Merkmals zur Lösung eines gegebenen Problems geschätzt, z.B. durch univariate (ANOVA) bzw. multivariate Varianzanalyse (MANOVA) [21] oder Informationstheoretische Maße [22]. Beim wrapper approach wird der komplette Entwurf des Klassifikators (Merkmalsauswahl, Merkmalsaggregation, Parameterbestimmung) durchgeführt. Die Güte des erzeugten Klassifikators entspricht dem Informationsgehalt der verwendeten Merkmale. Dieser Ansatz hat gegenüber dem filter approach den Vorteil, dass der Informationsgehalt direkt über dem später zum Einsatz kommenden Klassifikator bestimmt wurde. Auf der anderen Seite benötigt der wrapper approach deutlich mehr Rechenzeit, vor allem wenn für eine Absicherung der Ergebnisse eine Crossvalidierung durchgeführt wird.

Die Visualisierung des Informationsgehalts stößt schnell an Grenzen in der Darstellbarkeit. Der Informationsgehalt einzelner Merkmale oder Zeitreihen ist relativ einfach zu visualisieren. Die Visualisierung des Informationsgehalts von Merkmalskombinationen ist insbesondere dann schwierig, wenn die zeitliche Dimension berücksichtigt werden muss. Zwei übersichtliche und informative Darstellungen sind:

- Betrachtung des Informationsgehalts einzelner Zeitreihen über der Zeit (siehe z.B. Bild 2 und 3a)
- Relativer Zugewinn zum Informationsgehalt einer gegebenen Vorauswahl von s Zeitreihen und der Hinzunahme einer zusätzlichen Zeitreihe über der Zeit (siehe z.B. Bild 3b):

Die zweite Visualisierung kontrolliert Kombinationen von Merkmalen auf ihren Informationsgehalt. Für den Nutzer ist dabei wichtig, die Alternativen betrachten zu können, anstatt ausschließlich das Endergebnis. Der Informationsgehalt wird, wie bei einem Spektrogramm, als dritte Dimension farblich oder durch Graustufen dargestellt. Durch diese Art der Darstellung sind die zeitlichen Änderungen im Informationsgehalt leicht zu erfassen und verschiedene Merkmale können miteinander verglichen werden.

3.2 Ergebnisse

3.2.1 Brain Machine Interface

In den Abbildungen sind lediglich die Zeitreihen enthalten, die in einem Fenster der Größe 100 Abtastpunkte generiert wurden. Für die Klassifikation ergab sich bei dieser Fensterbreite der beste Kompromiss aus Klassifikationsergebnis und Verzögerung durch die Berechnung (siehe Tabelle 1). Die Daten der Tabelle zeigen die Klassifikationsergebnisse eines K4-Klassifikators [10] über Testdaten, der 6 Merkmale über MANOVA auswählt und einen Bayes-Klassifikator zur Klassifikation verwendet.

Bild 2 enthält den Informationsgehalt, der über die Klassifikationsgüte (links) beziehungsweise ANOVA (rechts) bestimmt wurde. In beiden Fällen wurden ausschließlich die Datentupel der Lerndaten (ohne Crossvalidierung) verwendet. Die Relevanzen der ECoG-Daten zeigen, dass in den Originalzeitreihen $(1, \dots, 64)$, im Gegensatz zu den neu extra-

Fensterbreite	20	30	40	50	60	70	80	90	100
Klassifikationsfehler	16%	14%	11%	9%	10%	9%	8%	10%	7%
Fensterbreite	120	160	200	240	280	320	380	480	580
Klassifikationsfehler	8%	8%	7%	10%	14%	19%	21%	25%	30%

Tabelle 1: Klassifikationsfehler bei verschiedenen Fensterbreiten für die Varianzberechnung der CSP-Merkmale im ECoG-Datensatz

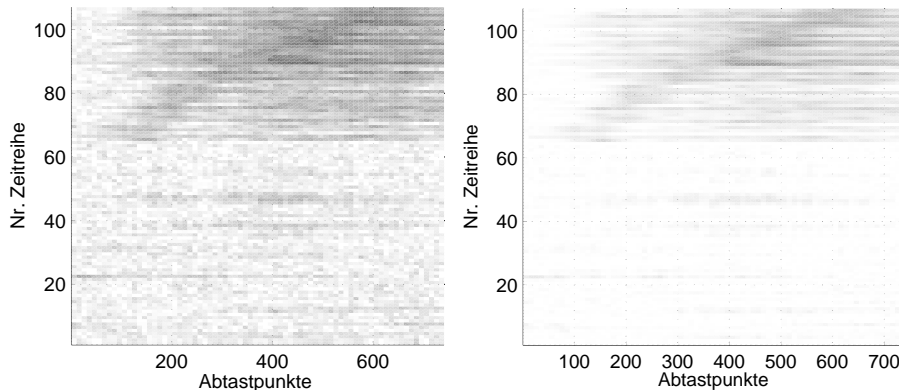


Bild 2: Informationsgehalt über Klassifikationsgüte (links) und ANOVA-Relevanzen (rechts). Weiß: keine Information, Schwarz: eindeutige Trennung möglich.

hierten Zeitreihen, nur wenig Informationen enthalten sind. Ein Entfernen dieser Originalzeitreihen führt allerdings zu einem leicht schlechteren Klassifikationsergebnis (bezogen auf die Ergebnisse in Tabelle 1). Allerdings können sie ebenso durch Verhältniszeitreihen verschiedener Frequenzbänder und Kanäle ersetzt werden.

Beide Bilder enthalten eine markante schräge Linie, die den zeitlichen Verlauf des Informationsgehalts in den extrahierten Zeitreihen widerspiegelt. Auffällig ist auch, dass die ANOVA-Relevanzen deutlich weniger Rauschen enthalten als die Relevanzen, die über den Klassifikationsfehler bestimmt wurden.

3.2.2 Ganganalyse

Der Informationsgehalt in Bild 3 bezieht sich auf die Unterscheidung von Patienten vor ihrer Behandlung und gesunden Probanden. Auch in diesem Datensatz enthalten unterschiedliche Zeitreihen Informationen. Es beginnt mit der 7. Zeitreihe, die die Stellung des Kniegelenks in der Sagittalebene enthält (Beugung/Streckung). Patienten neigen zu einem stark gebeugten Knie in der Standphase (für die einzelnen Phasen siehe z.B. [16]). In einer weiteren Phase sind leichte Informationen in der Zeitreihe des Sprunggelenks zu finden (ZR 14). In der Schwungphase (ab Abtastpunkt 60) dann wieder im Kniegelenk, allerdings in der Geschwindigkeitszeitreihe (ZR 19). Patienten gehen im Gegensatz zu gesunden Menschen mit relativ steifem Knie, so dass sich die Stellung des Kniegelenks nur langsam ändert. Auffällig ist, dass der informationstragende Bereich kurzzeitig unterbrochen ist. Dieser Bereich stellt die Schnittpunkte der Kurven der Patienten und Probanden dar. In diesem Bereich sind zwar die Kurvensteigungen unterschiedlich, die

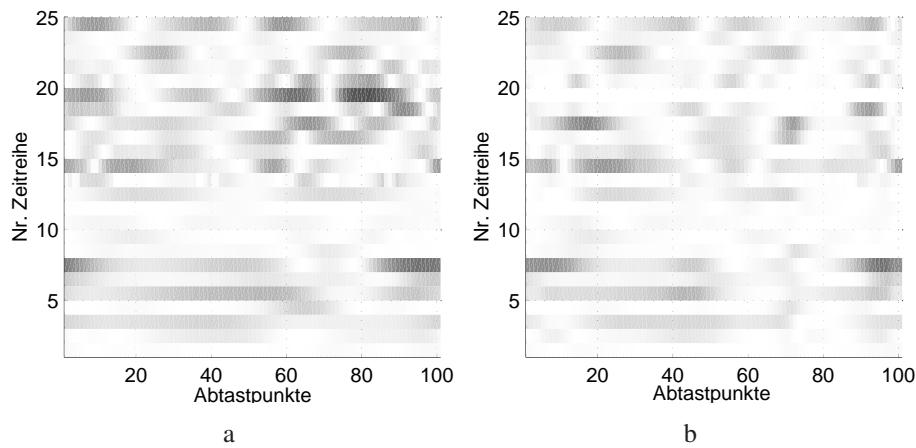


Bild 3: a. ANOVA- und b. MANOVA-Karte des Informationsgehalts (zusätzliche zweite Zeitreihe nach Auswahl von ZR 19) im GA-Datensatz. Weiß: keine Information, Schwarz: eindeutige Trennung möglich.

absoluten Werte jedoch kaum. Diese Lücke im Informationsgehalt kann durch Hinzunahme der zweiten Ableitung des Kniegelenkwinkelverlaufs vermieden werden. Für den Mediziner ist schnell nachvollziehbar, dass die Merkmale, mit denen der Klassifikator arbeitet, seinen eigenen Erwartungen entsprechen. Dies hilft, Vorbehalte gegen computergestützte Systeme abzubauen.

Die zweite Karte in Bild 3b zeigt, dass Zeitreihen, die einzeln betrachtet nur sehr wenige Informationen enthalten, im Zusammenspiel mit weiteren Zeitreihen hilfreich sein können. Die Zeitreihe 19 wurde vorausgewählt und der Informationsgewinn bei Hinzunahme einer Zeitreihe bestimmt. Bei Hinzunahme der Zeitreihe 17 ist ein deutlicher Anstieg des Informationsgehalts, insbesondere innerhalb der ersten Abtastpunkte zu erkennen. Die alleinige Betrachtung der Zeitreihe 17 führt zu einem deutlich schlechteren Ergebnis.

4 Unsicherheit von Klassifikationsentscheidungen

4.1 Einführung

Bei einem idealen Klassifikationsproblem sind die Klassen deutlich voneinander zu trennen (Bild 4a). In der Realität liegt jedoch meist der Fall aus Bild 4b vor. Die Klassen weisen Überlappungen im Merkmalsraum auf, eine fehlerfreie Klassifikation ist nicht möglich. Weiterhin können Datentupel auftreten, die in Bereichen liegen, die nicht durch Lerndaten abgedeckt sind (gekennzeichnet durch Sterne in Bild 4b).

In solchen Fällen von

- widersprüchlichen Informationen (die unbekannte Beobachtung liegt in einem Bereich, in dem mehrere Klassen ähnlich verteilt vorkommen) oder
- zu wenigen Informationen (die unbekannte Beobachtung liegt in einem Bereich, der nicht durch Lerndaten abgedeckt ist)

müssen Klassifikatoren in der Lage sein, Klassifikationsentscheidungen zurückzuweisen.

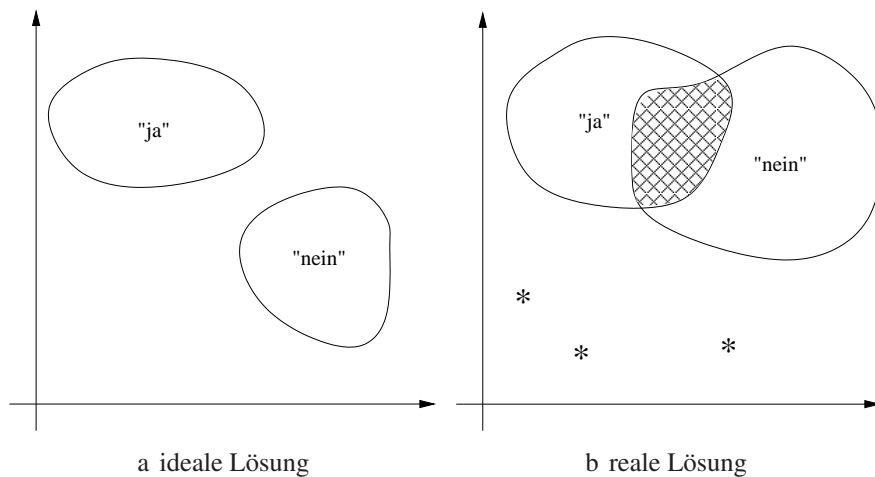


Bild 4: Schema des Problems

Dem Klassifikator wird eine prüfende Instanz angehängt, die eine Entscheidung verhindern kann und dem Nutzer (oder der aufrufenden Funktion) diese Rückweisung mitteilt, z.B. durch $\hat{y}[k] > C$, wobei $\hat{y}[k]$ die Klassenschätzung zum Zeitpunkt k bezeichnet und C die ursprüngliche Anzahl der Klassen im Klassifikationsproblem. Diese Rückweisung kann sowohl bei zeitvarianten als auch bei statischen Klassifikatoren angewendet werden.

Das Ziel eines Klassifikators ist, einen Kompromiss aus Rückweisungen und Fehlentscheidungen festzulegen. Chows Regel besagt, dass nur dann eine Entscheidung für ein unbekanntes Datentupel \mathbf{x} getroffen wird, wenn die maximale a-posteriori Wahrscheinlichkeit einer Klasse oberhalb einer definierten Schwelle T liegt [23]:

$$\hat{y} = \begin{cases} c = \operatorname{argmax}_{c \in \{1, \dots, C\}} P(c|\mathbf{x}), & \text{falls } \max_c P(c|\mathbf{x}) \geq T \\ C + 1 & \text{sonst} \end{cases} \quad (1)$$

Die Schwelle T muss problemspezifisch (z.B. über Kostenfunktionen [13]) gewählt werden.

Der Fall, dass Beobachtungen in einem Bereich liegen, die nicht durch Lerndaten abgedeckt sind, ähnelt der so genannten Ausreißerdetektion. Hierbei ist zwischen Lern- und Testdaten zu unterscheiden:

- Ausreißer in den Lerndaten müssen bereits vor dem Entwurf des Klassifikators ausgeschlossen werden.
- Für Ausreißer in den Testdaten darf der Klassifikator mangels Abdeckung durch Lerndaten keine Entscheidung treffen.

Die Detektion von Ausreißern in den Lerndaten ist ein kompliziertes Verfahren. In [24] werden verschiedene Möglichkeiten zur Definition und Detektion von Ausreißern gegeben, z.B. anhand von Verteilungen, Dichten oder Distanzen. Eine alternative Möglichkeit ist, einen Klassifikator zu entwerfen, der entscheidet, ob ein unbekanntes Datentupel zu der angelernten Klasse gehört oder nicht (one-class Klassifikatoren, ausführliche Übersicht in [25]).

In dieser Arbeit werden drei Verfahren vorgestellt, um die Entscheidung in widersprüchlichen Bereichen (Klasse $C + 1$) oder dünnbesetzten Bereichen (Klasse $C + 2$) zu verweigern:

- Ein Bayes-Klassifikator und ein distanzbasierter Ansatz zur Rückweisung in dünnbesetzten Bereichen (Kapitel 4.2)
- Eine Support-Vektor-Maschine und der Abstand zur Trennebene für die Rückweisung bei widersprüchlichen Informationen sowie ein unüberwachtes² one-class Verfahren zur Rückweisung in dünnbesetzten Bereichen (Kapitel 4.3)
- Ein k-nearest neighbor Klassifikator, der die Häufigkeiten verschiedener Nachbarn für die Rückweisung bei widersprüchlichen Informationen verwendet, und ein dichtebasierter Ansatz zur Rückweisung in dünnbesetzten Bereichen (Kapitel 4.4)

Die Rückweisung wegen zu wenigen Informationen hat eine höhere Priorität als die Rückweisung aufgrund widersprüchlicher Informationen.

4.2 Bayes-Klassifikator

Zur Erkennung widersprüchlicher Informationen ist das Verhältnis der a-posteriori Wahrscheinlichkeiten auszuwerten. Anhand des relativen Anteils

$$p_{rel}(\mathbf{x}|y = c) = \frac{p(\mathbf{x}|y = c)}{\sum_{c=1}^C p(\mathbf{x}|y = c)} \quad (2)$$

kann folgende Entscheidungsregel aufgestellt werden:

$$\hat{y} = \begin{cases} \hat{y} & \text{falls } p_{rel}(\mathbf{x}|y = \hat{y}) > s_u \\ C + 1 & \text{sonst} \end{cases}, \quad (3)$$

wobei \mathbf{x} ein unbekanntes Datentupel bezeichnet, $c \in \{1, \dots, C\}$ eine Klasse, y die tatsächliche Klassenzugehörigkeit, \hat{y} eine Schätzung der Klassenzugehörigkeit und s_u einen frei wählbaren Schwellwert.

Eine Entscheidung wird nur dann getroffen, wenn eine Klasse eine deutlich größere a-posteriori Wahrscheinlichkeit als die restlichen Klassen besitzt.

Für die Rückweisung in dünnbesetzten Bereichen wird der Abstand vom Klassenmittelwert verwendet. Die zugrunde liegende Annahme ist, dass der Klassifikator für einen bestimmten Bereich um seine Lerndaten zuverlässige Informationen enthält. Ist ein Datentupel zu weit entfernt, sind die Informationen dieser Klasse nicht zwingend auf dieses Datentupel anwendbar. Für die Lerndaten liegen bei Annahme einer Normalverteilung die Klassenmittelwerte $\bar{\mathbf{x}}_c$ und die geschätzten Kovarianzmatrizen $\hat{\Sigma}_c$ für jede Klasse $c = 1, \dots, C$ vor.

Eine Beobachtung \mathbf{x} wird zurückgewiesen, falls der Mahalanobis-Abstand zur besten Klasse größer als ein frei wählbarer Schwellwert s_a ist:

$$\hat{y} = \begin{cases} \hat{y} = c & \text{falls } (\bar{\mathbf{x}}_c - \mathbf{x})^T \hat{\Sigma}_c^{-1} (\bar{\mathbf{x}}_c - \mathbf{x}) \leq s_a \\ C + 2 & \text{sonst.} \end{cases} \quad (4)$$

Durch die Betrachtung des Mahalanobis-Abstands für die Rückweisung von Ausreißern wird implizit eine untere Schranke für die a-posteriori Wahrscheinlichkeit des unbekanntem Datentupels eingeführt.

²Die Definition von (un-)überwacht unterscheidet sich bei normaler Klassifikation und one-class Klassifikation. One-class Klassifikatoren sind unüberwacht, wenn ausschließlich Datentupel der zu lernenden Klasse für den Entwurf verwendet werden. Beispiele für Ausreißer sind nicht erforderlich.

4.3 SVM

Eine Support Vektor Maschine erzeugt eine optimale Trennebene zur Trennung zweier Klassen und bestimmt für die Klassifikation einen Abstand x_T von der Trennebene [26]. Ein kleiner Wert für x_T gibt einen geringen Abstand von der Trennebene und somit eine potentiell unsichere Entscheidung an. Die Entscheidungsregel lautet:

$$\hat{y} = \begin{cases} \hat{y} & \text{falls } x_T > s_u \\ C + 1 & \text{sonst} \end{cases} . \quad (5)$$

Bei der Rückweisung in dünnbesetzten Bereichen kann nicht der Abstand x_T verwendet werden, da die SVM den Merkmalsraum auf einen 1D-Merkmalsraum aggregiert und dadurch nur senkrecht auf der Trennebene stehende Ausreißer erkennen kann. In dieser Arbeit wird ein Verfahren aus [27] verwendet, das auf einer linearen Optimierung beruht.

Eine Abbildung

$$f_c(\mathbf{x}) = \sum_{i=1}^{N_L} a_{i,c} K(\mathbf{x}, \mathbf{x}_{i,c}) + b_c \quad (6)$$

beschreibt die Trennfläche zwischen Datentupeln, die zu der Klasse c gehören oder nicht dazugehören. N_L bezeichnet die Anzahl an Lerndaten, $\mathbf{x}_{i,c}$ den i . Lerndatentupel der Klasse c und K eine Kernel-Funktion, wie sie bei Support-Vektor-Maschinen üblich ist. In dieser Arbeit wird der RBF-Kernel verwendet [27].

\mathbf{x} wird als Ausreißer klassifiziert, falls $f(\mathbf{x}) < 0$ gilt („harte Grenzen“ [27]). Mit Hilfe der Lerndaten werden die Koeffizienten $a_{i,c}$ und b_c so berechnet, dass gilt:

$$f(\mathbf{x}_{i,c}) \geq 0, \quad i = 1, \dots, N_L. \quad (7)$$

Da häufig in Lerndaten Ausreißer enthalten sind, können Fehler bei der Klassifikation der Lerndaten zugelassen werden („weiche Grenzen“ [27]). Optimiert wird dann

$$W(a_c, b_c) = \sum_{i=1}^{N_L} \left[\underbrace{\sum_{j=1}^{N_L} a_{j,c} K(\mathbf{x}_{i,c}, \mathbf{x}_{j,c}) + b_c}_{=f(\mathbf{x}_{i,c})} \right] + \lambda \sum_{i=1}^{N_L} \zeta_{i,c} \rightarrow \min, \quad (8)$$

mit den Nebenbedingungen

$$f(\mathbf{x}_{i,c}) = \sum_{j=1}^{N_L} a_{j,c} K(\mathbf{x}_{i,c}, \mathbf{x}_{j,c}) + b_c \geq -\zeta_{i,c}, \quad i = 1, \dots, N_L \quad (9)$$

$$\sum_{j=1}^{N_L} a_{j,c} = 1, \quad \text{mit } a_{j,c} \geq 0. \quad (10)$$

wobei $\zeta_{i,c} \geq 0$ für alle $i = 1, \dots, N_L$. Die Nebenbedingung (10) sorgt dafür, dass keine Skalierungseffekte bewertet werden. Durch die Wahl von λ wird bestimmt, wie viele Lerndaten bei der Optimierung falsch klassifiziert werden dürfen. Die Lösung des Optimierungsproblems erfolgt mit Hilfe der frei verfügbaren lp_solve-Toolbox [28].

Für die Anwendung bei der Rückweisung von Datentupeln wird für jede Klasse $c \in \{1, \dots, C\}$ eine Abbildung f_c gemäß (6) konstruiert. Die Entscheidungsregel lautet dann:

$$\hat{y} = \begin{cases} \hat{y} & \text{falls } \nexists c \mid f_c(\mathbf{x}) < s_a \\ C + 2 & \text{sonst} \end{cases} . \quad (11)$$

4.4 k-nearest neighbor (k-NN)

Im Gegensatz zum üblichen k-nearest neighbor Klassifikator [29] wird der Parameter k dynamisch an das unbekannte Datentupel angepasst (Bezeichnung: $k(\mathbf{x})$). Es werden alle Nachbarn innerhalb eines bestimmten Abstands vom unbekanntem Datentupel in die Entscheidung einbezogen, mindestens jedoch k . Dieses Vorgehen hat den Vorteil, keine Datentupel zu vernachlässigen und wurde bereits in leicht unterschiedlicher Form in anderen Veröffentlichungen angewendet [30].

Erste Ansätze zur Rückweisung von Datentupeln mit k-nearest neighbor Klassifikatoren wurden bereits in den 1970er Jahren veröffentlicht [11]. Eine einfache Form der Rückweisung bei widersprüchlichen Informationen ist die Betrachtung des relativen Anteils der Nachbarn einer Klasse bezogen auf die Gesamtzahl der betrachteten Nachbarn. Eine Rückweisung erfolgt dann, wenn die Anzahl der Nachbarn der Klasse, für die sich der Klassifikator entschieden hat, einen bestimmten relativen Anteil unterschreitet:

$$\hat{y} = \begin{cases} \hat{y} & \text{falls } \frac{|\{\mathbf{x}_i | y_i = \hat{y}, \mathbf{x}_i \in \mathcal{L}_{k\text{-NN}}(\mathbf{x})\}|}{k(\mathbf{x})} > s_u, \\ C + 1 & \text{sonst} \end{cases}, \quad (12)$$

wobei $\mathcal{L}_{k\text{-NN}}(\mathbf{x})$ die $k(\mathbf{x})$ nächsten Nachbarn im Lerndatensatz und $|\cdot|$ die Kardinalität einer Menge bezeichnet.

Die Rückweisung in dünnbesetzten Bereichen geschieht durch Betrachtung der Anzahl an Nachbarn innerhalb einer bestimmten Umgebung β_{knn} um das unbekannte Datentupel. Sind zu wenig Lerndaten in der Umgebung vorhanden, wird keine Klassifikation durchgeführt:

$$\hat{y} = \begin{cases} \hat{y} & \text{falls } |\{\mathbf{x}_i | d(\mathbf{x}_i, \mathbf{x}) \leq \beta_{\text{knn}}, \mathbf{x}_i \in \mathcal{L}_{k\text{-NN}}(\mathbf{x})\}| \geq s_a, \\ C + 2 & \text{sonst} \end{cases}, \quad (13)$$

wobei $d(\cdot, \cdot)$ eine Metrik bezeichnet.

4.5 Ergebnisse

4.5.1 Brain Machine Interfaces

Mit Hilfe der Rückweisung von Klassifikationsentscheidungen wegen widersprüchlicher Informationen kann die Anzahl korrekter Entscheidungen erhöht werden. Dabei kommt das Verfahren des Bayes-Klassifikators zum Einsatz. Nachteil ist, dass für einige Datentupel keine Entscheidung getroffen wird. Eine Rückweisung wegen zu wenigen Informationen wurde nicht verwendet.

Bei der Bewertung des Klassifikationsfehlers dürfen nach eine Rückweisung nur noch diejenigen Datentupel bewertet werden, die klassifiziert wurden. Um die Anzahl der zurückgewiesenen Datentupel betrachten zu können, werden so genannte „accuracy-rejection“-Plots verwendet [31]. Bei diesen Plots wird die Klassifikationsgüte dem relativen Anteil an zurückgewiesenen Klassifikationsentscheidungen gegenübergestellt. In Bild 5a ist das Ergebnis für den ECoG-Datensatz zu sehen. Der Klassifikator wurde über den Lerndaten entworfen, die Validierung erfolgte über Testdaten. Als Zeitpunkte für die Ermittlung der

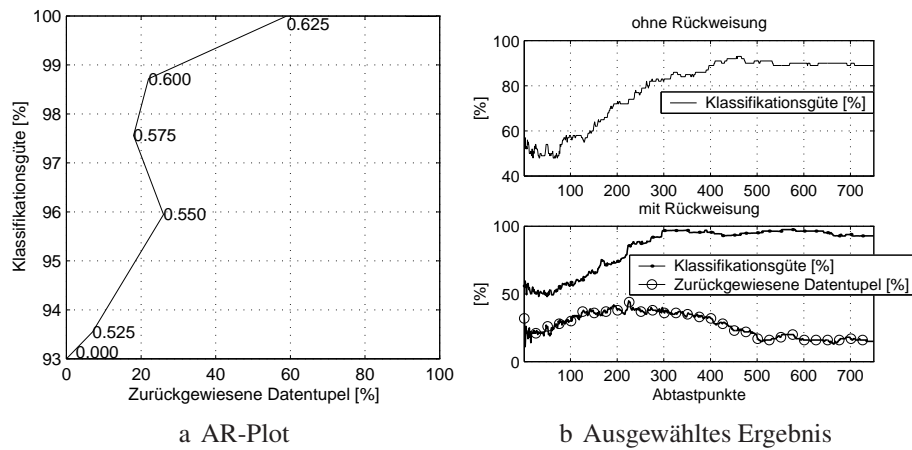


Bild 5: Accuracy-Rejection Plot im ECoG-Datensatz und ausgewähltes Ergebnis für $s_u = 0.575$ vor und nach der Rückweisung (Ergebnis über Testdaten)

Klassifikationsgüte und des relativen Anteils an zurückgewiesenen Datentupeln wurde jeweils der Zeitpunkt des minimalen Klassifikationsfehlers verwendet. Dadurch erklärt sich auch die geringe Abnahme der Rückweisung bei Erhöhung des Schwellwerts von 0.55 auf 0.575. In die Kurve sind die verwendeten Schwellwerte eingetragen. Der erste Schwellwert entspricht der Klassifikation ohne Rückweisung. Hier wird eine Güte von 93% erreicht. Durch Steigerung der Schwellwerte steigt die Anzahl an Datentupeln, die nicht klassifiziert werden. Die Güte der Klassifikation nimmt ebenso zu. Ab einem Schwellwert von $s_u = 0.625$ wird eine Klassifikationsgüte von 100% erreicht, allerdings mit einer erheblichen Anzahl an zurückgewiesenen Datentupeln.

Bild 5b zeigt das Ergebnis des Klassifikators vor und nach einer Rückweisung mit $s_u = 0.575$. Die Kurve der Klassifikationsgüte kann näher an das optimale Klassifikationsergebnis gebracht werden (auf ca. 97.5%), bei vertretbarer Rückweisung von knapp 20% der Datentupel. Interessanterweise ist die Rückweisung zu Beginn des Versuchs sehr gering. Offensichtlich sind in diesem Datensatz viele Beispiele enthalten, die zu Beginn eines Versuchs deutlich einer anderen Klasse zuzuordnen sind. Nur so ist die geringe Rückweisung mit gleichzeitig hohem Klassifikationsfehler zu erklären.

4.5.2 Therapieauswahl in der instrumentellen Ganganalyse

Die für die Operationskategorie „KFe“ (knöcherner Eingriff am Oberschenkel) ausgewählten Merkmale sind in Bild 6a abgebildet. Es ist deutlich zu sehen, dass eine automatische Zuordnung der Therapie zu den Patienten nur mit großen Fehlern möglich ist. Die Verwendung weiterer Merkmale erhöht die Güte der Zuordnung kaum, zumindest nicht in diesem Datensatz.

Hier werden die vorgestellten Algorithmen zur Rückweisung von Klassifikationsentscheidungen so verwendet, dass Regionen mit unterschiedlichen Sicherheiten generiert werden [17]. Dafür werden verschiedene Schwellwerte s_u verwendet. Das ursprüngliche Klassifikationsproblem „Operation wurde durchgeführt (ja)/Operation wurde nicht durchgeführt (nein)“ wird durch diese Schwellwerte erweitert (ja, eher ja, unentschlossen, eher nein, nein). Durch gleichmäßiges Einfügen von Datentupeln kann der Klassifikator Regionen

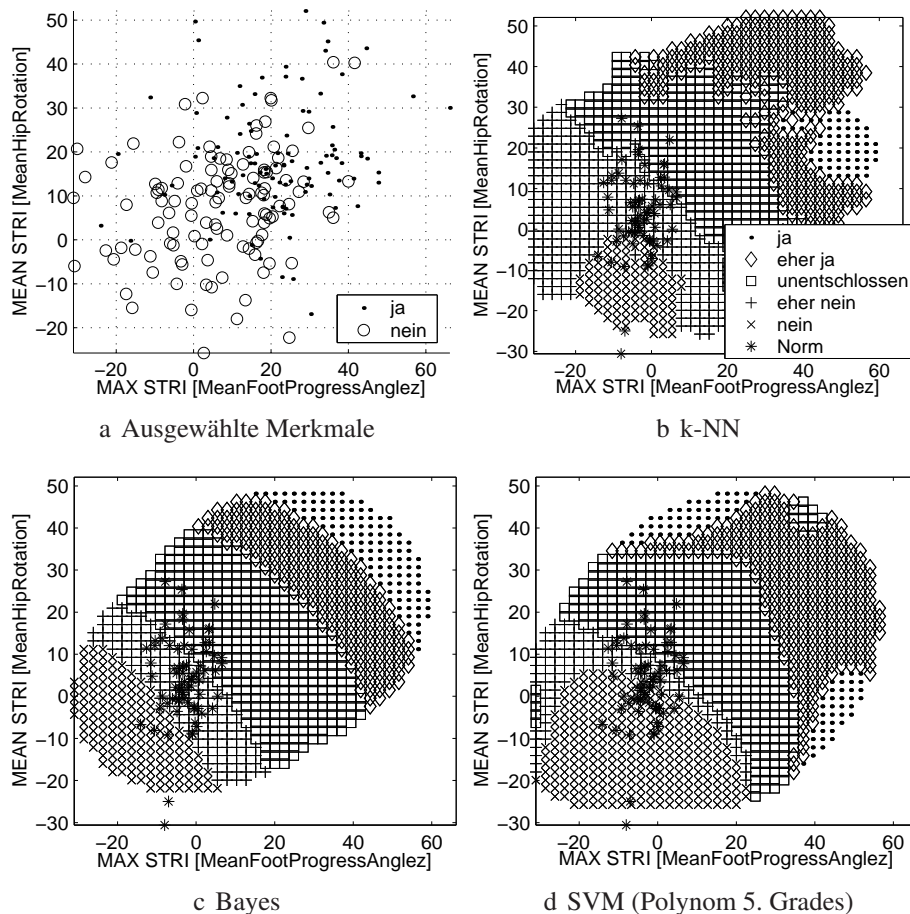


Bild 6: Generierung von Regionen für die Unterstützung bei der Therapieentscheidung

generieren und für den Anwender visualisieren. Dies ist für die verschiedenen Verfahren in den Bildern 6b-6d dargestellt. Als Schwellen wurden folgende Werte verwendet:

- k-NN: $k = 15$, Größe der Region: 0.15, $\beta_{knn} = 0.15$, $s_a = 3$, $s_{u1} = 0.9$, $s_{u2} = 0.7$, $s_{u3} = 0.5$
- Bayes: $s_a = 8$, $s_{u1} = 0.9$, $s_{u2} = 0.75$, $s_{u3} = 0.5$
- SVM: $s_a = -0.05$, $s_{u1} = 1.5$, $s_{u2} = 1$, $s_{u3} = 0$, $\lambda = 5$, RBF-Kernel: 0.8

Die Bilder zeigen, dass das vorgestellte Prinzip der Verwendung mehrerer Schwellwerte gut funktioniert. Welches Verfahren den anderen überlegen ist, ist schwierig zu sagen. Die für den k-nearest neighbor verwendeten Schwellwerte sind etwas restriktiver, wodurch die sicheren Regionen (insbesondere „Nein“) eher klein werden. Der k-nearest neighbor Klassifikator ist durch seine starke Einbindung der einzelnen Lerndaten gegenüber den anderen Verfahren eher in der Lage, kleinere, unförmige Regionen zu erstellen und hat eine selektivere Ausreißerdetektion. Der Bayes-Klassifikator erzeugt relativ glatte Klassengrenzen, wodurch die Regionen teilweise etwas aneinander gereiht erscheinen.

Eine erste Bewertung der Verfahren wurde mittels Kostenfunktionen realisiert. Die Ergebnisse wurden bisher aber nur auf den Lerndaten angewendet und noch nicht mittels Crossvalidierung evaluiert. Außerdem kann die verwendete Kostenfunktion nur eingeschränkt eine Bewertung der Ergebnisse vornehmen, da sie die Entscheidungen für bekannte Patienten bewertet, nicht aber die generierten Regionen.

Alle Verfahren erreichen einen positiven Wert (Tabelle 2). Der k-nearest neighbor Klas-

sifikator hat die kleinsten Regionen mit einer „sicheren“ Entscheidung, was sich auch in der kleinsten Anzahl an Fehlentscheidungen niederschlägt. Aufgrund der variableren Klassengrenzen ist der k-nearest neighbor Klassifikator für die Therapieauswahl vorzuziehen.

Beschreibung	Verfahren		
	k-NN	SVM	Bayes
Korrekte Entscheidung ja/nein (+4)	17	25	20
Korrekte Entscheidung eher ja/eher nein (+2)	63	48	54
Unentschieden (0)	120	126	125
Falsche Entscheidung ja/nein (-20)	0	2	4
Falsche Entscheidung eher ja/eher nein (-10)	12	7	8
Rückweisung wegen zu wenigen Informationen (-4)	4	8	5
Summe Kostenfunktion	58	54	8

Tabelle 2: Ergebnis der Kostenfunktion bei Klassifikation mit Rückweisungsmöglichkeit

5 Zusammenfassung

Diese Arbeit beschreibt verschiedene Aufgaben und Probleme von Data-Mining Anwendungen in der Medizin. Neben der Bestimmung und Visualisierung des Informationsgehalts und der Klassifikation von zeitvarianten Zeitreihen werden Verfahren und Ansätze zur Herstellung der nötigen Sicherheit von Data-Mining Systemen in der Medizin betrachtet. Für jeden der behandelten Bereiche wird die grundsätzliche Funktionalität der Ansätze an verschiedenen Datensätzen aus dem Bereich der Brain Machine Interfaces und der instrumentellen Ganganalyse demonstriert.

Literatur

- [1] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: From Data Mining to Knowledge Discovery in Databases. *AI Magazine* 17 (1996), S. 37–54.
- [2] Knorr, E. M.; Ng, R. T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: *Proc., 24th Very Large Data Bases Conference*. New York, USA. 1998.
- [3] Kaplan, B.: Evaluating Informatics Applications – Clinical Decision Support Systems Literature Review. *International Journal of Medical Informatics* 64 (2001), S. 15–37.
- [4] Kadous, M.: *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series*. Dissertation, The University of New South Wales. 2002.
- [5] Reischl, M.: *Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen*. Dissertation, Universität Karlsruhe, Universitätsverlag Karlsruhe. 2006.
- [6] Mikut, R.; Reischl, M.; Burmeister, O.; Loose, T.: Data mining in medical time series. *Biomedizinische Technik* (2006). Akzeptierter Beitrag.
- [7] Keogh, E. J.; Pazzani, M. J.: A simple dimensionality reduction technique for fast similarity search in large time series databases. In: *Proc., 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, S. 122–133. Kyoto, Japan. 2000.
- [8] Lemm, S.; Schäfer, C.; Curio, G.: BCI Competition 2003 – Data Set III: Probabilistic Modeling of Sensorimotor μ Rhythms for Classification of Imaginary Hand Movements. *IEEE Transactions on Biomedical Engineering* 51(6) (2004), S. 1077–1080.

- [9] Burmeister, O.; Reischl, M.; Mikut, R.: Zeitvariante Klassifikatoren für Brain Machine Interfaces und Neuroprothesen. In: *Proc., 15. Workshop Computational Intelligence*, S. 129–143. Universitätsverlag Karlsruhe. 2005.
- [10] Burmeister, O.; Reischl, M.; Gröll, L.; Mikut, R.: Zeitvariante Klassifikatoren zur Steuerung von Brain Machine Interfaces und Neuroprothesen. *at - Automatisierungstechnik* 54(11) (2006). Im Druck.
- [11] Hellmann, M. E.: The Nearest Neighbor Classification Rule with a Reject Option. *IEEE Transactions on Systems, Science, and Cybernetics* SSC-6(3) (1970), S. 179–185.
- [12] Beck, S.; Mikut, R.; Jäkel, J.: A Cost-Sensitive Learning Algorithm for Fuzzy Rule-Based Classifiers. *Mathware and Soft Computing* 11(2-3) (2004), S. 175–195.
- [13] Beck, S.: *Ein Beitrag zum automatischen Entwurf von Fuzzy-Entscheidungssystemen bei unvollständiger Information*. Dissertation, Universität Karlsruhe, Universitätsverlag Karlsruhe. 2005.
- [14] Lal, T. N.; Hinterberger, T.; Widman, G.; Schröder, M.; Hill, J.; Rosenstiel, W.; Elger, C. E.; Schölkopf, B.; Birbaumer, N.: Methods Towards Invasive Human Brain Computer Interfaces. In: *Advances in Neural Information Processing Systems*, S. 737–744. Cambridge, MA, USA: MIT Press. 2005.
- [15] Ramoser, H.; Müller-Gerking, J.; Pfurtscheller, G.: Optimal Spatial Filtering of Single Trial EEG during Imagined Hand Movement. *IEEE Transactions on Rehabilitation Engineering* 8(4) (2000), S. 441–446.
- [16] Loose, T.: *Konzept für eine modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse*. Dissertation, Universität Karlsruhe, Universitätsverlag Karlsruhe. 2004.
- [17] Burmeister, O.; Mikut, R.; Wolf, S.: Towards a Data Mining Based Decision Support System for Treatment Planning in Instrumented Gait Analysis. *Biomedizinische Technik, Proc. Gemeinsame Jahrestagung der Deutschen, Österreichischen und Schweizerischen Gesellschaften für Biomedizinische Technik* (2006).
- [18] Wolf, S.; Loose, T.; Schablowski, M.; Döderlein, L.; Rupp, R.; Gerner, H. J.; Bretthauer, G.; Mikut, R.: Automated Feature Assessment in Instrumented Gait Analysis. *Gait & Posture* 23(3) (2006), S. 331–338.
- [19] Blum, A. L.; Langley, P.: Selection of relevant features and examples in machine learning. *Artificial Intelligence* 97 (1997), S. 245–271.
- [20] Kohavi, R.; John, G. H.: Wrappers for Feature Subset Selection. *Artificial Intelligence* 97 (1-2) (1997), S. 273–324.
- [21] Tatsuoka, M. M.: *Multivariate Analysis*. New York: Macmillan. 1988.
- [22] Shannon, C. E.: A Mathematical Theory of Communication. *The Bell System Technical Journal* 27 (1948), S. 379–423.
- [23] Chow, C. K.: On Optimum Error and Reject Tradeoff. *IEEE Transactions on Information Theory* 16 (1970), S. 41–46.
- [24] Hautamäki, V.; Kärkkäinen, I.; Fränti, P.: Outlier Detection using K-Nearest-Neighbour Graph. In: *Proc., International Conference on Pattern Recognition*, S. 430–433. Cambridge, UK. 2004.
- [25] Tax, D. M. J.: *One-Class Classification*. Dissertation, Technische Universität Delft. 2001.
- [26] Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining* 2(2) (1998), S. 121–167.
- [27] Campbell, C.; Bennett, K. P.: A Linear Programming Approach to Novelty Detection. In: *Proc., Neural Information Processing Systems Conference*. 2000.
- [28] Berkelaar, M.; Eikland, K.; Notebaert, P.: Open Source (Mixed-Integer) Linear Programming System, Version 5.5.0.7. 2005.
- [29] Cover, T.; Hart, P.: Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1) (1967), S. 21–27.
- [30] Giacinto, G.; Roli, F.: Dynamic Classifier Selection Based on Multiple Classifier Behaviour. *Pattern Recognition* 34 (2001), S. 1879–1881.
- [31] Giacinto, G.; Roli, F.; Bruzzone, L.: Combination of Neural and Statistical Algorithms for Supervised Classification of Remote-Sensing Images. *Pattern Recognition Letters* 21 (2000), S. 385–397.

Fuzzy-Konzepte für die Begriffe Gesundheit, Kranksein und Krankheit

Darstellung und Implementierung eines Ansatzes
von Kazem Sadegh-Zadeh

Julia Limberg und Rudolf Seising

Institut für Medizinische Experten- und Wissensbasierte Systeme
Besondere Einrichtung für Medizinische Statistik und Informatik

Medizinische Universität Wien

Spitalgasse 23, A-1090 Wien, Österreich

Tel. (0043) 1 40400 6664

Fax (0043) 1 404006667

E-Mail: rudolf.seising@meduniwien.ac.at

1 Einleitung

Gesundheit, Krankheit und Kranksein sind medizintheoretische Begriffe, die sich der strengen zweiwertigen Logik entziehen. Der iranisch-deutsche Mediziner und Medizinphilosoph Kazem Sadegh-Zadeh (Abb. 1) diskutiert deshalb seit den 1980er Jahren die Bedeutung dieser Begriffe in einer neuen Fassung. Dafür wählte er einen fuzzy-theoretischen Zugang ([1], S. 607):

“health is a matter of degree,
illness is a matter of degree,
and disease is a matter of degree.”

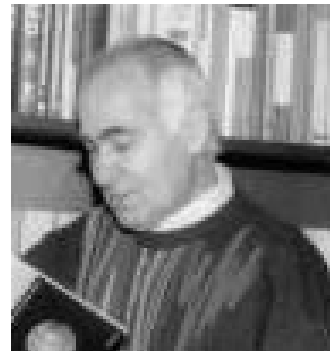


Abb. 1: Kazem Sadegh-Zadeh.

Schon 1982 hatte er bei einer Konferenz über Medizin und Philosophie den Begriff der *patienthood* (von Leiden gequält, being afflicted by a malady) als neuen medizintheoretischen Begriff eingeführt “of which the notion of *health* will be the additive inverse in the following sense:

$$\text{Health} = 1 - \text{patienthood} \text{ [2].}$$

Im Jahre 2000 präsentierte Sadegh-Zadeh einen konzeptuellen Vorschlag auf den Grundlagen der Fuzzy Set Theorie. In seinem Artikel „Fuzzy Health, Illness, and Disease“ [1] fuzzifizierte er die Begriffe „Gesundheit“, „Kranksein“ und „Krankheit“, und erweiterte so ihren Sinngehalt. Besondere Beachtung schenkte er dabei dem Begriff Fuzzy-Krankheit (fuzzy disease), dessen Interpretation und Definition sich als sehr komplex erweist.

In diesem Beitrag wird nach einem kurzen Abriss dieses medizintheoretischen Fuzzy-Konzepts von Sadegh-Zadeh eine Software vorgestellt, mit dem dieser Ansatz umgesetzt wurde. Aufgrund von differenzierten Angaben, die ein Patient oder Arzt (bzw. beide gemeinsam) über die Symptome eingeben kann, soll die Identifikation von Krankheiten ermöglicht werden.

2 Motivation

Die genaue Deutung der Begriffe „Gesundheit“, „Kranksein“ und „Krankheit“ wurde am Ende des 20. Jahrhunderts von verschiedenen philosophischen Richtungen diskutiert. Letzten Endes konnte sich keine einheitliche Definition herauskristallisieren, da den unterschiedlichen Diskussionsgrundlagen eine logische Analyse der drei Begriffe fehlte. Ihnen gemeinsam war der Versuch, die Begriffe Gesundheit, Kranksein und Krankheit in eine klassische, zweiwertige logische Struktur zu bringen, für die folgende Aussagen gelten würden:

1. Ein Individuum ist gesund oder nicht gesund, kann aber nie beides zugleich sein.
2. Ein Individuum ist krank oder nicht krank, kann jedoch nie beides zugleich sein.
3. Ein Individuum hat eine Krankheit oder nicht, aber nie beides zugleich.

Als weitere, verbreitete Annahme ist zu nennen:

- Gesundheit und Krankheit sind Gegensätze, die einander ausschließen. Gesundheit bedeutet die Abwesenheit von Krankheit und umgekehrt.

Sadegh-Zadeh geht davon aus, dass diese Annahmen nicht zulässig sind und die Begriffe Gesundheit, Kranksein und Krankheit nicht im Sinne der klassischen Logik erfasst werden können. Es kann sehr wohl sein, dass eine Person gesund und gleichzeitig nicht gesund oder krank und gleichzeitig nicht krank ist. Auch kann man eine Krankheit haben und nicht haben. Die Begriffe Gesundheit und Krankheit schließen sich nicht aus: Eine Person kann gesund sein und trotzdem eine Krankheit haben, oder eben nicht gesund sein, ohne eine Krankheit zu haben. Die Begriffe Gesundheit, Kranksein und Krankheit haben offenbar unscharfe Grenzen. Sie entziehen sich der klassischen zweiwertigen Logik. Deshalb versuchte Sadegh-Zadeh, sie als fuzzy set theoretische Begriffe zu definieren.

3 Fuzzy Sets – theoretische Grundlagen

„Fuzzy Sets“ wurden im Juni 1965 in einem Artikel der Zeitschrift *Information and Control* von Lotfi A. Zadeh (Abb. 2), einem Professor für Elektrotechnik an der University of California in Berkeley, als Grundbestandteile einer neuen mathematischen Theorie eingeführt [3-5]. Während für eine Menge in der gewöhnlichen Mengentheorie eindeutig feststeht, ob ein bestimmtes Objekt ihr als Element angehört oder nicht, ist die Zugehörigkeit eines Objekts zu einer Fuzzy Menge unscharf. Diese Unschärfe wird durch Werte zwischen 0 und 1 ausgedrückt, die jedem Objekt durch die Zugehörigkeitsfunktion (membership function) der Fuzzy Menge zugeordnet werden (Abb. 3).



Abb. 2: Lotfi A. Zadeh.

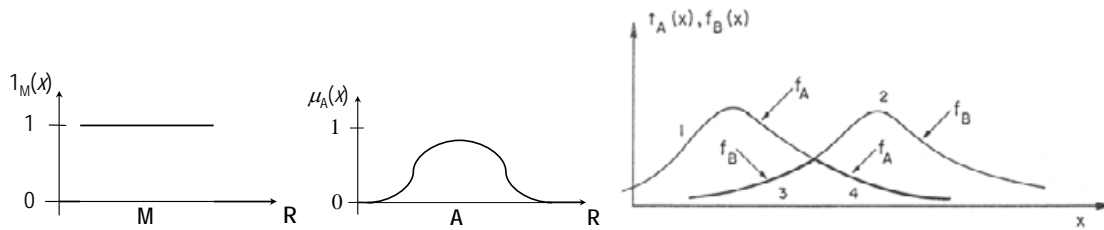


Abb. 3: Links: Charakteristische Funktion einer Menge M . Mitte: Zugehörigkeitsfunktion einer Fuzzy Menge A . Rechts: Zugehörigkeitsfunktionen $f_A(x)$ und $f_B(x)$ der Fuzzy Sets A und B sowie ihrer Vereinigung und ihres Durchschnitts. $f_{A \cup B}(x)$ ergibt sich aus den beiden Graphenstücken 1 und 2, $f_{A \cap B}(x)$ ergibt sich aus den stärker gezeichneten Stücken 3 und 4. Zadeh bezeichnete Zugehörigkeitsfunktionen in diesem ersten Artikel noch mit dem Buchstaben f , später hat sich dafür der griechische Buchstabe μ allseits eingebürgert. ([3], S. 342).

Mit unscharfen Mengen kann ähnlich operiert werden, wie mit gewöhnlichen, „scharfen“ Mengen: es gibt das leere Fuzzy Set ($\mu_A(x) = 0 \forall x$ in der Grundmenge) und das Komplement jedes Fuzzy Sets ($\mu_{A^c}(x) = 1 - \mu_A(x)$); Fuzzy Sets können vereinigt werden und man kann ihren Durchschnitt bilden. Die beiden letzten Operationen definierte Zadeh analog zur gewöhnlichen Mengentheorie: Die Zugehörigkeitsfunktion der Vereinigung zweier Fuzzy Sets A, B ist das Maximum der Zugehörigkeitsfunktionen beider Fuzzy Sets und die ihres Durchschnitts ist deren Minimum (Abb. 4).

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)),$$

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)).$$

Eine mathematische Theorie zur Beschreibung von Zuständen operiert mit Variablen, die gewisse Werte annehmen können, und diese sind in den klassischen wissenschaftlichen Theorien numerisch. Beispielsweise wird die Variable *Länge* in numerischen Angaben zur Einheit Meter, die Variable *Zeit* durch eine Anzahl von Sekunden angegeben; bezogen auf ein Lebewesen erhält die Variable *Alter* z. B. in der medizinischen Diagnostik einen Wert durch die Anzahl der Lebensjahre eines Patienten.

Zadehs “Outline of a New Approach to the Analysis of Complex Systems and Decision Processes” aus dem Jahre 1973 enthält die Einführung des Begriffs “linguistischer Variablen”, “that are variables whose values may be sentences in a specific natural or artificial language. [6] Die Werte dieser Variablen werden in Worten ausgedrückt, so beispielsweise in der linguistischen Variable “Alter” durch *jung*, *sehr jung*, *nicht sehr jung*, *alt*, *sehr alt* etc. (Abb. 5). Dem numerisch bestimmten Alter eines Patienten von 100 Jahre, würde z. B. ein linguistischer Ausdruck wie „sehr alt“ entsprechen.

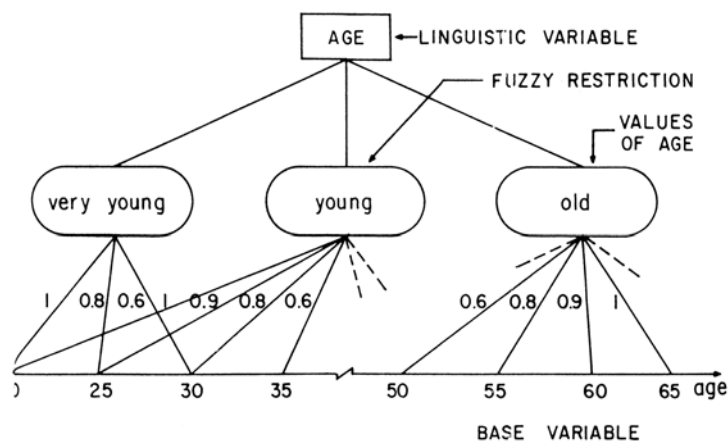


Abb. 5: Zadehs linguistische Variable *Alter* (*age*) [6].

Linguistische Variablen haben sich seither zu einem guten Werkzeug für jene Fälle entwickelt, in denen man ohne exakte numerische Werte auskommen muss, sei es weil diese zu berechnen oder zu messen nicht möglich sind, sei es weil zu viel Zeit benötigt würde und dies mit zu hohen Kosten verbunden ist. Das Konzept hat sich als sehr erfolgreich erwiesen, z. B. bei Systemen zur Regelung oder Entscheidungsunterstützung – nicht zuletzt auch in der medizinischen Diagnose [7].

Sadegh-Zadeh definiert den *Gesundheitszustand* eines Individuums als linguistische Variable (Abb. 6) mit folgender Termmenge:

$$T_{state\ of\ health} = \{ wohl, nicht\ wohl, sehr\ wohl, krank, nicht\ krank, \dots \}.$$

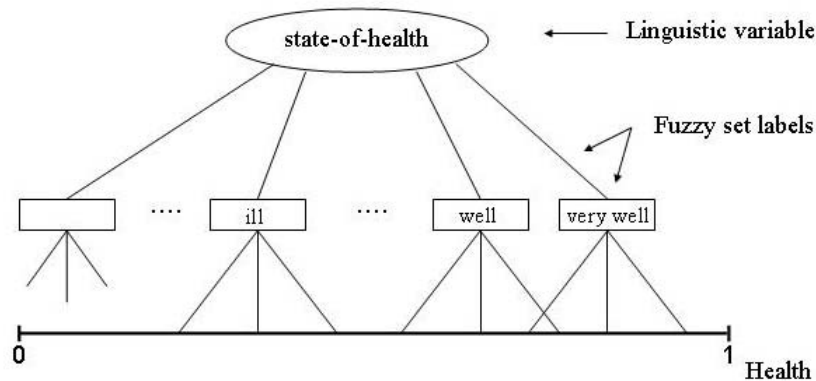


Abb. 6: Sadegh-Zadehs linguistische Variable *Gesundheitszustand* (*state of health*) [1].

Dies wirft zusätzliche Fragen auf: Während das Wörtchen „nicht“ intuitiv als Komplementbildung in der Fuzzy Set Theorie interpretiert werden kann, ist die fuzzy set-theoretische Deutung des Wörtchens „sehr“ nicht unmittelbar klar. Auch hier griff Sadegh-Zadeh auf Lotfi Zadehs Definitionen von 1973 zurück, in denen dieser „Modifikatoren“ für Fuzzy Mengen eingeführt hatte – darunter die „Konzentration“ und die „Dilatation“ [6]: Sei A eine Fuzzy Menge so ist deren

- Konzentration (*concentration*) $CON A = A^2$,
- Dilatation (*dilatation*) $DIL A = A^{1/2}$,

Das Wort „sehr“ konzentriert beziehungsweise verstärkt den Sinngehalt eines Terms, während ein „mehr oder weniger“ seinen Sinngehalt ausdehnt bzw. dilatiert. Für eine Fuzzy Menge A soll demnach gelten:

- *sehr* A ist eine Fuzzy Menge mit $\mu_{sehr(A)}(x) = [\mu_A(x)]^2$
- *mehr oder weniger* A ist eine Fuzzy Menge mit $\mu_{mehr-oder-weniger(A)}(x) = [\mu_A(x)]^{1/2}$.

Sadegh-Zadeh übernahm diese Definitionen und erhielt folgende Beziehungen für die entsprechenden Zugehörigkeitsfunktionen:

- $\mu_{sehr-wohl}(x) = (\mu_{wohl}(x))^2$
- $\mu_{mehr-oder-weniger-wohl}(x) = (\mu_{wohl}(x))^{1/2}$
- $\mu_{Kranksein}(x) = ((\mu_{Kranksein}(x))^{1/2})^2 = (sehr(\mu_{unwohl}(x)))^2 = \mu_{unwohl}(x)^4$

Im Falle der oben eingeführten linguistischen Variable $T_{state\ of\ health}$ kann man also schreiben: *sehr wohl* entspricht $wohl^2$, *mehr oder weniger wohl* entspricht $wohl^{1/2}$ und Kranksein entspricht *sehr (mehr oder weniger Kranksein)*, was wiederum *sehr sehr unwohl*, also $unwohl^4$ entspricht. Insgesamt lässt sich folgern, dass Kranksein nicht das begriffliche Gegenteil von Gesundheit ist.

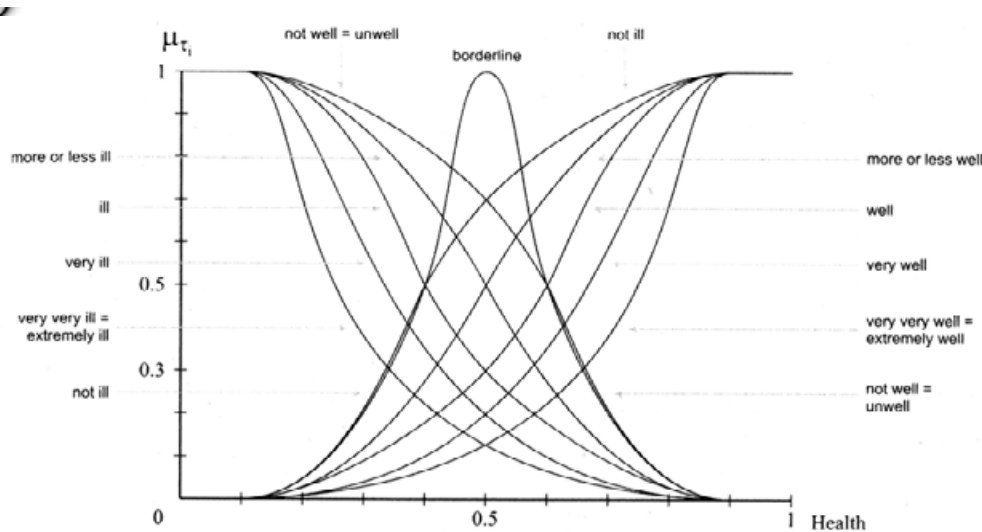


Abb. 7: Illustration der Fuzzy Mengen zum Gesundheitszustand (*state of health*) [1]. Alle Fuzzy Mengen entstehen durch Komplementbildung, Konzentration oder Dilatation der Fuzzy Menge „wohl“ (*well*)

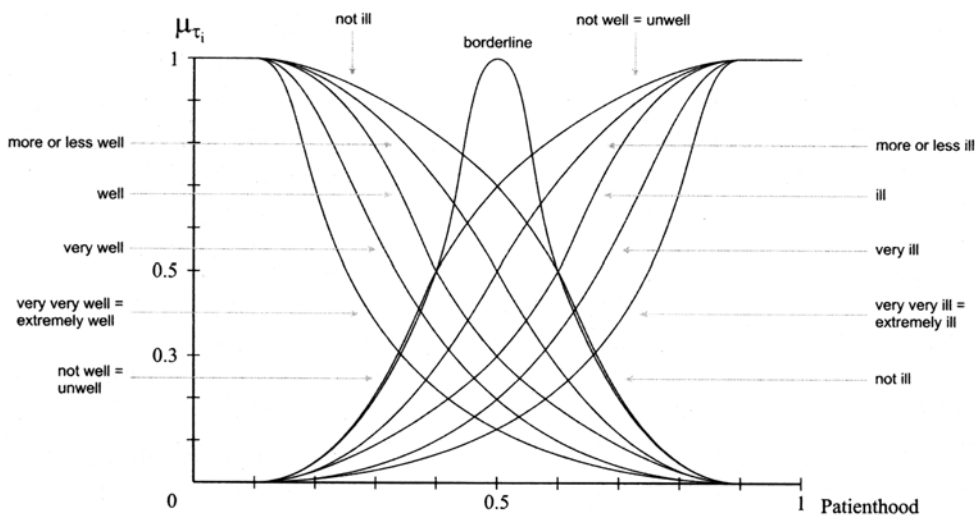


Abb. 8: Illustration der gleichen Fuzzy Mengen wie in Abb. 7, jedoch auf dem Begriff *Patienthood* beruhend [1].

4 Fuzzy-Gesundheit

Die WHO (World Health Organization, Weltgesundheitsorganisation) definiert: "Health is a state of complete physical, mental and social well-being and not merely the absence of disease or infirmity" [8] – Gesundheit ist mehr als nur die Abwesenheit von Krankheit. Gesundheit ist ein Zustand des vollkommenen körperlichen, seelischen und sozialen Wohlbefindens. Aus dieser Definition geht bereits hervor, dass „Gesundheit“ nicht das begriffliche Gegenteil von „Krankheit“ ist. Dieser Meinung schließt sich Sadegh-Zadeh an und er befindet vielmehr den Begriff des „Leidens“ als den Gegensatz zur Gesundheit. Dazu führt er den Begriff „patienthood“ ein. Patienthood bedeutet, von Leiden

gequält zu sein. Folglich definiert Sadegh-Zadeh diesen Begriff als das (Fuzzy-) Inverse zum Begriff Gesundheit:

$$\text{Gesundheit} = 1 - \text{Patienthood.}$$

Für ein Individuum x ergibt sich daraus:

$$\mu_{\text{Gesundheit}}(x) = 1 - \mu_{\text{Patienthood}}(x).$$

Ein Individuum hat beispielsweise den Zugehörigkeitsgrad 0.6 zur Menge H (health, Gesundheit) der gesunden Individuen, wenn es den Zugehörigkeitsgrad 0.4 der Menge P (patienthood, Leiden) hat. Operationen auf diesen Mengen verhalten sich nach den Regeln der Fuzzy Set Theorie.

Der Durchschnitt der Fuzzy Mengen im oben genannten Beispiel ergibt sich durch die Berechnung des Zugehörigkeitsgrades $\min(0.6, 0.4) = 0.4$. Das Ergebnis ist nicht leer.

Gesundheit und Patienthood sind also gemäß der Fuzzy Set Theorie komplementäre Mengen – anders als in der zweiwertigen Logik aber nicht unvereinbar.

Es sei nochmals darauf hingewiesen, dass Gesundheit und Krankheit durchaus nebeneinander existieren können. Es mag sein, dass ein Patient beispielsweise von *calcinosis circumscripta* (Kalkgicht) der Schilddrüse befallen ist. Durch diese Krankheit werde der Patient jedoch nicht von *Leiden* gequält. Deshalb ist sein Zugehörigkeitsgrad zur Menge Gesundheit gleich 1. Krankheit beeinflusst also nicht den Zustand der Gesundheit, solange die Krankheit den Wert der Patienthood nicht erhöht.

5 Fuzzy- Kranksein

Im deutschen Sprachgebrauch wird nicht oder nur selten zwischen Kranksein und Krankheit differenziert. Die englische Sprache unterscheidet dagegen explizit zwischen Krankheit (disease) und Kranksein (illness) und in der neueren Medizinsoziologie [9] wird dieser Unterschied betont, da jemand eine Krankheit haben kann, ohne sich krank zu fühlen. Ebenso kann man leiden ohne eine Krankheit zu haben.

Zur Beurteilung des Gesundheitsstatus eines Menschen gilt im Allgemeinen folgende gesellschaftliche Übereinkunft: Kranksein und Gesundheit können nicht nebeneinander existieren. Doch wie im Falle des Krankheitsbegriffs, kann man auch beim Begriff des Krankseins nicht vom exakten logischen Gegenteil zum Gesundheitsbegriff sprechen. Kranksein und Wohlbefinden sind nur zwei von vielen möglichen Ausprägungen des Gesundheitszustands eines Lebewesens. Der Kranksein-Begriff wird daher eingeführt, um den Gesundheitsstatus eines Individuums zu beschreiben, und schließlich einen Brückenschlag zum Begriff der Fuzzy-Krankheit zu vollziehen.

6 Fuzzy-Krankheit

Was ist Krankheit? – Zur Definition des Begriffs Fuzzy-Krankheit ist zunächst diese Frage zu klären. Da Krankheit auf verschiedene Arten, unter anderem linguistisch und sozial, definiert werden kann, führt Sadegh-Zadeh zunächst potentielle Kandidaten für Krankheiten ein. Dies sind komplexe menschliche Beschaffenheiten – oder Befindlichkeiten – wie Herzinfarkt, Apoplexie, Brustkrebs, etc. Aussagen wie „Herzinfarkt ist eine Krankheit“ sind im üblichen Sprachgebrauch bekannt. Weitere in der Gesellschaft als Krankheiten identifizierte Befindlichkeiten sollen nun hinzugezogen werden, um dem Krankheitsbegriff näher zu kommen.

Diese Beschaffenheiten ergeben sich nicht nur aus dem biologischen Zustand des Körpers. Sie können als große Fuzzy Mengen dargestellt werden, die ganz verschiedene, auch religiöse oder soziale Aspekte des Umfeldes des Kranken erfassen.

Hierzu gehören auch Befindlichkeiten, die mit „Schmerz“, „Leid“, „Gefühl des Alleinseins“, usw. ausgedrückt werden.

Zur umfassenden Definition der Krankheit gehört eine Menge von menschlichen Beschaffenheiten (D) mit den zugehörigen Kriterien (C). Es gilt:

1. Jedes Element der Grundmenge $\{D_1, D_2, \dots, D_n\}$ ist eine Krankheit, und
2. jedes Ereignis, das einer Krankheit hinsichtlich der Kriterien $\{C_1, C_2, \dots, C_n\}$ ähnelt, ist eine Krankheit.

Die erste Aussage erscheint einleuchtend, die zweite Aussage stellt jedoch ein Problem dar, da die *Ähnlichkeit* dargestellt werden muss. Zu diesem Zweck wird zunächst die Fuzzy Mengen-Differenz zweier Fuzzy Mengen A und B – Differenz (A, B) – eingeführt, die folgendermaßen berechnet wird:

$$\text{Differenz } (A, B) = \frac{\sum_i \max(0, \mu_A(x_i) - \mu_B(x_i)) + \sum_i \max(0, \mu_B(x_i) - \mu_A(x_i))}{c(A \cup B)}$$

Im Nenner ist c die Summe der Zugehörigkeitswerte der entsprechenden Fuzzy Menge (*fuzzy set count*). Sei beispielsweise eine Fuzzy Menge X gegeben mit $X = \{(x, 0.6), (y, 0.9)\}$, so ergibt sich $c(X)$ zu: $0.6 + 0.9 = 1.5$.

Zurück zur Fuzzy-Differenz. Es sei eine Fuzzy Menge Y mit $Y = \{(x, 0.7), (y, 0.4)\}$ gegeben. Für den Wert von $c(X \cup Y)$ ergibt sich: $0.7 + 0.9 = 1.6$.

Die Fuzzy-Differenz Differenz (X, Y) errechnet sich durch $\frac{(0 + 0.5) + (0.1 + 0)}{1.6}$ zu 0.375.

X unterscheidet sich von Y also zu einem Ausmaß von 0.375.

Die Ähnlichkeit zweier Fuzzy Mengen ist das Inverse zur Fuzzy-Differenz. Auf das Beispiel bezogen ergibt sich die Ähnlichkeit zu: $1 - 0.375 = 0.625$.

Um nun nicht Äpfel mit Birnen zu vergleichen, sollte sich die Darstellung von Ähnlichkeiten auf eine vergleichbare Teilmenge von Attributen der Kriterien beschränken. Man stellt sich also zum Beispiel die Frage, wie ähnlich sich zwei Krankheiten D_i und D_j hinsichtlich einiger Kriterien $\{C_1, C_2, \dots, C_m\}$ sind, die sie zu einem gewissen Ausmaß teilen.

Sei A eine Fuzzy Menge beliebiger Größe und X ein Teil dieser Menge, also $A \setminus X$. Menschliche Beschaffenheiten wie Herzinfarkt und Magengeschwür kann man hinsichtlich ihrer vergleichbaren Kriterien $\{C_1, C_2, \dots, C_m\}$ einteilen:

- Herzinfarkt $\{(C_1, a_1), (C_2, a_2), \dots, (C_m, a_m)\}$
- Magengeschwür $\{(C_1, b_1), (C_2, b_2), \dots, (C_m, b_m)\}$
- Herzinfarkt $\{(körperliche_Verletzung, 1), (Schmerz, 0.7), (Leid, 0.8)\}$
- Magengeschwür $\{(körperliche_Verletzung, 1), (Schmerz, 0.3), (Leid, 0.5)\}$

Um Ähnlichkeiten zwischen Fuzzy Mengen zu berechnen, wird folgendes Theorem benutzt:

$$\textit{Theorem: similar}(A, B) = \frac{c(A \cap B)}{c(A \cup B)}$$

Teilweise Vergleiche beinhalten gewisse Grade teilweiser Ähnlichkeit ($p = \text{partiell}$) – symbolisiert als $p\text{-similar}(A \setminus X, B \setminus Y)$ – nach der folgenden Definition:

$$p\text{-similar}(A \setminus X, B \setminus Y) = r, \text{ wenn similar}(X, Y) = r.$$

Für das obige Beispiel würde dies unter Verwendung des obigen Theorems bedeuten:

$$p\text{-similar}(\text{Herzinfarkt} \setminus X, \text{Magengeschwür} \setminus Y) = 0.72$$

Sei $\{D_1, \dots, D_n\}$ eine kleine Menge menschlicher Beschaffenheiten. Jede dieser Beschaffenheiten wird in einer bestimmten menschlichen Gesellschaft, aufgrund einer Menge von Kriterien $\{C_1, \dots, C_n\}$, die diese Beschaffenheiten zu einem bestimmten Maß gemeinsam haben, als Krankheit angesehen. Für solch eine Gesellschaft gibt es eine Übereinstimmung zu einem Grad ε der partiellen Ähnlichkeit. Dieser Grad dient als Pfeiler für den Krankheitsbegriff dieser Gesellschaft:

1. Jedes Element der Grundmenge $\{D_1, \dots, D_n\}$ ist eine Krankheit.
2. Eine menschliche Beschaffenheit $H \setminus X$ ist eine Krankheit, wenn es eine Krankheit $D_i \setminus Y \in \{D_1, \dots, D_n\}$ und ein $\varepsilon > 0$ gibt, so dass $p\text{-similar}(H \setminus X, D_i \setminus Y) \geq \varepsilon$.

Gegeben sei die Menge Herzinfarkt $\{(C_1, 1), (C_2, 0.7), (C_3, 0.8)\}$ als Element der Grundmenge $\{D_1, \dots, D_n\}$ und somit ist sie per Definition eine Krankheit.

Die Frage, ob etwas, das nicht in der Grundmenge $\{D_1, \dots, D_n\}$ vertreten ist – wie die Eigenschaft, Hämorrhoiden zu haben – als Krankheit bezeichnet werden kann, wird nach dem Grad ε der partiellen Ähnlichkeit entschieden.

Wenn beispielsweise $\varepsilon = 0.6$ verlangt wird und es eine menschliche Beschaffenheit wie Hämorrhoiden $\{(C_1, 0.9), (C_2, 0.2), (C_3, 0.55)\}$ gibt, ergibt das:

$$p\text{-similar}(\text{Hämorrhoiden} \setminus X, \text{Herzinfarkt} \setminus Y) = 0.66.$$

Da $0.66 > 0.6$ kann die Eigenschaft, Hämorrhoiden zu haben in diesem Falle als Krankheit bezeichnet werden.

Die richtige Wahl eines ε ist bei dieser Definition von Krankheit entscheidend: Je kleiner ε gewählt wird, desto mehr Krankheiten gibt es und umgekehrt. Den Wert für ε hat jedoch nicht der Mediziner, sondern die Gesellschaft zu bestimmen.

Dieser Krankheitsbegriff ist allerdings ein mit der zweiwertigen Logik zur erfassender Begriff, denn er unterscheidet scharf zwischen Zuständen, die einer Krankheit entsprechen, und solchen, die dies nicht tun. Sadegh-Zadeh erweitert diesen Krankheitsbegriff nun zu einem Begriff der „Krankheit zu einem gewissen Grad“.

Dies ergibt sich durch folgende Definition:

Sei \mathcal{H} die Menge der menschlichen Beschaffenheiten. Eine Fuzzy Menge \mathcal{D} über \mathcal{H} wird nur dann eine Menge von Krankheiten genannt, wenn eine Teilmenge $\{D_1, \dots, D_n\}$ von \mathcal{H} und eine Funktion $\mu_{\mathcal{D}}: \mathcal{H} \rightarrow [0,1]$ gegeben sind, so dass für diese Funktion gilt:

$\mu_{\mathcal{D}}(H_i \setminus X) =$

- 1, wenn $H_i \setminus X \in \{D_1, \dots, D_n\}$, genannt Musterkrankheit
- ε , wenn es eine Musterkrankheit $H_j \setminus Y$ mit $p\text{-similar}(H_i \setminus X, H_j \setminus Y) = \varepsilon$ gibt, und es keine Musterkrankheit $H_k \setminus Z$ mit $p\text{-similar}(H_i \setminus X, H_k \setminus Z) > \varepsilon$ gibt und $\mathcal{D} = \{(H_i, \mu_{\mathcal{D}}(H_i)) \mid H_i \in \mathcal{H}\}$.

In dieser erweiterten Definition wird nun eine Fuzzy Menge folgender Art geschaffen:

$\mathcal{D} = \{(D_1, \mu_{\mathcal{D}}(D_1)), \dots, (D_q, \mu_{\mathcal{D}}(D_q))\}$, die aus individuellen Krankheits-Modellen besteht, die alle zu verschiedenen Graden der Menge \mathcal{D} zugehörig sind. Der Zugehörigkeitsgrad $\mu_{\mathcal{D}}(D_i)$ liegt im Intervall $[0,1]$.

Diese neuen Erkenntnisse werden nun auf das Hämorrhoiden-Beispiel angewandt:

Hämorrhoiden $\setminus \{(C_1, 0.9), (C_2, 0.2), (C_3, 0.55)\}$. Die Kriterien werden mit einer Musterkrankheit verglichen. Hinzugezogen wird die bereits verwendete Herzinfarkt-Menge: Herzinfarkt $\setminus \{(C_1, 1), (C_2, 0.7), (C_3, 0.8)\}$.

Durch diesen Vergleich ergibt sich nach der vorherigen Berechnung, dass man bei Vorliegen von Hämorrhoiden zu 0.66 von einer Krankheit sprechen kann.

Zöge man ein anderes Individuum, mit einer anderen Menge – Hämorrhoiden $\setminus \{(C_1, 0.2), (C_2, 0.1), (C_3, 0.1)\}$ – zu einem Vergleich mit Herzinfarkt heran, ergäbe sich ein Krankheits-Zugehörigkeitsgrad von 0.16, wie man leicht nachrechnen kann.

Daraus wird offensichtlich, dass eine Person zu einem gewissen Grad eine Krankheit haben und gleichzeitig zu einem gewissen Grad keine Krankheit haben kann.

7 Implementierung der Fuzzy-Krankheit

7.1 Motivation

Die Software wurde basierend auf dem Ansatz, Krankheiten über Fuzzy Mengen zu definieren, entwickelt. Über eingegebene Zugehörigkeitsgrade bestimmter Symptome werden Ähnlichkeiten zu bestehenden Krankheiten ermittelt und es wird aufgezeigt, ob die eingegebene Symptomgruppe Hinweise auf eine Krankheit gibt. Des Weiteren sollen neue Musterkrankheiten und Symptome angelegt und modifiziert werden können.

7.2 Software-Beschreibung

Die Software ist in der Programmiersprache Java geschrieben. Grundlage der Software-Architektur ist die Gliederung in verschiedene Schichten. Dadurch wird eine bessere Wartbarkeit erreicht. Als Datenbank wird HSQLDB eingesetzt.

7.3 Datenbank-Beschreibung

Die Datenbank verfügt über die Entitäten Krankheit, Symptom und Symptomliste. Eine Krankheit wird über eine laufende Nummer und einen Namen eindeutig identifiziert. Analoges gilt für Symptome. Die Entität Symptomliste dient als Bindeglied zwischen Symptom und Krankheit: In einem Symptomlisteneintrag wird eine Krankheits-ID, eine Symptom-ID und die entsprechende Fuzzy Menge gespeichert. Durch diese Datenbank-Architektur ist die Datenbank leicht modifizierbar und stabil.

Zur Demonstration sind in der Datenbank bereits neun Symptome und 12 Krankheiten mit bestimmten Zugehörigkeitsgraden angelegt. Ein Teil der Krankheiten und ihrer Fuzzy Mengen wurde direkt den Vorschlägen von Sadegh-Zadehs Artikel entnommen. Weitere Krankheiten und Zugehörigkeitsgrade wurden in Zusammenarbeit mit einem Internisten ermittelt.

Folgende Symptome sind vorangelegt: Schmerz, Leiden, Verletzung, Schnupfen, Husten, Fieber, Atemnot, Übelkeit, Schwindel.

Als Musterkrankheiten dienen Herzinfarkt, Schlaganfall, Magengeschwür, Fieberhafte Erkältung, Grippe, Asthma, Gallenblasenentzündung, Nierenkolik, Gicht, Migräne, Divertikulitis, Gürtelrose.

Die folgenden zwei Tabellen geben einen Überblick über die bereits angelegten Krankheiten, Symptome und Zugehörigkeitsgrade:

	Herzinfarkt	Schlaganfall	Magengeschwür	Fieberhafte Erkältung	Grippe	Asthma
Schmerz	0,7	0,3	0,3	0,2	0,8	0
Leiden	0,8	0,6	0,5	0,4	1	1
Verletzung	1	1	1	0,2	0,8	0,3
Schnupfen	0	0	0	0,9	0,3	0,1
Husten	0	0	0	0,7	0,5	0,2
Fieber	0	0	0	0,4	0,9	0
Atemnot	0,5	0	0	0	0,2	1
Übelkeit	0,3	0,2	0,5	0,2	0,2	0,1
Schwindel	0,1	0,3	0	0,1	0,4	0,2

	Gallenblasen-entzündung	Nierenkolik	Gicht	Migräne	Divertikulitis	Gürtelrose
Schmerz	0,9	1	1	0,9	0,7	0,8
Leiden	0,9	1	0,8	0,9	0,8	0,5
Verletzung	0,8	0,5	0,3	0,1	0,5	0,3
Schnupfen	0	0	0	0	0	0
Husten	0	0	0	0	0	0
Fieber	0	0	0	0	0,7	0,2
Atemnot	0	0	0	0	0	0
Übelkeit	0,8	0,8	0,3	0,5	0,3	0,8
Schwindel	0,2	0,2	0	0,3	0	0

Abb. 9: Übersichtstabellen zu vorangelegten Symptomen, Krankheiten und Zugehörigkeitsgraden.

7.4 Programmoberflächen-Beschreibung

Bei erstmaligem Start des Programms – durch Doppelklick auf das Fuzzy.jar-File – wird ein Datenbank-Verzeichnis „fuzzydb“ auf dem C:-Laufwerk angelegt und eine Datenbank installiert. Ein Hinweis, dass eine Datenbank angelegt werden muss, erscheint. Der Benutzer muss sich damit einverstanden erklären, dass eine Datenbank „fuzzydb1“ initialisiert wird.

Daraufhin, sowie bei zukünftigen Programmstarts, gelangt der Anwender in das Hauptmenü.

Darin sind alle angelegten Musterkrankheiten mit ihren Zugehörigkeitsgraden zu bestimmten Symptomen aufgelistet. Die Symptome sind entsprechend ihren Zugehörigkeitsgraden sortiert. Das Symptom mit dem größten Zugehörigkeitsgrad steht an erster Stelle, links, und das Symptom mit dem kleinsten Zugehörigkeitsgrad steht an letzter Stelle, ganz rechts.

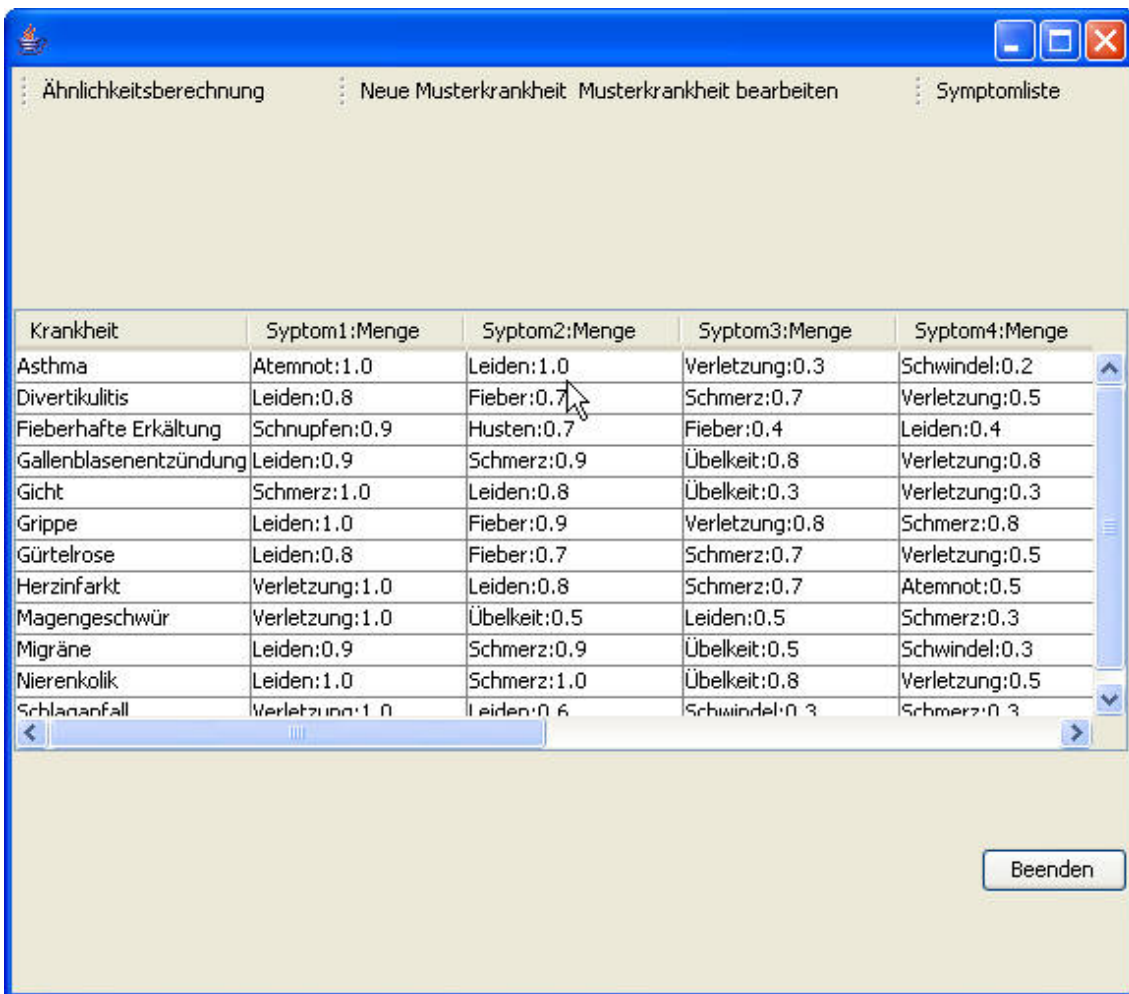


Abb. 10: Programm-Hauptmenü.

Des Weiteren verfügt das Hauptmenü über eine Tool-Bar.

Das Programm bietet dem Benutzer folgende Optionen:

1. Es kann Ähnlichkeiten berechnen. Hierzu klickt der Benutzer auf den Button „Ähnlichkeitsberechnung“ in der Tool-Bar.
2. Der Anwender kann eine neue Musterkrankheit eingeben und klickt hierzu auf den Button „Neue Musterkrankheit“
3. Er hat die Möglichkeit, bereits angelegte Musterkrankheiten zu bearbeiten. Um dieser Option nachzugehen, klickt der Benutzer in der Musterkrankheiten-Liste auf den Krankheitsnamen der zu ändernden Musterkrankheit und klickt auf den Button „Musterkrankheit bearbeiten“.
4. Über den Button Symptomliste können alle angelegten Symptome eingesehen und neue Symptome angelegt werden.

Im Folgenden soll nun die Option „Ähnlichkeitsberechnung“ detaillierter betrachtet werden.

7.5 Menüpunkt Ähnlichkeitsberechnung

Der Anwender hat in der folgenden Eingabemaske die Möglichkeit, zu bestimmten Symptomen die Zugehörigkeitsgrade anzugeben.

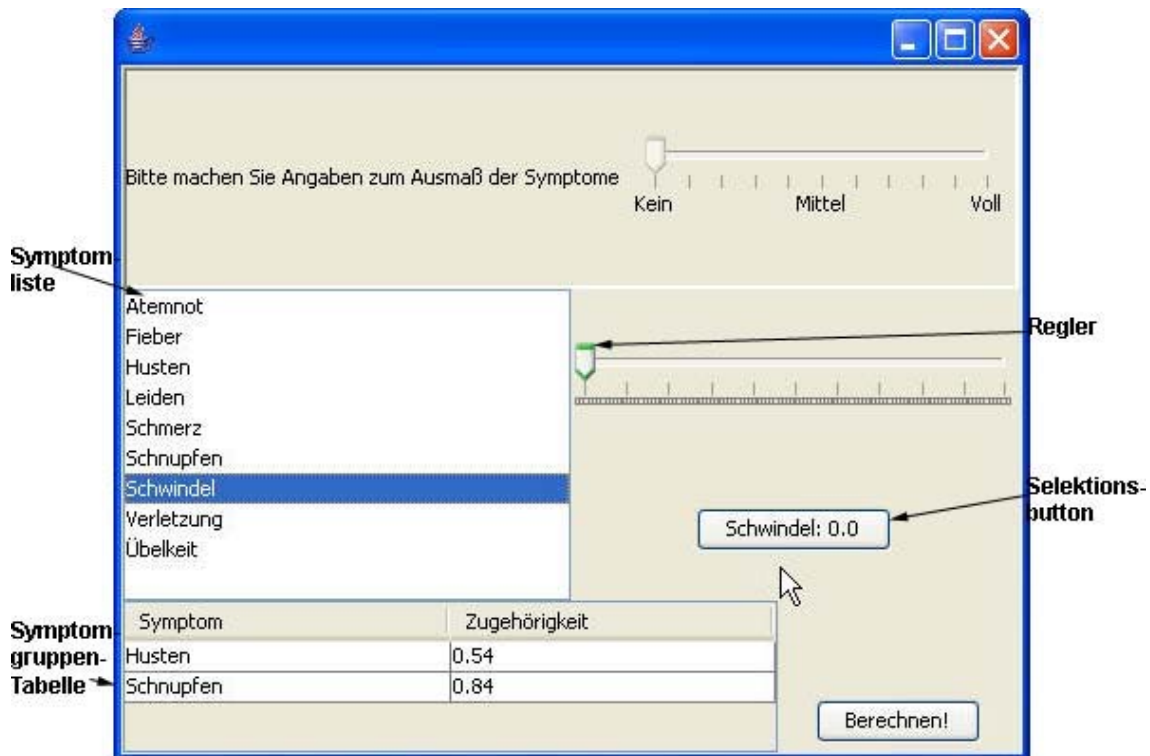


Abb. 11: Maske zur Ähnlichkeitsberechnung – Eingabe.

Alle in der Datenbank angelegten Symptome sind in einer Liste aufgeführt. Wünscht der Anwender den Zugehörigkeitsgrad zu einem bestimmten Symptom anzugeben, selektiert er dieses und gibt den Zugehörigkeitsgrad an, indem er den Regler bedient. Das selektierte Symptom ist mit dem momentan eingestellten Zugehörigkeitsgrad in einem Selektionsbutton ersichtlich. Sobald der Regler auf den korrekten Grad verweist, klickt der Benutzer auf den Selektionsbutton und das Symptom wird mit dem eingestellten Zugehörigkeitsgrad in die Symptomgruppen-Tabelle übernommen. Enthält diese Tabelle alle Symptome, zu denen der Anwender Angaben machen will, klickt er auf den Button ‚Berechnen!‘, und der Benutzer gelangt zu der Auswertungsmaske.

Die Ähnlichkeit der Symptomgruppe zu bestehenden Musterkrankheiten wird nun berechnet. Die Berechnung basiert auf der von Kazem Sadegh-Zadeh angeführten Formel für die partielle Ähnlichkeit zweier Symptomgruppen:

$$\text{similar}(A, B) = \frac{c(A \cap B)}{c(A \cup B)}$$

In der folgenden Auswertungsmaske sind in einer Ähnlichkeitstabelle nun alle Musterkrankheiten mit der jeweiligen Ähnlichkeit zur eingegebenen Symptomgruppe aufgeführt. Die Tabelleneinträge sind gemäß der Ähnlichkeit absteigend von oben nach unten sortiert. Die der Symptomgruppe ähnlichste Musterkrankheit steht also ganz oben, und der höchste Ähnlichkeitswert dient als Referenzwert.

Nach Sadegh-Zadeh legt die jeweilige Gesellschaft einen Grad ε fest, der bestimmt, inwiefern und ob man bei einer Symptomgruppe von einer Krankheit sprechen kann.

Dieser ε -Wert ist auf 0.6 voreingestellt und kann mittels des ε -Reglers verändert werden. Unterschreitet ε den Wert 0.5 wird der Anwender darauf hingewiesen, dass der gewählte Wert sehr klein ist und somit viele Symptomgruppen in ein Krankheitsmuster fallen würden. Er hat dennoch die Möglichkeit den ε -Wert beizubehalten, denn bekanntlich muss ε nur größer als 0.0 sein und kann höchstens einen Wert von 1.0 annehmen.

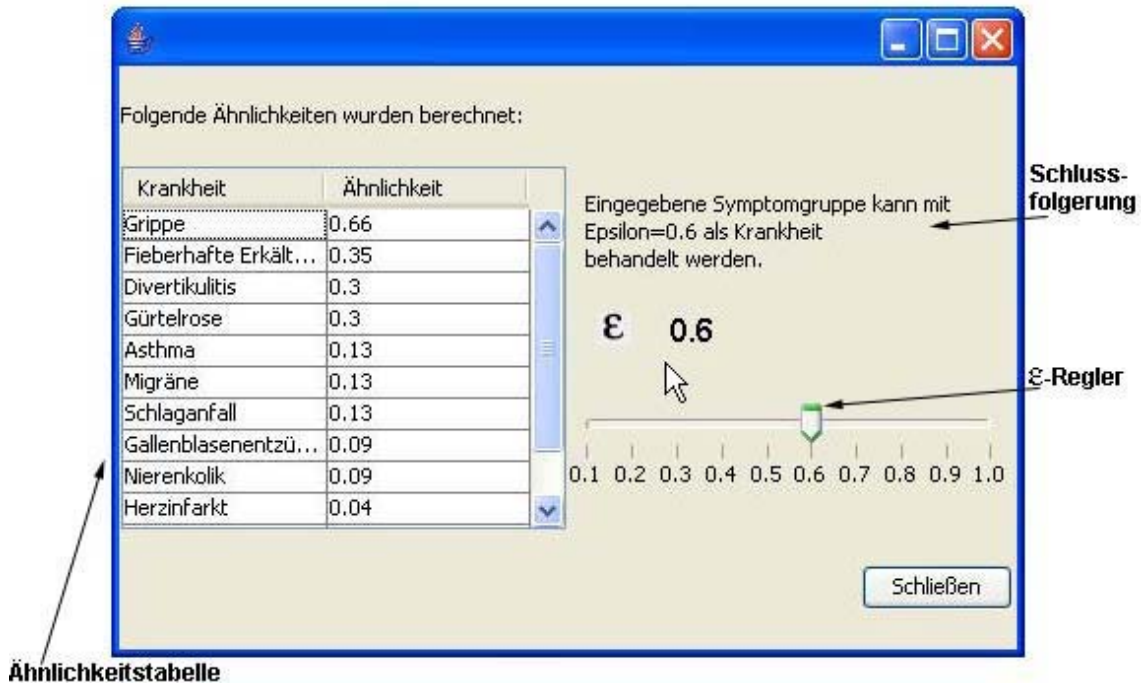


Abb. 12: Maske zur Ähnlichkeitsberechnung – Auswertung.

Der eingestellte ε -Wert wird mit dem Referenzwert, also dem höchsten Ähnlichkeitswert, verglichen. Wenn der ε -Wert größer als der Referenzwert ist oder ihm zumindest entspricht, kann daraus geschlossen werden, dass die eingegebene Symptomgruppe als Krankheit zum Grad ε behandelt werden kann.

Bei entsprechend großem Referenzwert könnte die Symptomgruppe als neue Musterkrankheit angelegt werden.

7.6 Diskussion und Ausblick

Das Programm soll Ansätze der Ideen des Autors umsetzen. Es ist auf diesem Stand keinesfalls repräsentativ, sondern soll lediglich die Idee der Umsetzung andeuten. Für eine stärkere Aussagekraft müssten Krankheiten feiner spezifiziert werden und mehr bzw. differenziertere Symptome hinzugefügt werden. Bewusst wurden auch Symptome miteinbezogen, die (relativ) eindeutig und objektiv zu bestimmen sind, wie z. B. Fieber. Dies geht über die Ansätze in Sadegh-Zadehs Artikel „Fuzzy Health, Illness and Disease“ hinaus, Krankheiten über ‚menschliche Beschaffenheiten‘ und vor allem psychische Kriterien zu definieren. Ein solcher Ansatz erfordert weitaus mehr Angaben zur Symptomatik der krankhaften Erscheinung als in diesem Programm möglich ist. Daher sind auch Aussagen zu möglichen Diagnosen mit Vorsicht zu genießen und vom Programm nicht vorgesehen.

8 Literatur

- [1] Kazem Sadegh-Zadeh: Fuzzy Health, Illness, and Disease. In: *The Journal of Medicine and Philosophy*, 25, 2000, S. 605-638.
- [2] Kazem Sadegh-Zadeh: *Organism and disease as fuzzy categories*, presented at the conference on Medicine and Philosophy, Humboldt University of Berlin, July 2, 1982.
- [3] Zadeh, Lotfi A.: Fuzzy Sets. *Information and Control*, 8, 1965, pp. 338-353.
- [4] Seising, Rudolf: 1965 - Zadeh Begins: „A New View on System Theory“. Ein Beitrag zum 40. Jubiläum der Fuzzy Set Theorie. In: R. Mikut, M. Reischl (Hrsg.), Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik Universität Karlsruhe (TH) Band 9: Proceedings 15. Workshop „Computational Intelligence“ des GMA-FA 5.14 „Computational Intelligence“ (Fachausschuss der VDI/VDE Gesellschaft für Mess- und Automatisierungstechnik) und der GI-FG „Fuzzy-Systeme und Soft-Computing“ (Fachgruppe der Gesellschaft für Informatik) 16-19. November 2005, Universitätsverlag Karlsruhe 2005, S. 1-19.
- [5] Rudolf Seising: *Die Fuzzifizierung der Systeme. Die Entstehung der Fuzzy Set Theorie und ihrer ersten Anwendungen - Ihre Entwicklung bis in die 70er Jahre des 20. Jahrhunderts*. Stuttgart: Franz Steiner Verlag (Boethius: Texte und Abhandlungen zur Geschichte der Mathematik und der Naturwissenschaften Band 54) 2005.
- [6] Zadeh, Lotfi A.: On the Analysis of Large-Scale Systems. Memorandum No. ERL-M418, Electronic Research Laboratory, College of Engineering, University of California, Berkeley, January 8, 1974.
- [7] Seising, Rudolf: Eine kleine Geschichte der Fuzzy-Systeme in der Medizintechnik. In: Tagungsband des 12. Workshop „Fuzzy Systeme“ des GMA-FA 5.22 „Fuzzy Control“ (Fachausschuss der VDI/VDE Gesellschaft für Mess- und Automatisierungstechnik) und der GI-FG „Fuzzy-Systeme und Soft-Computing“ (Fachgruppe der Gesellschaft für Informatik) am 14./15. November 2002, Gästehaus der Universität Dortmund „Haus Bommerholz“.
- [8] WHO. *Preamble to the Constitution of the World Health Organization as adopted by the International Health Conference*, New York, 19-22 June 1946, and entered into force on 7 April 1948.
Auszüge: http://www.who.int/bulletin/bulletin_board/83/ustun11051/en/
- [9] H.-R. Wicker: *Leitfaden für die Vorlesung Einführung in die Medizinethnologie*. www.anthro.unibe.ch/downloads/leitfaden_medethno.pdf

Fuzzy Sets als Werkzeug der Wissenschaftstheorie

Rudolf Seising

Medizinische Statistik und Informatik
Medizinische Universität Wien
Spitalgasse 23, A-1090 Wien, Österreich
Tel. (0043) 1 40400 6664
Fax (0043) 1 404006667
E-Mail: rudolf.seising@meduniwien.ac.at

1 Einführung

Über unsere Wissenschaft, ihr Funktionieren, ihre Entwicklung, ihre Bedingungen und ihre Folgen reflektieren Wissenschaftshistoriker und -philosophen. Wie sich wissenschaftliche Theorien bewähren oder ob sie scheitern, wie sie entstehen und sich in das schon bestehende System einpassen, wird von beiden Disziplinen – freilich mit unterschiedlichen Methoden – untersucht, denn dabei es geht um fundamentale Veränderungen im wissenschaftlich-technischen System. Die Entstehung der Fuzzy Set Theorie und ihre großen Erfolge lösten solche Veränderungen in Wissenschaft und Technik aus, dennoch wurde über diese Veränderung bisher kaum wissenschaftstheoretisch reflektiert.

In meinen wissenschaftshistorischen Untersuchungen zu diesem Thema [1] habe ich betont, dass die offensichtliche Kluft zwischen mathematisch formulierten Theorien und beobachtbaren realen Systemen Lotfi A. Zadeh, einen Elektrotechnikprofessor an der University of California in Berkeley, dazu veranlassten, über eine „neue Mathematik“ nachzudenken und diese schließlich auch zu etablieren, „the mathematics of cloudy or fuzzy quantities ...“ [2], die er 1965 in seinem Artikel „Fuzzy Sets“ in der Zeitschrift *Information and Control* erstmals einem größeren wissenschaftlichen Publikum vorstellte [3]. Diese „Kluft“ birgt ein tiefes philosophisches Problem der modernen Wissenschaftsphilosophie: Nachdem das reduktionistische Programm des Empirismus gescheitert ist, alle naturwissenschaftlichen Begriffe bzw. Sätze auf Begriffe bzw. Sätze von Beobachtbarem zurückzuführen, steht die Frage nach der Natur der theoretischen Größen oder Begriffe im Raum. Wissenschaftliches Arbeiten hat stets eine theoretische und eine empirische Komponente. Experimente werden durchgeführt, um Theorien oder Hypothesen zu bestätigen oder zu widerlegen. Einerseits werden somit theoretische Begriffe gebraucht, andererseits wird beobachtet und gemessen, werden Daten gesammelt. Passen diese Daten mit den theoretischen Gesetzen zusammen so ist die Theorie brauchbar. Wie wird dieser Spagat zwischen Empirie und Theorie realisiert? Welcher Art ist die Verbindung zwischen theoretischen Größen und Beobachtungstermen? Wie bringen wir in der Wissenschaft Theorie und Realität zusammen? In den 1960-er Jahren hatte Zadeh zur Überbrückung dieser Kluft die Fuzzy Sets bzw. die Fuzzifizierungen von Begriffen, Methoden und Theorien eingeführt [1]. Dieser Schritt soll im Folgenden wissenschaftstheoretisch nachvollzogen und reflektiert werden, und dazu bietet sich die so genannte strukturalistische Theorienauffassung an, deren Gerüst hier entsprechend modifiziert bzw. ergänzt wird: Die Theorie der Fuzzy Sets wird an dieser Stelle erstmals als Werkzeug in der Wissenschaftstheorie benutzt, um die oben genannte Kluft zwischen Erfahrung und Theorie zu überbrücken.

2 Entwicklung des wissenschaftstheoretischen Strukturalismus

Mit formalen Sprachen umzugehen ist selbst für Mathematiker und Informatiker oft schwierig und unbequem. Darauf zugunsten einer gefälligeren Darstellungsmöglichkeit zu verzichten ist daher stets willkommen, zumal wenn das gestellte Ziel auch auf informellem Weg erreicht werden kann.

In den 1930er Jahren begann eine französische Gruppe von Mathematikern, die sich das gemeinsame Pseudonym Nicolas Bourbaki gab, mit der Arbeit an einem Lehrbuch der Mathematik, den *Éléments de mathématique* [4], wobei sie motiviert von David Hilberts axiomatischer Darstellung der Mengenlehre ein Projekt mengentheoretischer Axiomatisierungen in der gesamten Mathematik verfolgte; die Darstellung in einer formalen Sprache erübrigte sich damit. Für jedes Teilgebiet der Mathematik sollte zwischen seinen Grundbegriffen, Definitionen sowie Axiomen und schließlich daraus abgeleiteten Theoremen unterschieden werden.

Die mengentheoretische „Axiomatisierung“ gelang durch Einführung eines jeweils mengentheoretischen Prädikats, etwa „ist ein Vektorraum“, „ist eine Topologie“, „ist eine Gruppe“ oder „ist eine euklidische Struktur“. Beispielsweise lässt sich die Gruppentheorie durch explizites Definieren des mengentheoretischen Prädikats „ist eine Gruppe“ axiomatisieren; dazu werden eine Menge G als undefiniertes Grundobjekt und eine Verknüpfungsrelation (z. B. $+$, \cdot) gebraucht, wobei für letztere in einem Axiom die Assoziativität gefordert wird, und ein zweites und drittes Axiom fordern die Existenz eines inversen bzw. eines neutralen Elements. Analog kann die euklidische Geometrie durch Definition des mengentheoretischen Prädikats „ist eine euklidische Struktur“ axiomatisiert werden, wobei es hierzu drei undefinierte Grundobjekte (*Punkte, Geraden, Ebenen*) und drei Relationen (*koinzidiert, liegt zwischen* und *ist kongruent mit*) gibt.

In den 1950er Jahren empfahl der Wissenschaftsphilosoph und Statistiker Patrick Suppes dieses Bourbaki-Verfahren zur Axiomatisierung mathematischer Theorien auch für die Naturwissenschaften zu übernehmen. Auch deren Theorien sollten durch die Angabe einer mathematischen Struktur charakterisiert werden. Suppes Vorschlag bedeutete somit, die Axiomatisierung naturwissenschaftlicher Theorien an den Standard der mathematischen Theorien heranzuführen. Die Aufgabe der informellen mengentheoretischen Axiomatisierung empirischer Theorien bestand nun darin, diese hinsichtlich ihrer Struktur so präzise zu beschreiben, dass sie ebenfalls durch ein mengentheoretisches Prädikat charakterisiert werden können; z. B. wird die klassische Partikelmechanik durch explizite Definition des Prädikats „ist eine Partikelmechanik“ axiomatisiert. [5]

Anders als die präzise konstruierten Theorien der Mathematik lagen viele der naturwissenschaftlichen Theorien damals allerdings erst in einer Form vor, die nicht als axiomatisiert bezeichnet werden kann. Außerdem – so ließen Kritiker verlauten – sei ein präziser Einbau physikalischer Theorien in das mengentheoretisch axiomatisierte System kaum realisierbar, da solche naturwissenschaftlichen Theorien zum einen mehr als nur eine mathematische Theorie darstellten und zum anderen gebe es hier Vagheiten zu berücksichtigen, die sie ja gerade der exakten Mathematisierung entzögen.

Erst Jahrzehnte später löste Joseph D. Sneed dieses Problem, indem er die Methode von Suppes durch eine informelle Semantik ergänzte, mit der neben den Beziehungen der mathematischen Strukturen untereinander auch jene zur Außenwelt berücksichtigt werden konnten [6]. Mit dem Suppes-Sneed-Programm des wissenschaftstheoretischen Strukturalismus (der damals allerdings noch nicht so genannte wurde), ließen sich nun

auch Vagheiten der naturwissenschaftlichen Theorien beschreiben. Im Folgenden werden wir sehen, dass gerade dieser Aspekt noch ausgebaut werden kann.

An die Phase der Entstehung des wissenschaftlichen Strukturalismus in den 1970er Jahren schloss sich eine Phase der Konsolidierung an, in der diese wissenschaftstheoretische Richtung systematisch ausgearbeitet wurde – neben Sneed sind hier vor allem Wolfgang Stegmüller, C. Ulises Moulines und Wolfgang Balzer zu nennen. Letztere beiden schrieben gemeinsam mit Sneed das 1987 erschienene Buch *An Architectonic for Science* mit dem ein gewisser Abschluss dieser Periode markiert wird [7].

3 Empirische und theoretische Strukturen

Wie arbeiten Wissenschaftler? – Sie beobachten reale Systeme oder Phänomene, sammeln Daten, stellen Gesetze auf, führen eine Theorie ein, die besagt, dass die Gesetze für diese Daten gelten und überprüfen sie. Die Daten werden an realen Objekten oder Systemen beobachtet bzw. gemessen, z. B. die Länge oder Masse eines Gegenstandes, die Spannung zwischen zwei Kondensatorplatten oder der Widerstand eines Verbrauchers in einem Stromkreis. Von diesen Systemen bzw. ihren Eigenschaften wird mit Hilfe von Begriffen abstrahiert, die in der Theorie eine definierte Bedeutung haben, und meistens die gleiche Bezeichnung wie die Daten tragen (z. B. Länge, Masse, Spannung, Stromstärke), zuweilen ist der Zusammenhang aber auch komplizierter (z. B. Masse und Volumen in der Relativitätstheorie oder die Zustandsfunktion in der Quantenmechanik, deren theoretische Größen nicht mit den in der Realität gemessenen Größen identisch sind, sondern als Relationen zwischen ihnen und dem Bezugssystem bzw. durch weitere mathematische Operationen aus ihnen berechnet werden). Diese theoretisch-begrifflichen Konstrukte fungieren als theoretische „Abbilder“ der realen Systeme, und diejenigen Strukturen aus theoretischen Begriffen und deren Relationen, die den aufgestellten Gesetzen genügen, werden im wissenschaftstheoretischen Strukturalismus als „Modelle“ der Theorie bezeichnet. Um die realen Systeme einer wissenschaftlichen Untersuchung zugänglich zu machen, bringt man sie also mit einer theoretischen Struktur in Verbindung, und dazu wird auch ihnen selbst eine Struktur zugeschrieben. „Wie dies genau geschieht, ist weithin unklar und darf wohl als eines der zentralen Probleme der Wissenschaftstheorie bezeichnet werden.“ schrieb der Wissenschaftstheoretiker Wolfgang Balzer 1982 in seinem einführenden Buch: „Das Problem besteht darin, einen Zusammenhang zwischen konkreten Systemen und „theoretischen“ Strukturen herzustellen. Wir nehmen im Folgenden an, daß ein solcher Zusammenhang hergestellt werden kann. Ohne diese Annahme hat es keinen Sinn, von empirischer Wissenschaft zu reden.“ ([8], S. 289) Solche mit einer theoretischen Struktur verbundenen Systeme der Realität werden im wissenschaftlichen Strukturalismus die „intendierten Systeme“ der Theorie genannt. Der hier noch vage bleibende Unterschied zwischen „realen“ und „intendierten“ Systemen einer Theorie wird im folgenden Abschnitt eingehender beleuchtet, und dabei werden wir uns vom klassischen wissenschaftstheoretischen Strukturalismus lösen müssen.

Ausgehend von einem realen bzw. intendierten System und erhält der Wissenschaftler einerseits eine Datenstruktur, andererseits erstellt er ein Modell, das die Struktur des Systems darstellen soll, oft wird in diesem Zusammenhang von einer „Abbildung der Realität durch Theorie“ gesprochen, wobei das theoretische Bild des realen Systems hier aus den Gesetzen der Theorie und den Daten aus dem System zusammengefügt wird. Dies ist allerdings eine sehr vereinfachte Ausdrucksweise für eine sehr komplexe „Abbildung“, wie die folgenden Abschnitte zeigen werden.

Nach dieser einleitenden Betrachtung soll zunächst ein von dem Wissenschaftsphilosophen Wolfgang Balzer in seinem Lehrbuch zur Wissenschaftstheorie von 1997 eingeführter „lokaler“ Theoriebegriff vorgestellt werden, der sich nur auf einen „zeitlichen Querschnitt durch den historischen Prozess, in dem die Theorie entsteht und sich entwickelt“ bezieht und daher globale Hintergründe ausblendet, die in der Wissenschaftsforschung aber ansonsten auch noch berücksichtigt werden ([9], S. 50):

„Eine wissenschaftliche Theorie in diesem lokalen Sinne besteht im wesentlichen aus vier Teilen:

- einer Klasse **M** von Modellen
- einer Klasse **I** intendierter Systeme
- einer Klasse **D** von Datenstrukturen
- einem Approximationsapparat **U**:

$$\mathbf{T} = \langle \mathbf{M}, \mathbf{I}, \mathbf{D}, \mathbf{U} \rangle.$$

Nachdem die Elemente der Klassen **M**, **I** und **D** schon kurz eingeführt wurden, wenden wir uns dem noch nicht erklärten Approximationsapparat **U** zu. Approximationen werden gebraucht, da die Gesetze von den Daten immer nur näherungsweise erfüllt bzw. umgekehrt die Daten von den Gesetzen immer nur näherungsweise erfasst werden. Dabei ist allerdings noch überhaupt nicht gesagt, was diese Annäherung bedeutet, zu welchem Grad sie die Daten und Gesetze einander annähert, wie gut sie also ist. Aus diesem Grunde braucht eine wissenschaftliche Theorie einen Approximationsapparat, durch den erklärt wird, was Approximation (Annäherung, Ähnlichkeit, Abstand etc.) bedeuten soll, und außerdem werden durch ihn bestimmte Abstände, Gütegrade, Signifikanzniveaus etc. festgelegt. Aus historischen Gründen ist sofort klar, dass in den meisten Fällen der Approximationsapparat einer wissenschaftlichen Theorie ein geeigneter statistischer oder topologischer Formalismus war bzw. ist.

Ohne den Approximationsapparat **U** näher zu spezifizieren, lässt sich sagen, dass er das Verhältnis zwischen den Daten **D**, die aus intendierten Systemen von **I** stammen und den Modellen **M** (und damit auch den Gesetzen) repräsentiert; **U** ist also dafür verantwortlich dass und wie gut die Daten und Modelle zueinander passen, oder auch nicht, denn es muss auch Daten geben, die nicht zu den Modellen einer Theorie passen, somit ist es auch möglich, dass eine Theorie durch bestimmte Daten bzw. Beobachtungen widerlegt wird. Die vier Grundelemente einer Theorie sind also auf bestimmte Weise miteinander verknüpft, die Balzer in zwei fundamentalen Axiomen formuliert: Eine Struktur $\mathbf{T} = \langle \mathbf{M}, \mathbf{I}, \mathbf{D}, \mathbf{U} \rangle$ soll Theorie heißen, wenn gilt:

Axiom 1 (Passungsbedingung)

Modelle **M** und Daten **D** müssen zueinander passen.

Axiom 2 (Widerlegbarkeitsbedingung)

Es muss „mögliche“ Daten geben, zu denen **M** nicht passt.

Der bisher vorgestellte Theoriebegriff geht davon aus, dass intendierte Systeme und Daten schon zur weiteren wissenschaftlichen Verarbeitung vorliegen, damit wurde allerdings ein großer Anteil wissenschaftlicher Abstraktion übersprungen, denn weder ist erörtert worden wie die Daten erhoben werden, noch ist klar, wie die intendierten Systeme zu fassen sind. Balzer schreibt dazu:

„Methoden der Datenfindung können beliebig komplex sein und brauchen der Komplexität von Gesetzen um nicht nachzustehen. Insbesondere werden Daten in der Regel nicht direkt durch die Sinne wahrgenommen. Die Einigung über Daten erfolgt in anderen Formen als die über Gesetze, aber diese Formen geben für sich keinen Hinweis darauf, dass Daten ein objektives Fundament der Erkenntnis bilden. Für die korrekte Einschätzung unseres Theoriebegriffs ist es jedenfalls wichtig, die reiche Vorgeschichte von Daten und deren hypothetischen Charakter nicht aus den Augen zu verlieren. Es ist nicht Aufgabe der Wissenschaftstheorie, zu beweisen, dass wissenschaftliche Theorien absolut sicher sind, weil sie absolute sichere Beobachtungen systematisieren; diese Aufgabe können wir getrost halbreligiösen Wahrheitssuchern überlassen.“ ([9], S. 50f)

Zur „reichen Vorgeschichte“ der Daten gehören die intendierten Systeme, die nicht mit den realen Systemen identisch sind. Zu ihrer Unterscheidung bzw. zur Problematik, eine Verbindung von den realen zu den intendierten Systemen zu finden, schreibt Balzer schon 1982: „Auf der einen Seite der zu klärenden Verbindung steht ein „unstrukturierter“ System, ein Teil der Welt, den wir auf irgendeine Weise zum Gegenstand unseres Interesses, unserer Neugier gemacht haben. Auf der anderen Seite steht ein von uns mit unseren Begriffen „durchdrungenes“ System, d.h. das ursprüngliche System wird von uns „durch die Brille unserer Begriffe“ gesehen. Während wir z. B. vorher nur ein Karussell sahen, haben wir nach der Herstellung der fraglichen Verbindung ein System von Partikeln vor uns, die sich auf bestimmten Bahnen bewegen.“ ([8], S. 289.)

Die „reiche Vorgeschichte von Daten“, wie ein Wissenschaftler Phänomene an realen Systemen wahrnimmt, wie er zu intendierten Systemen gelangt, soll hier nicht nur nicht aus den Augen verloren werden, sondern das Thema der folgenden Diskussion sein. So wie ein Approximationsapparat **U** den Übergang von den Daten zu den Modellen charakterisiert, so soll ein weiterer hier einzuführender Apparat **F** den Übergang von den intendierten Systemen zu den Daten repräsentieren. Die intendierten Systeme werden mit den Wahrnehmungen identifiziert, die der Wissenschaftler von den realen Systemen erhält. Es soll hier auch nicht etwa bewiesen werden, dass es sich dabei um einen absolut sicheren Übergang handelt; vielmehr wird betont, dass dieser Übergang die Unschärfen von Wahrnehmungen mittels der Konzepte der Fuzzy Set Theorie erfasst.

Der Übergang von den realen zu den intendierten Systemen entspricht dem Blick „durch die Brille unserer Begriffe“ und unsere Begriffe sind sprachlich gefasst. Wir formulieren unsere Wahrnehmungen in Worten und Sätzen. Die „Vorgeschichte“ der Daten, das von einem wissenschaftlichen Beobachter an realen Systemen Wahrgenommene, die intendierten Systeme sollen daher linguistisch ausgedrückt und mathematisch durch Fuzzy Sets und Fuzzy-Relationen dargestellt werden. Der Übergang von den realen zu den intendierten Systemen entspricht insofern einer Fuzzifizierung, während der von den die intendierten Systeme darstellenden Fuzzy Sets und Fuzzy-Relationen zum Datenmaterial der Theorie eine „Defuzzifizierung“ ist – von den unscharfen Wahrnehmungen zu den scharfen Daten.

Der hier einzuführende Apparat **F** für den Übergang von den intendierter Systemen zu den Daten ist deutlich verschieden vom Approximationsapparat **U** zwischen den Daten und den Modellen. Es ist also klar, dass Fuzzy Set Theorie und Statistik (oder Topologie) in dieser wissenschaftstheoretischen Theorie völlig verschiedene Rollen spielen.

Das hier vorgestellte Verfahren ist offenkundig völlig neu und von dem von Balzer als unwissenschaftlich abgelehnten Vorgehen zu unterscheiden, daher gerät die Theorie der Fuzzy Sets auf diese Weise auch keineswegs in den Dunstkreis von halbreligiösen Wahrheitssuchern.

4 Wahrnehmungen und Fuzziness

Analog zum Approximationsapparat **U**, der das Verhältnis zwischen den Daten und den Modellen einer Theorie repräsentiert, wird ein Theoretisierungsapparat **F** eingeführt, der das Verhältnis zwischen den realen Systemen, den Wahrnehmungen (*perceptions*) eines Beobachters, die wir mir den intendierten Systemen **I** identifiziert haben, und den daraus extrahierten Daten regelt, aus denen dann unter Hinzunahme der Gesetze eine wissenschaftliche Theorie entsteht. Wir konstruieren also zwischen die „Ebene“ der realen Systeme und die „Ebene“ der Daten und theoretischen Modelle eine „Ebene“ der intendierten Systeme bzw. der Wahrnehmungen des Wissenschaftlers, so dass der Übergang von der Realität zur wissenschaftlichen Theorie in zwei Teilübergängen nachvollzogen werden kann: von den realen Systemen und Phänomenen zu den intendierten Systemen und von diesen zu den Daten und Modellen. Die intendierten Systeme bzw. Wahrnehmungen werden durch Fuzzy Sets dargestellt, während die Daten numerisch scharfe Werte sind. Der Apparat **F** besteht folglich aus zwei Teilapparaten: einer „Fuzzifizierung“ der intendierten Systeme bzw. Phänomene zu den Wahrnehmungen und einer „Defuzzifizierung“ dieser intendierten Systeme bzw. Wahrnehmungen zu Daten bzw. Modellen.

Schon seit den letzten Jahrzehnten im 20. Jahrhundert verknüpft der Berkeley-Professor Lotfi Zadeh die von ihm in den 1960er Jahren eingeführten Fuzzy Sets [1, 10-12] mit einer Theorie der menschlichen Wahrnehmungen. In seinem Artikel “From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions” propagierte er eine *Computational Theory of Perceptions* (kurz CTP) [13].

In dieser Theorie spielen Wörter die Rolle von Bezeichnungen für Wahrnehmungen. Menschen drücken ihre Wahrnehmungen mittels der Worte und Sätze ihrer natürlichen Sprache aus. Mit der CTP beabsichtigt Zadeh eine Alternative zum üblichen Rechnen mit Zahlen für jene Fälle zu etablieren, in denen die einem Beobachter zugänglichen Informationen über ein reales System zu ungenau sind, als dass sie in Zahlen ausgedrückt werden könnten, oder für die ein Toleranzbereich für Ungenauigkeiten genutzt werden kann, um größere Robustheit oder leichtere Lenkbarkeit zu erreichen. Dabei ließ er sich – wie schon in den 1960er Jahren bei der Begründung der Fuzzy Set Theorie – von bemerkenswerten menschlichen Fähigkeiten inspirieren, nämlich von der “remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are parking a car, playing golf, deciphering sloppy handwriting and summarizing a story. Underlying this capability is the brain's crucial ability to reason with perceptions – perceptions of time, distance, speed, force, direction, shape, intent, likelihood, truth and other attributes of physical and mental objects.” ([14], p. 903).

Messungen sind scharf, Wahrnehmungen sind fuzzy und aufgrund endlicher Auflösungsvermögen unserer Sinnesorgane können wir nicht beliebig nahe Punkte voneinander unterscheiden, Wahrnehmungen sind also auch körnig, oder wie Zadeh im Jahre 2001 im *AI Magazine* schrieb: „perceptions, in general, are both fuzzy and granular or, for short *f*-granular. For example, a perception of age can be described as very young, young, middle aged, old, and very old, with very young, young, and so on, constituting the granules of the variable *age*. In this perspective, natural languages can be viewed as systems whose primary function is to describe perceptions.” ([15], S. 74)

Schon 1973 hatte Zadeh in dem Artikel “Outline of a New Approach to the Analysis of Complex Systems and Decision Processes” so genannte „linguistische Variable“ einge-

führt (Abb. 1), um mit Worten anstatt mit Zahlen rechnen zu können [16] und in dem 1996 erschienenen Artikel “Fuzzy Logic = Computing with Words” betonte er dann gemäß der Artikelüberschrift: “the main contribution of fuzzy logic is a methodology for computing with words. No other methodology serves this purpose.” ([17], p. 103.)

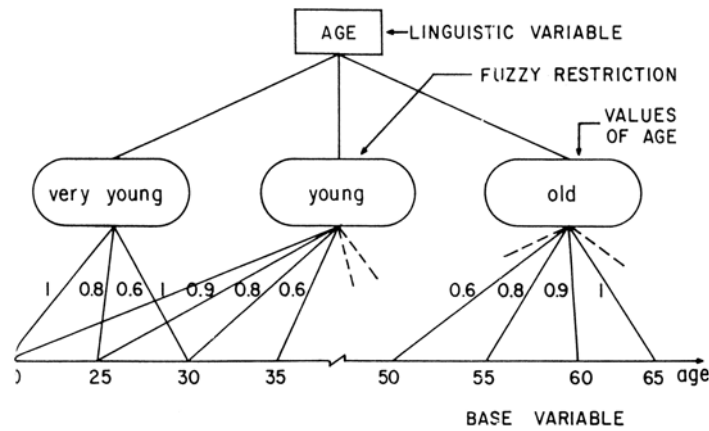


Abb. 1: Zadehs linguistische Variable *Alter (age)* [16].

Abb. 2 zeigt Zadehs Gegenüberstellung der scharfen (*crisp*, *C*) und der unscharfen (*fuzzy*, *F*) Körnung der linguistischen Variable *age* aus einem Artikel des Jahres 2001.

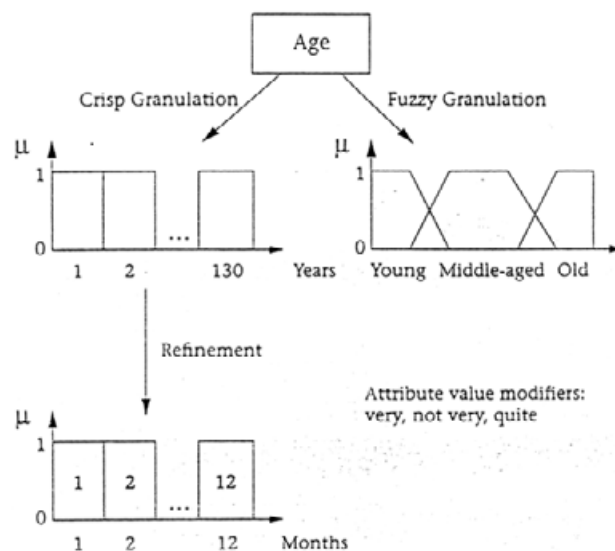


Abb. 2: F-Granulation and C-Granulation of *Age*. [17].

Fuzzy Sets and Fuzzy Logic sind die Basismethodologien für das Computing with Words (CW), und CW liegt der „Computational Theory of Perceptions“ (CTP) zugrunde, so dass insgesamt eine Hierarchie von Methodologien vorliegt (Abbildung 3).

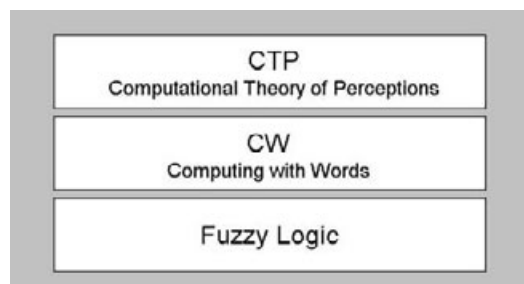


Abb. 3: Zadehs Methodologienhierarchie.

Zadeh ist davon überzeugt, dass diese Methoden zukünftig einen bleibenden und erfolgreichen Platz unter den wissenschaftlich-technologischen Methoden haben werden: “In coming years, computing with words and perceptions is likely to emerge as an important direction in science and technology.” [15]

Hier soll die CTP nicht eingehender erläutert, wohl aber betont werden, dass diese Theorie die menschlichen Wahrnehmungen als einen wichtigen Bestandteil wissenschaftlicher Erkenntnis hervorhebt. Dies ist ein wissenschaftshistorisch und -philosophisch bedeutsamer Punkt, denn mit den im 20. Jahrhundert neu entstandenen physikalischen Theorien der Relativitätstheorie und der Quantenmechanik wurde die Rolle des Beobachters und seiner Wahrnehmungen beim wissenschaftlichen Erkenntnis- wie auch beim Theoretisierungsprozess zentral.

Viele traditionelle wissenschaftstheoretische Auffassungen differenzieren allerdings nicht genügend deutlich zwischen Realität, Wahrnehmung und Theorie, daher sollen die Wahrnehmungen hier an die *Computational Theory of Perceptions* anschließend in das Gerüst des wissenschaftstheoretischen Strukturalismus eingebaut werden, das bisher zwar empirische und theoretische Strukturen repräsentiert, nicht aber die Wahrnehmungen eines Wissenschaftlers bei seinen Beobachtungen. Wir ergänzen daher die in Abschnitt 3 von Balzer zitierte Definition einer wissenschaftlichen Theorie:

Eine wissenschaftliche Theorie besteht im Wesentlichen aus fünf Teilen:

- einer Klasse **M** von Modellen
- einer Klasse **I** intendierter Systeme
- einer Klasse **D** von Datenstrukturen
- einem Approximationsapparat **U**
- einen Theoretisierungsapparat **F**:

$$\mathbf{T} = \langle \mathbf{M}, \mathbf{I}, \mathbf{D}, \mathbf{U}, \mathbf{F} \rangle.$$

Wie schon oben gesagt wurde, soll die Beziehung zwischen den realen Systemen und Phänomenen (reale Ebene) und den konstruierten Daten und Modellen (theoretische Ebene) in zwei Beziehungen geteilt werden, indem zwischen diese Ebenen eine Ebene der Wahrnehmungen (Fuzzy-Ebene) eingeschoben wird. Derjenige Teil des Apparates **F**, der von der realen Ebene zur Fuzzy-Ebene führt, erhält den Namen *Fuzzifizierung*, der von der Fuzzy-Ebene zur theoretischen Ebene weiterführende Teil wird *Defuzzifizierung* genannt.

4.1 Fuzzifizierung

Ein Wissenschaftler beobachtet die intendierten Systeme und Phänomene in der Natur oder im Experiment. Beispielsweise beobachtete Isaac Newton auf dem Wege zur Theorie der Mechanik die Systeme „Sonne – Erde“, „Erde – Mond“, das ganze Sonnensystem, aber auch frei fallende Körper und die Gezeiten. Die Sinnesorgane verhelfen dem Forscher dabei zu den Wahrnehmungen, aufgrund derer er schließlich seine Beobachtungsdaten zusammenstellt, Gesetzmäßigkeiten findet und schließlich eine Theorie formuliert. Im Gegensatz zu diesen Daten, die als Zahlenwerte von zuvor festgelegten Messgrößen fungieren, sind die Wahrnehmungen des Wissenschaftlers noch unscharf und grobkörnig. Wie oben angekündigt, sollen zur Repräsentation solcher Wahrnehmungen Fuzzy Sets dienen, etwa A^F , B^F , C^F , ... Auch kann es sein, dass der Wissenschaftler nicht nur Einzelheiten sondern miteinander in Beziehung stehende Phänomene

wahrnimmt, z. B. bewegt sich etwas gleichsinnig oder gegensinnig, schneller bzw. langsamer als etwas anderes, ist es heller bzw. dunkler, oder es riecht ähnlich usw. Solche Beziehungen sollen als *Fuzzy-Relationen* f^F, g^F, h^F, \dots charakterisiert werden. Die hier beschriebene „Fuzzifizierung“ beobachteter Phänomene wird in Abbildung 4 dargestellt.

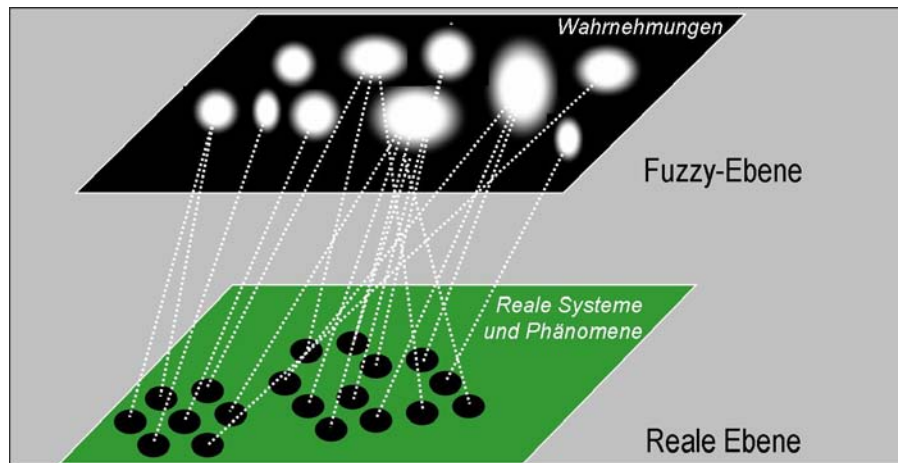


Abb. 4: Fuzzifizierung der realen Systeme und Phänomene zu Wahrnehmungen.

4.2 Defuzzifizierung

Wissenschaftler, die mit der Theorie der Fuzzy Sets vertraut sind, können auch mit den Entitäten der Fuzzy-Ebene arbeiten, Vertreter der exakten Wissenschaften sind allerdings darauf angewiesen, mit Entitäten der gewöhnlichen Mathematik zu operieren, und dazu wurden in den letzten Jahrhunderten Theorien der exakten Wissenschaften entwickelt, für die aus den ungenauen Wahrnehmungen des Beobachters gewisse theoretische Größen abgeleitet wurden, deren Werte im Experiment gemessen werden und die zur Grundlage für Axiomensysteme und Berechnungen der Theorie wurden. Entsprechend dem Galileo Galilei zugeschriebenen Satz: „Alles messen, was messbar ist – und messbar machen, was noch nicht messbar ist.“ ging man bei der Entwicklung moderner wissenschaftlicher Theorien von den Wahrnehmungen zu Messgrößen über. Dieser Übergang entspricht einer Defuzzifizierung. Von den die Wahrnehmungen und ihre Beziehungen darstellenden Fuzzy Sets A^F, B^F, C^F, \dots und Fuzzy-Relationen f^F, g^F, h^F, \dots gelangt man zu gewöhnlichen Mengen A^C, B^C, C^C, \dots und Relationen f^C, g^C, h^C, \dots . Aus diesen Mengen und Relationen lassen sich auf der theoretischen Ebene Daten und Modelle wissenschaftlicher Theorien konstruieren.

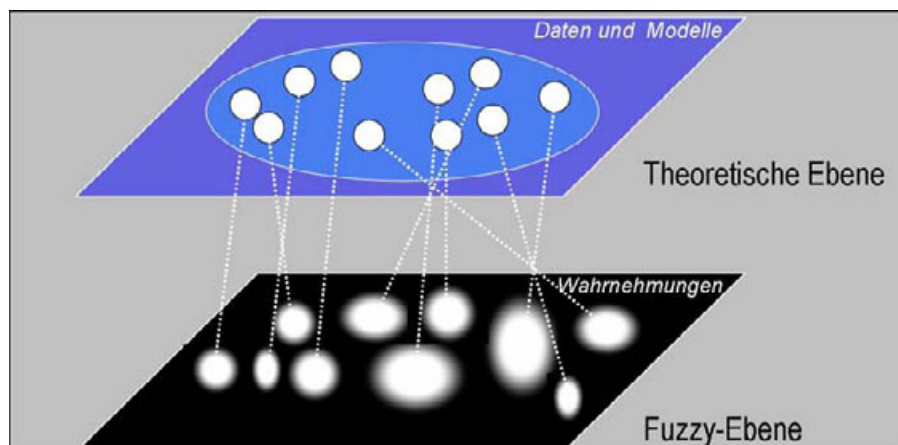


Abb. 5: Defuzzifizierung der Wahrnehmungen zu Daten und Modellen.

4.3 Fuzzifizierung/Defuzzifizierung und Theoretisierung

Die in den letzten beiden Abschnitten vorgestellten Teilprozesse der Fuzzifizierung und Defuzzifizierung bilden hintereinander kombiniert den Apparat **F**, der mit einigem Recht als Theoretisierungsapparat bezeichnet werden kann, denn er überführt die Phänomene auf der empirischen Ebene in Strukturen auf der theoretischen Ebene (Abb. 6).

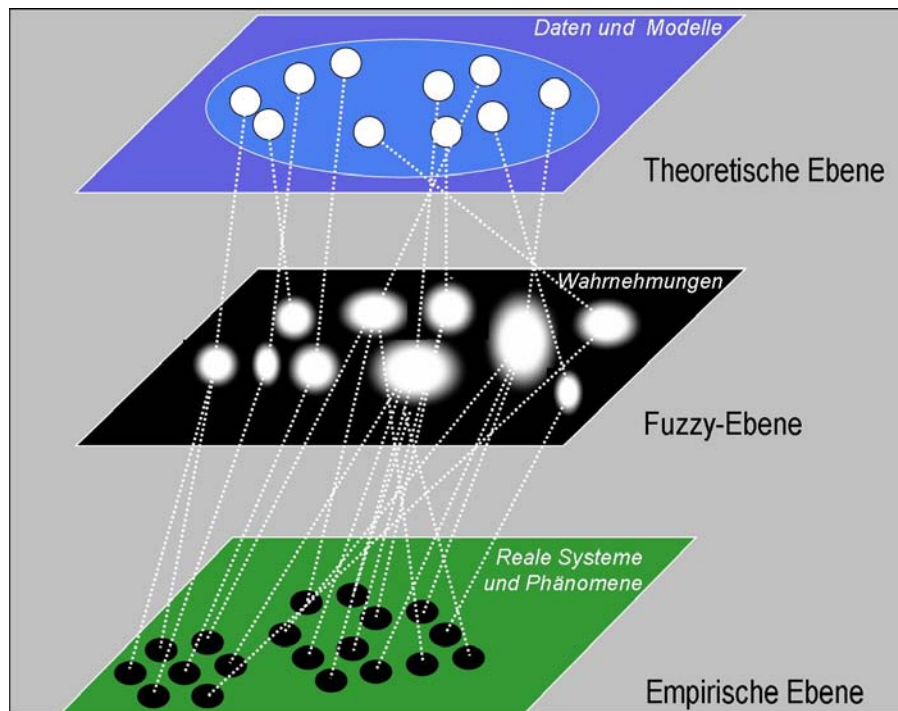


Abb. 6: Der Theoretisierungsapparat F als Kombination von Fuzzifizierung und Defuzzifizierung.

Der Begriff der Theoretisierung ist im wissenschaftstheoretischen Strukturalismus allerdings schon für eine intertheoretische Relation reserviert – eine Beziehung zwischen Theorien **T** und **T'**. Zwischen diesen Theorien besteht diese Theoretisierungsrelation, wenn die Theorie **T'** aus der Theorie **T** durch Hinzunahme neuer Grundgrößen und Einführung neuer Gesetze hervorgeht, wobei die neuen Gesetze die (alten) theoretischen Größen von **T** mit den (neuen) theoretischen Größen von **T'** verknüpfen.

Durch sukzessives Hinzufügen neuer theoretischer Größen kann somit eine Theorienhierarchie errichtet, und ein komparativer Theoretizitätsbegriff eingeführt werden. Auf diese Weise entsteht beispielsweise die *Raum-Zeit-Theorie* aus der (*Euklidischen*) *Geometrie* wenn man die „Zeit“ der „Länge“ als neue Größe zur Seite stellt, und aus dieser klassischen *Raum-Zeit-Theorie* entsteht durch Hinzunahme der Größe „Geschwindigkeit“ die klassische *Kinematik*. Aus dieser Theorie wiederum erhält man die klassische (Newtonsche) *Mechanik*, wenn man die Größen „Kraft“ und „Masse“ als neue theoretische Größen einführt.

Die alte Theorie **T** wird von der neuen Theorie **T'** sozusagen mit einer neuen „theoretischen Schicht“ überdeckt; **T**-theoretische Grundbegriffe, sind nicht **T'**-theoretisch, sondern nunmehr nicht-theoretisch. Umgekehrt dürfen keine der schon in **T** nicht-theoretischen Begriffe eine Hierarchiestufe höher **T'**-theoretisch sein! Die alte Theorie darf von der neuen Theorie also in keiner Weise verändert werden.

In dieser Theorienhierarchie gilt: Je „höher“ man in der Hierarchie steigt, desto theoretischer sind die Begriffe, während die niedrigeren Schichten das nicht-theoretische Fundament charakterisieren.

Was passiert aber in der untersten Ebene dieser Theorienhierarchie? – Es gibt hier eine Theorie **T** mit theoretischen Größen und Relationen, aber sie ist nicht Theoretisierung einer anderen Theorie. Vielmehr überzieht sie Phänomene der empirischen Ebene erstmals mit **T**-theoretischen Größen. Diese ist sozusagen eine *initiale* Theoretisierung, denn die **T**-theoretischen Begriffe sind die einzigen theoretischen Begriffe. Sie wurden direkt aus beobachteten Phänomenen als Messgrößen abgeleitet. Diese Ableitung haben wir im letzten Abschnitt als Operation des Apparates **F** eingeführt, der sich aus Fuzzifizierung und Defuzzifizierung kombinieren lässt. Insofern lässt sich die Operation des Apparates **F** als „initiale“ Theoretisierungsrelation einer wissenschaftlichen Theorie **T** und der empirischen Ebene der Phänomene charakterisieren.

5 Strukturen wissenschaftlicher Theorien

Der wissenschaftstheoretische Strukturalismus zeichnet ein sehr differenziertes Bild von Strukturen wissenschaftlicher Theorien, ihren Eigenschaften und Beziehungen untereinander. In Abschnitt 3 wurden schon die Strukturklassen **D** (Datenstrukturen), **M** (Modelle) und **I** (intendierte Systeme) vorgestellt, wobei die beiden ersten der theoretischen Ebene angehören, während **I** in der darunter liegenden nicht-theoretischen Ebene liegt. Balzer hatte diese Klasse der intendierten Systeme der Realität zugeordnet (das Bild von den Ebenen hat er nicht benutzt), wenn auch mit dem Anspruch: „Möglichst wenig Struktur für **I**“! ([8], S. 291) Wir haben im letzten Abschnitt dafür argumentiert, **I** in eine zwischen der realen und der theoretischen Ebene liegende Fuzzy-Ebene einzubetten. Die Klasse **I** enthält demnach keine realen Systeme sondern die Wahrnehmungen eines Beobachters realer Systeme, die wir als Fuzzy-Strukturen identifiziert haben.

Im wissenschaftlichen Strukturalismus werden noch weitere Strukturklassen definiert, um den Theoriebegriff mengentheoretisch erfassen zu können: die Klasse **M_p** der potentielle Modelle, die Klasse **M_{pp}**, die Klasse **Q** der Querverbindungen und schließlich wird auch die Klasse **M** der Modelle einer Theorie präzisiert. In diesem Abschnitt sollen diese Strukturen eingeführt und mit der im letzten Abschnitt modifizierten Ebenen-Hierarchie in Einklang gebracht werden. Dazu definieren wir die Klassen **M_p**, **M** und **M_{pp}**, wobei **M_p** und **M** – gemäß dem klassischen wissenschaftlichen Strukturalismus – Klassen herkömmlicher mengentheoretischer Strukturen sind, während **M_{pp}** – vom klassischen wissenschaftstheoretischen Strukturalismus abweichend – eine Klasse von Fuzzy-Strukturen bildet.

Strukturen bestehen aus nichtleeren Mengen D_1, \dots, D_k und Relationen f_1, \dots, f_m über D_1, \dots, D_k , das sind Teilmengen des kartesischen Produkts irgendwelcher der Mengen D_i . Fuzzy-Strukturen bestehen demgegenüber aus Fuzzy Sets D_1^F, \dots, D_k^F und Fuzzy-Relationen f_1^F, \dots, f_m^F . Insbesondere sollen die im Folgenden betrachteten (Fuzzy-) Relationen "von gleichem Typ" sein, das heißt, die bei den kartesischen Produkten benutzten Indizes sind gleich und sie treten in der gleichen Reihenfolge auf.¹ Von Relationen glei-

¹ D_1, \dots, D_k und D_1', \dots, D_n' seien Mengen, $k \leq n$. f sei Relation über D_1, \dots, D_k und f' sei Relation über D_1', \dots, D_n' . f und f' sind von gleichem Typ $\Leftrightarrow \exists i_1, \dots, i_m \in \{1, \dots, k\}$, so dass gilt: $f \subseteq \prod_{j \leq n} D_{i_j}$ und $f' \subseteq \prod_{j \leq m} D'_{i_j}$. Das Analoge gilt für Fuzzy-Relationen.

chen Typs gelangt man zu *Strukturen gleichen Typs*, wenn sie die gleichen Anzahlen an (Fuzzy) Mengen und (Fuzzy-)Relationen haben, und letztere von gleichem Typ sind.²

Die Menge \mathbf{M}_p der potentiellen Modelle einer Theorie wird nun als eine Menge von Strukturen von gleichem Typ definiert:

Definition (*potentielle Modelle*)

\mathbf{M}_p ist eine Klasse *potentieller Modelle* für eine Theorie gdw.

- a) \mathbf{M}_p ist eine Klasse von Strukturen.
- b) Alle Strukturen in \mathbf{M}_p sind von gleichem Typ.

Die Modelle für eine Theorie \mathbf{T} bilden eine Teilmenge der potentiellen Modelle für \mathbf{T} :

Definition (*Modelle*)

\mathbf{M} ist eine Klasse von Modellen für \mathbf{M}_p gdw.

- a) \mathbf{M}_p ist eine Klasse potentieller Modelle für eine Theorie.
- b) $\mathbf{M} \subseteq \mathbf{M}_p$.

Im konkreten Fall wird anzugeben sein, inwiefern nicht alle potentiellen Modelle aus \mathbf{M}_p auch zu \mathbf{M} gehören. Die einschränkenden Bedingungen sind die Gesetze der Theorie \mathbf{T} , in denen die Komponenten der potentiellen Modelle durch Relationen miteinander verknüpft werden. Nur wenn diese Gesetze von einem potentiellen Modell erfüllt werden, ist es auch ein Modell der Theorie. Die Modelle der klassischen Partikelmechanik müssen z. B. das Newton'sche Kraftgesetz erfüllen, die Modelle der Quantenmechanik die Schrödingergleichung usw.

Potentielle Modelle einer Theorie \mathbf{T} enthalten insbesondere solche Mengen und Relationen, deren Werte in einer von dieser Theorie abhängigen Weise beobachtet und gemessen werden, das sind die \mathbf{T} -theoretischen Größen. Wenn alle diese \mathbf{T} -theoretischen Größen aus den potentiellen Modellen der Theorie entfernt werden, dann bleiben Strukturen aus Mengen und Beziehungen zurück, die unabhängig von \mathbf{T} beobachtbar und messbar sind. Diese Strukturen werden im wissenschaftlichen Strukturalismus zur Klasse \mathbf{M}_{pp} der partiell potentiellen Modelle für die Theorie \mathbf{T} zusammengefasst, und insbesondere sind die intendierten Systeme darin enthalten, es gilt nämlich: $\mathbf{I} \subseteq \mathbf{M}_{pp}$.

Wie schon angekündigt, soll an dieser Stelle vom klassischen wissenschaftstheoretischen Strukturalismus abgewichen werden. Die notwendige Modifikation wird mit der Forderung, dass \mathbf{I} Untermenge von \mathbf{M}_{pp} ist unmittelbar klar. Da wir die Menge der intendierten Systeme mit den Wahrnehmungen eines wissenschaftlichen Beobachters identifiziert und diese als Fuzzy Sets definiert haben, müssen auch die partiell potentiellen Modelle in \mathbf{M}_{pp} Fuzzy Sets enthalten. Wir fordern also nicht – wie im klassischen wissenschaftstheoretischen Strukturalismus –, dass alle \mathbf{T} -theoretischen Größen aus den potentiellen Modellen der Theorie entfernt werden, sondern dass sie fuzzifiziert werden. Sind die \mathbf{T} -theoretischen Größen etwa die Mengen D_1, \dots, D_k und die Relationen f_1, \dots, f_m über D_1, \dots, D_k , so wird von ihnen zu den Fuzzy Sets D_1^F, \dots, D_k^F und Fuzzy-

² $x = \langle D_1, \dots, D_k; f_1, \dots, f_m \rangle$ und $x' = \langle D_1', \dots, D_n'; f_1', \dots, f_s' \rangle$ seien zwei Strukturen. x und x' sind vom gleichen Typ \Leftrightarrow a) $k = n$ und $m = s$, und b) $\forall i \leq m$ gilt: f_i ist vom gleichen Typ wie f_i' .

eingeführt wurden. Der erste Teilprozess strukturiert die realen Systeme unscharf, der zweite Teil schärft diese Strukturierung durch theoretische Begriffe. Der Übergang von den realen zu den intendierten Systeme setzt damit nur so viel (unscharfe) Struktur zu ihrer Darstellung voraus, wie unbedingt nötig ist. Insbesondere werden dazu keine theoretischen Größen benutzt, so dass auf der empirischen Ebene nur nicht-theoretische Größen existieren. Abb. 7 soll diese Zusammenhänge illustrieren.

Der um die Klasse **I** der intendierten Systeme erweiterte Theorie-Kern stellt im wissenschaftstheoretischen Strukturalismus die Theorie **T** selbst dar:

Definition (*empirische Theorie*)

T ist eine empirische Theorie gdw. es einen Kern **K** und eine Menge **I** der intendierten Systeme gibt, sodass

- a) $\mathbf{T} = \langle \mathbf{K}, \mathbf{I} \rangle$
- b) **K** ist ein Kern für eine Theorie.
- c) **I** passt zu **K**.
- d) Die Strukturen in **I** erfassen die intendierten Anwendungen für **K**.

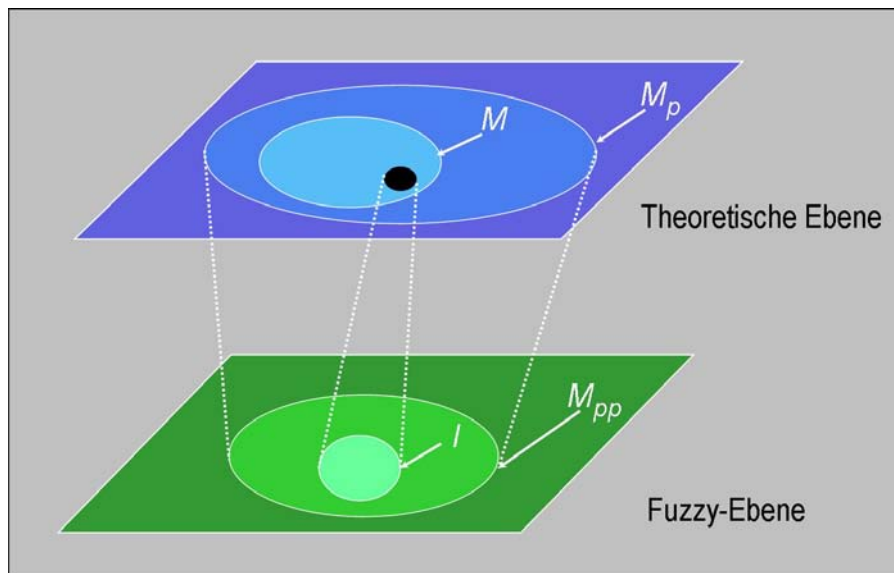


Abb. 7: Strukturen des (modifizierten) wissenschaftstheoretischen Strukturalismus.

6 Ausblick auf einen Fuzzy-Strukturalismus

Wissenschaftshistorisch lässt sich zeigen, dass die Theorie der Fuzzy Sets zur Überbrückung der Kluft zwischen wissenschaftlicher Theorie und Praxis entwickelt wurde. Dieses Ergebnis kann wissenschaftstheoretisch gewendet werden. Dazu wurde hier auf den Ansatz des wissenschaftstheoretischen Strukturalismus zurückgegriffen, der die Strukturen wissenschaftlicher Theorien durch mengentheoretische Prädikate charakterisiert. In diesem Beitrag wurde die Beziehung zwischen Theorie und Empirie durch eine Theoretisierungsbeziehung mit zwei Teilprozessen, der Fuzzifizierung und Defuzzifizierung, interpretiert. Dadurch lässt sich das Problem der theoretischen Begriffe neu fassen und die Theorie der Fuzzy Sets erhält einen wissenschaftstheoretische Anwendung bzw. Deutung, die die Rolle des Beobachters einbezieht.

Der klassische wissenschaftstheoretische Strukturalismus basiert auf gewöhnlicher Mengentheorie, und hier wurde bereits eine diesen klassischen Ansatz modifizierende Fuzzifizierung vorgenommen. Mit Zadehs Erweiterungsprinzip können aber auch weitere Strukturen wissenschaftlicher Theorien fuzzifiziert werden. Damit entsteht das Forschungsziel eines wissenschaftstheoretischen Fuzzy-Strukturalismus.

7 Literatur

- [1] Seising, R.: *Die Fuzzifizierung der Systeme. Die Entstehung der Fuzzy Set Theorie und ihrer ersten Anwendungen - Ihre Entwicklung bis in die 70er Jahre des 20. Jahrhunderts.* Stuttgart: Franz Steiner Verlag 2005.
- [2] Zadeh, L. A.: From Circuit Theory to System Theory, *Proceedings of the IRE*, Vol. 50, Nr. 5, May 1962, pp. 856-865.
- [3] Zadeh, L. A.: Fuzzy Sets, *Information and Control*, 8, 1965, pp. 338-353.
- [4] Bourbaki, Nicolas (pseudo.): *Elements of Mathematics: Theory of Sets*, Addison-Wesley, Reading, Mass. 1968.
- [5] Suppes, P.: A comparison of the meaning and uses of models in mathematics and the empirical sciences, *Synthese* 12, 1960, pp. 287-301.
- [6] Sneed, J. D.: *The Logical Structure of Mathematical Physics*, Dordrecht: Reidel 1971.
- [7] Balzer, W., Moulines, C. U., Sneed, J. D.: *An Architectonic for Science. The Structuralist Program.* Dordrecht: Reidel 1987.
- [8] Balzer, W.: *Empirische Theorien: Modelle – Strukturen – Beispiele. Die Grundzüge der modernen Wissenschaftstheorie.* Braunschweig/ Wiesbaden: Fr. Vieweg & Sohn 1982.
- [9] Balzer, W.: *Die Wissenschaft und ihre Methoden. Grundsätze der Wissenschaftstheorie. Ein Lehrbuch.* München: Verlag Karl Alber 1997.
- [10] Zadeh, L. A.: Fuzzy Sets, *Information & Control*, 8, 1965, pp. 338-53.
- [11] Zadeh, L. A.: Fuzzy Sets and Systems, in: J. Fox (ed.): *System Theory.* Microwave Res. Inst. Symp. Ser. XV, Brooklyn, NY: Polytech. Press 1965, p. 29-37.
- [12] Seising, R.: 1965 - Zadeh Begins: „A New View on System Theory“. Ein Beitrag zum 40. Jubiläum der Fuzzy Set Theorie. In: R. Mikut, M. Reischl (Hrsg.), Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik Universität Karlsruhe (TH) Bd. 9: 15. Workshop „Computational Intelligence“ des GMA-FA 5.14 „Computational Intelligence“ (FA VDI/VDE Gesellschaft für Mess- und Automatisierungstechnik) und der GI-FG „Fuzzy-Systeme und Soft-Computing“ (FG der Gesellschaft für Informatik) 16-19. Nov. 2005, Universitätsverlag Karlsruhe 2005, S. 1-19.
- [13] Zadeh, L. A.: From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions, *IEEE Transactions on Circuits And Systems-I: Fundamental Theory and Applications*, Vol. 45, No. 1, Jan. 1999, pp. 105-119.
- [14] Zadeh, L. A.: The Birth and Evolution of Fuzzy Logic – A Personal Perspective, *Journ. Japan Soc. for Fuzzy Theory and Systems*, Vol. 11, 6, 1999, pp. 891-905.
- [15] Zadeh, L. A.: A New Direction in AI. Toward a Computational Theory of Perceptions. *AI-Magazine*, Vol. 22, No. 1, 2001, pp. 73-84.
- [16] Zadeh, L. A.: Outline a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Trans. on Systems*, Vol. 3, 1973, pp. 28-44.
- [17] Zadeh, L. A., Fuzzy Logic = Computing with Words, *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 2, 1996, pp. 103-111.

Ein immunsystem-inspirierter Ansatz zum Überwachen des Lernens in Neuro-Fuzzy-Systemen*

Werner Brockmann, Daniel Michael Meyer, Alexander Horst

Universität Osnabrück
Institut für Informatik
Albrechtstraße 28
49069 Osnabrück
Werner.Brockmann@uos.de

1 Einleitung

In vielen automatisierungstechnischen Anwendungen werden lernfähige oder adaptive Systeme benötigt. Eine bewährte Methode dafür sind Neuro-Fuzzy-Systeme, weil sie universelle Funktionsapproximatoren sind, die ohne ein formales Prozessmodell auskommen. Wichtig für ihren Einsatz als lernfähiger Regler ist, dass sie das Einbringen von Vorwissen sowie das Absichern von sicherheitskritischen Betriebsbereichen erlauben. Denn das Lernen im geschlossenen Regelkreis stellt besonders hohe Anforderungen an das verwendete Lernparadigma, weil es durch die Closed Loop-Kopplung über den geregelten Prozess zu einem chaotischen Systemverhalten kommen kann. Denn das, was und wie gelernt wird, hat Auswirkungen auf das, was in Zukunft gelernt werden wird.

Nichtsdestotrotz können sich durch den Lernvorgang dynamisch beliebige Kennflächen ausprägen, was zu einem zwar zulässigen, aber suboptimalen Regel- und Lernverhalten führen kann. Damit es auch im nicht-sicherheitskritischen Betriebsbereichen zu keinem unerwünschten Lernverhalten kommt, muss der Lernvorgang online überwacht und gesteuert werden können. Dies ist die Aufgabe des SILKE-Ansatzes (System to Immunize Learning Knowledge-based Elements). Er nimmt dazu strukturell-funktionale Anleihen beim Immunsystem von höheren Lebewesen. Im Folgenden werden das Konzept und erste Untersuchungen vorgestellt. Sie entstanden im Rahmen des DFG-Schwerpunktprogramms „Organic Computing“ (SPP 1183).

2 Hintergrund der Arbeiten

2.1 Organic Computing

Die Motivation des Organic Computing ist, dass moderne technische Systeme wie Automobile oder Flugzeuge immer komplexer werden, aber in immer kürzerer Zeit entwickelt werden müssen. Gleichzeitig wachsen die Anforderungen an ihre Zuverlässigkeit und Verfügbarkeit. Daraus resultiert ein zunehmendes Engineering-Problem, insbesondere weil eine ausreichende Modellierung der technischen Systeme und/oder möglicher Fehlerszenarien nicht mehr möglich ist. Dadurch steigen die Wahrscheinlichkeiten von Engineering-Fehlern (Entwurf, Implementierung) und von Fehlern im laufenden Betrieb.

* Dieses Projekt wird gefördert durch die DFG unter BR 1980/1-1.

Auf der anderen Seite können sich organische Systeme an eine komplexe, unbekannte und oft sogar lebensfeindliche Umgebung anpassen und trotz vielfältiger, nicht vorhersehbarer Störungen wie Verletzungen oder Krankheiten überleben. Das Ziel des Organic Computing [1] ist es, die Organisationsprinzipien organischer Systeme zur Anpassung an und Behandlung von solchen Anomalien zu übernehmen, um komplexe, technische Systeme einfacher und schneller entwickeln zu können und gleichzeitig tolerant gegenüber unvorhergesehenen Situationen wie Störungen und Fehler zu machen. Entscheidend ist dabei, keine formalen (Fehler-)Modelle zu benötigen, was ansonsten den Entwurf komplexer technischer Systeme durch das explizite Engineering von Fehlertoleranzeigenschaften weiter erschweren würde.

Insbesondere sollen von organischen Systemen Eigenschaften übernommen werden, die ihnen das Überleben in einer unbekanntem, lebensfeindlichen Umgebung und bei Verletzungen ermöglichen, nämlich Fähigkeiten wie:

- Selbstoptimierung,
- Selbstorganisation,
- Selbstkonfigurierung,
- Selbstschutz,
- Selbstheilung.

Besonderes Augenmerk wird auf emergente Systemeigenschaften und Selbstorganisation gelegt.

Das vegetative Nervensystem und das Immunsystem dienen deshalb als Hauptideengeber. Während die Eigenschaften des (vegetativen) Nervensystems schon seit längerem z.B. durch reaktive Kontrollarchitekturen in autonomen mobilen Robotern und durch lernfähige Systeme auf Basis künstlicher neuronaler Netze nachgebildet werden, wird das natürliche Immunsystem erst seit Mitte der 90er-Jahre zunehmend als Metapher und Ideengeber benutzt. Es dient als Vorbild für robuste, lernfähige Systeme, die sinnvoll auf eine Vielzahl völlig unbekannter Situationen reagieren können. Die Basis dafür ist die Erkennung charakteristischer Muster von unbekanntem Ereignissen und die Generierung geeigneter Gegenmaßnahmen. Einen Überblick über die vielfältigen Anwendungen gibt [2].

2.2 Natürliche und künstliche Immunsysteme

Das natürliche Immunsystem [3] existiert parallel zu den lebenswichtigen Organen und hat mehrere sich gegenseitig ergänzende Wirkmechanismen, die die Antigene, also körperfremde Moleküle und Organismen wie Viren und Bakterien, erkennen und mittels spezifischer Antikörper unschädlich machen. Es werden das angeborene und das adaptive Immunsystem unterschieden. Das angeborene Immunsystem ist statisch und bekämpft Antigene recht unspezifisch. Im Gegensatz dazu ist das adaptive Immunsystem hochgradig anpassungsfähig, um auch neue Erreger bekämpfen zu können. Es bildet für unbekanntem Antigene spezifische Antikörper, die diese schneller und wirksamer bekämpfen als das angeborene Immunsystem. Allerdings benötigt das adaptive Immunsystem Zeit, um sich auf neue Erreger einzustellen. Solange übernimmt das angeborene Immunsystem den Schutz des Organismus.

Ein wesentlicher Bestandteil des Immunsystems sind die Lymphozyten (B- und T-Zellen). Sie erkennen spezifische Oberflächenmuster der Antigene und lösen die Immunantwort aus. Diese Oberflächenmuster sind i.d.R. nur für einen Teil der

Oberfläche der Antigene charakteristisch. Auf neue Antigene reagiert das adaptive Immunsystem durch eine Produktion neuer Rezeptoren mit einer zufälligen Rekombination von DNA-Molekülen und deren Selektion nach Wirksamkeit und Affinität zu bereits vorhandenen Antikörpern. Dadurch erhält das Immunsystem seine große Anpassungsfähigkeit und Robustheit. Gerade dieser Mechanismus wird bevorzugt von künstlichen Immunsystemen als Vorbild für lernfähige Systeme genutzt.

Ähnlich wie das Immunsystem in organischen Systemen dafür zuständig ist, Krankheitserreger zu erkennen und unschädlich zu machen, soll ein künstliches Immunsystem (AIS – Artificial Immune Systems) gute (self) und schlechte, anormale (nonself) Zustände eines technischen Systems unterscheiden und bei Bedarf für die Überführung in den intakten oder zumindest einen besseren Zustand sorgen. Diese Anomalieerkennung und –behandlung geschieht beim natürlichen Immunsystem auf Zell- oder Molekülebene, sichert aber den Gesundheitszustand des ganzen Organismus.

Bei lernfähigen Systemen wird der Mechanismus zur spezifischen Bildung von Antikörpern für neue Muster durch evolutionäre Algorithmen nachgebildet. Sie emulieren die zufällige Generierung neuer Antikörper und Auswahl durch Selektion. In manchen Fällen werden auch die Mechanismen des Immunnetzwerkes angewendet, um einen Pool an Antikörpern fester Größe zu haben, der aber auch alte, lange nicht mehr benötigte Antikörper vergisst und so ständig Platz für neue schafft. Durch diese Nachbildung des Immunnetzwerkes wird die Wahrscheinlichkeit erhöht, gute Lösungen zu finden und deren Qualität beizubehalten, selbst wenn sich die zugrunde liegenden Bedingungen mit der Zeit ändern.

2.3 Anwendungen künstlicher Immunsysteme

Die Anwendungen von künstlichen Immunsystemen reichen von der Hardware-Immunität [4] über Klassifikationsaufgaben bis zu regelungstechnischen Anwendungen. Das Hauptanwendungsgebiet ist allerdings die Klassifikation, wo sie besonders häufig für das Erkennen von Eindringlingen in Computernetzen, also die Self/Nonself-Unterscheidung bei der Intrusion Detection, eingesetzt werden.

Für diesen Artikel sind künstliche Immunsysteme in regelungstechnischen Anwendungen von Interesse. Dabei sind Ansätze zu unterscheiden, die ein AIS einsetzen, um direkt Parameter eines Reglers zu adaptieren [5], und solche, die (Struktur-)Parameter von Künstlichen Neuronalen Netzen (KNN) oder von Fuzzy-Systemen (FS) verändern und dadurch nur indirekt auf das Regelverhalten wirken [6].

[5] bildet das Verhalten von Lymphozyten dadurch nach, dass das AIS Rezeptoren als Indikatoren für Problemklassen bildet. Die Indikatoren sind analog zu den Antikörpern mit dem zu verwendenden Satz an Reglerparametern verknüpft, die das Problem optimal lösen. Der Ablauf ist dazu in die zwei Phasen Nonself-Erkennung und Bestimmung der Gegenmaßnahme (vgl. Antikörper-Entwicklung) aufgeteilt. In der ersten Phase erfolgt eine Systemidentifikation für die Simulation des aktuellen Prozesszustandes, damit dieser für den evolutionären Algorithmus simuliert werden kann. Dieser bestimmt dann in der zweiten Phase die optimierte Einstellung der Reglerparameter. Eine Datenbasis nimmt dann die so bestimmten Kombinationen aus Muster zur Situationserkennung und dem zugehörigen Parametersatz auf. Als Demonstrationsbeispiel dient die adaptive Regelung einer simulierten Boeing 747 bei Turbulenzen und sich änderndem Aktuatorverhalten. Die Adaption erfolgt „schnell“ und gut, weil die Reglerstruktur vorgegeben ist und „nur“ die Reglerparameter angepasst werden. Dadurch ist der Suchraum überschaubar. Dazu erfordert dieser Ansatz aber

auch viel Vorwissen und ist durch die Systemidentifikation, durch den evolutionären Algorithmus und die Datenbank relativ aufwändig.

Auch [6] bezweckt eine mitlaufende Anpassung an eine sich ständig ändernde Strecke und nutzt dazu ebenfalls einen evolutionären Algorithmus zur Optimierung. Dieser verändert hier strukturelle bzw. parametrierende Teile (Building Blocks) von Neuronalen Netzen (Verbindungsstärke) bzw. Fuzzy-Systemen (Verknüpfung von zwei Fuzzy-Eingangstermen zu einem Ausgangsterm). In beiden Fällen werden Neuronen als universelle Repräsentation verwendet. Das vorgeschlagene System ist einfach und schnell in der Lage, ein Helikopter-Modell mit Online-Identifikation und adaptiver (Neuro-)Regelung zu regeln bzw. das Verhalten von Piloten eines simulierten Trainingsflugzeugs zu modellieren. Diese Lösung ist sehr flexibel, aber ebenso wie der erste Ansatz sehr aufwändig und nicht online, closed-loop unter harten Echtzeitbedingungen anwendbar.

3 Der SILKE-Ansatz

3.1 Zielsetzung

Im Gegensatz zu den vorgenannten Ansätzen hat der nachfolgende Ansatz nicht die Aufgabe, das Reglerverhalten direkt zu adaptieren, sondern das dynamische Verhalten des Lernvorgangs in einem online adaptierten Regler zu überwachen bzw. bestimmte Eigenschaften der Reglerkennfläche zu gewährleisten. Er hat also eine (Metaebenen-)Kontrolle des dynamischen Lernvorgangs als Ziel, was insbesondere bei interagierenden, lernfähigen Systemen von Bedeutung ist.

Während andere Ansätze das Immunsystem recht detailliert nachbilden, werden hier Analogien auf abstrakterer, funktional-struktureller Ebene genutzt, weswegen der im Folgenden präsentierte Ansatz als „immunsystem-inspiriert“ bezeichnet wird. Er arbeitet zurzeit noch analog zum angeborenen Immunsystem mit einem statischen, nicht lernfähigen „Immunsystem“ und wirkt nur indirekt auf das Regelverhalten ein.

Die Ausgangsbasis sind Neuro-Fuzzy-Systeme in Form der einfachsten Variante eines Takagi-Sugeno-Systems mit non-fuzzy Singletons in den Konklusionen und einer gitterförmigen, orthogonalen Partitionierung des Eingangsraums. Das Lernen erfolgt überwacht (Struktur s. Bild 1), wobei bei Bedarf Vorwissen und ggf. ein Tagging-Mechanismus angewendet werden, um das Verändern der Regeln zu verhindern, die das Betreten sicherheitskritischer Betriebsbereiche ausschließen. Während das Lernen und der Tagging-Mechanismus direkt auf das Regelverhalten wirken, arbeitet der hier vorgestellte Ansatz komplementär dazu und soll die Freiheiten, die der Tagging-Mechanismus dem Lernvorgang noch lässt, überwachen oder gezielt kontrollieren, um beispielsweise

- zu harte Regelvorgänge zu vermeiden,
- die Konvergenz zu stabilisieren und oszillierendes Lernen zu minimieren,
- ggf. auch uninitialisierte Betriebsbereiche zu vermeiden.

Dabei wird gefordert, dass die Sicherheit des online adaptierten Systems nicht beeinträchtigt wird, kein (Prozess-)Modell erforderlich ist und die Algorithmen mit vertretbarem Aufwand auch harte Echtzeitbedingungen erfüllen.

3.2 Wirkprinzip

Die Metapher des Immunsystems auf lernende Neuro-Fuzzy-Systeme zu übertragen heißt hier, dass die Regeln bzw. Partitionen im Eingangsraum einzelnen Zellen, das lernende Neuro-Fuzzy-System einem Organ und das adaptierte Gesamtsystem dem Organismus entspricht. Krankheitserreger entsprechen Anomalien in der Kennfläche, die sich lokal auf Zell-/Regelebene ausprägen, aber zu einem unerwünschten, anormalen Systemverhalten führen. Reaktionen auf bestimmte „krankhafte“, pathologische Muster erfolgen wie beim natürlichen Immunsystem so lokal wie möglich, um die Auswirkungen der Anomalien und hier auch der Gegenreaktionen so klein wie möglich zu halten.

Der SILKE-Ansatz verfolgt deshalb die Überwachung des Lernvorgangs auf Zell- bzw. Regelebene durch lokale Operatoren, die in Anlehnung an das Immunsystem T-Zelle (Template-Zelle) genannt werden. Eine T-Zelle ist im ersten Schritt dafür zuständig, Anomalien wie Monotonie- oder Stetigkeitsverletzungen zu erkennen. Außerdem können, T-Zellen zum lokalen Korrigieren der Kennfläche eingesetzt werden, um das globale Verhalten auf Organismus-/Systemebene günstig zu beeinflussen.

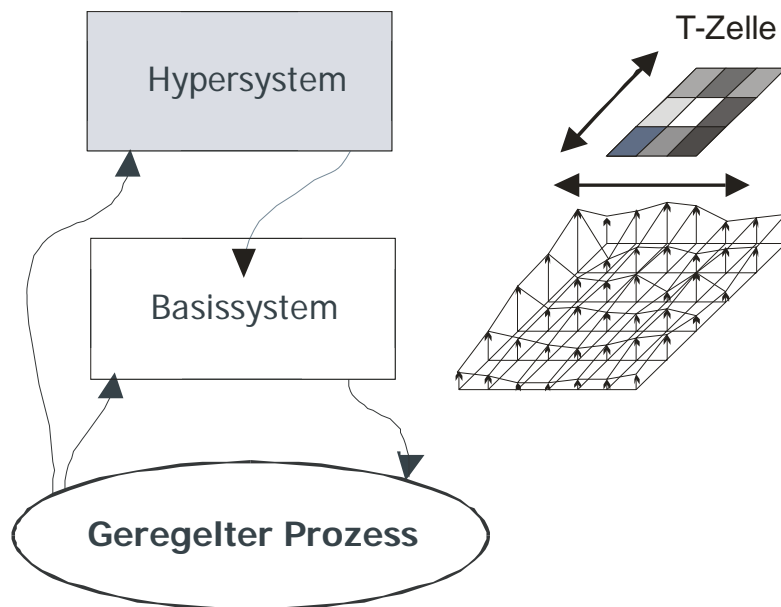


Bild 1: Prinzipbild des SILKE-Ansatzes

Wie Bild 1 verdeutlicht, regelt ein Basissystem den Prozess und wird durch ein so genanntes Hypersystem im laufenden Betrieb adaptiert. Das Hypersystem enthält dazu das Adaptionsgesetz. Parallel dazu werden ein oder mehrere Templates der T-Zelle(n) auf die gerade aktiven Zellen, also Regeln, im Basissystem angewendet. So wird modelliert, dass das natürliche Immunsystem parallel zu den lebentragenden Organen existiert und permanent den aktuellen Zustand des Organismus bewertet und entsprechend eingreift.

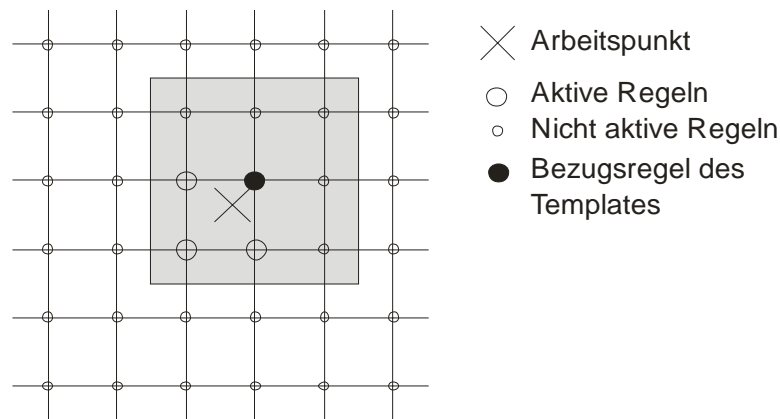


Bild 2: Struktur des verwendeten Templates

Für alle durch den aktuellen Arbeitspunkt aktivierten Regeln wird ein Template angewendet, das dann die jeweils benachbarten Regeln einbezieht. In Bild 2 ist das beispielhaft für einen zweidimensionalen Eingangsraum durch die grau unterlegte Fläche angedeutet. Der Ablauf des Lernens mit dem SILKE-Ansatz führt dabei zu jeder Abtastung folgende Aktionen aus. Dabei sind die Schritte 3. bis 5. spezifisch für den SILKE-Ansatz.

1. Basissystem auswerten und Stellaktion(en) ausgeben
2. Hypersystem auswerten und Korrekturvorgabe für das Basissystem generieren,
3. Konklusionen aller aktiven Regeln des Basissystems gemäß Vorgabe lernen,
4. für alle aktiven Regeln das oder die Templates anwenden und die Ergebnisse zwischenspeichern (Bei den anderen dabei beteiligten aktiven Regeln wird der gerade gelernte Wert ohne Modifikation durch ein Template verwendet.),
5. die so ermittelten Werte für alle aktiven Regeln in die Regelbasis eintragen.

Für die Templates sind verschiedene Funktionalitäten und Operatoren denkbar. Die erste Stufe ist der Einsatz zur Überwachung des „Gesundheitszustands“ des Systems. Das entspricht hier der Überprüfung der Kennfläche auf bestimmte Eigenschaften, d.h. also das Erkennen und Signalisieren charakteristischer, „pathologischer“ Muster. Die nächste Stufe ist der direkte Eingriff in die dynamisch adaptierte Kennfläche, um bestimmte Eigenschaften sicherzustellen. Dies ist anspruchsvoller, weil dadurch ein direkter Einfluss auf den Lernprozess ausgeübt wird. Die Auswirkungen sind vergleichbar mit dem Aufbau eines chaotischen Systems wie beim Online-Closed Loop-Lernen, nur mit einer zusätzlichen Indirektionsstufe. Trotzdem wurde bei den nachfolgenden Untersuchungen direkt mit dem SILKE-Ansatz auf das Regelverhalten eingewirkt, um Erfahrungen bzgl. des dynamischen Lernverhaltens in interagierenden lernfähigen Systemen zu sammeln, die dann auch für den Einsatz zur Überwachung genutzt werden.

Die Templates überprüfen lokale Nachbarschaften auf Eigenschaften wie Stetigkeit, Steigung, Monotonie usw. Weil dies in einem mehrdimensionalen Hyperraum passiert, ergibt sich ein Rechenaufwand, der von der Dimensionalität und davon abhängt, ob nur orthogonale oder auch diagonale Nachbarn einbezogen werden (Tabelle 1). Der Aufwand pro betrachteter Konklusion wächst bei orthogonaler Nachbarschaft linear, aber exponentiell bei Berücksichtigung auch der die diagonalen Nachbarn. Das ist

besonders krass, wenn nicht nur die unmittelbaren Nachbarn (1er-Nachbarschaft), sondern auch die übernächsten Nachbarn eingehen (2er-Nachbarschaft). Weil aber die Fuzzy-Eingangsvariablen i.d.R. in weniger als ein Dutzend Fuzzy-Mengen gegliedert werden, macht ein Operator der fünf von ihnen verarbeitet wenig Sinn, denn die angestrebte lokale Wirkung wird dadurch in Frage gestellt. Deshalb wird nur eine 1er-Nachbarschaft verwendet. So bleibt der Realisierungsaufwand in der gleichen Größenordnung wie die Auswertung des Fuzzy-Systems selbst, wodurch die Umsetzung insbesondere bei orthogonaler Nachbarschaft auch auf Mikrocontrollern in Echtzeit möglich ist.

Tabelle 1: Rechenaufwand für ein Template bei n Eingangsvariablen

Aufwand	1er-Nachbarschaft	2er-Nachbarschaft
Orthogonal	$2 \cdot n \cdot 2^n$	$4 \cdot n \cdot 2^n$
Orthogonal + diagonal	$(3^n - 1) \cdot 2^n$	$(5^n - 1) \cdot 2^n$

4 Anwendungsbeispiel

4.1 Aufbau der Untersuchung

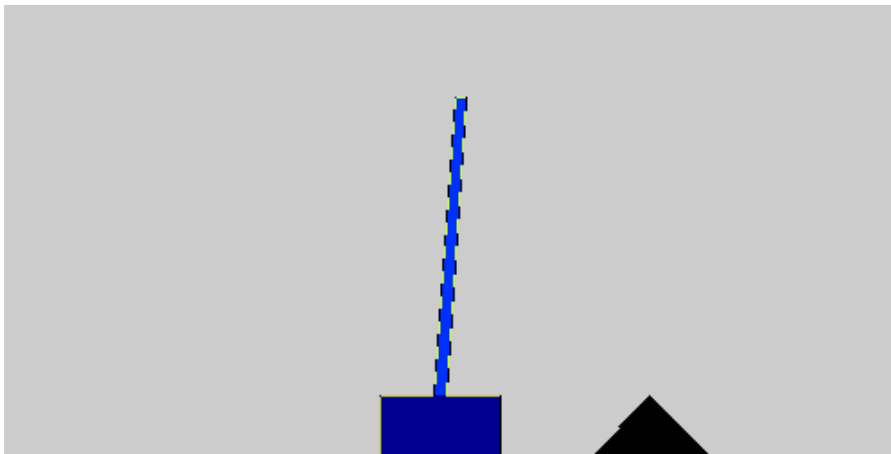


Bild 3: Struktur der verwendeten Regelstrecke

Als Anwendungsbeispiel dient die adaptive Regelung eines simulierten inversen Pendels, dessen Schlitten an einer vorgebbaren Stelle positioniert werden soll (Bild 3). Dazu wird eine Kaskadenstruktur aus einem Positions- und einem Winkelregler verwendet, die beide aus lernfähigen Neuro-Fuzzy-Reglern bestehen. Beide Regler sind unabhängig von einander mit einem übergeordneten Hypersystem versehen, das das jeweilige Adaptionsgesetz enthält und Lernvorgaben für den zugehörigen Regler (Basissystem) generiert. Die Lernvorgaben werden unmittelbar als Einzelmuster gelernt und angewendet.

Die simulierte Abtastrate des Positioniersystems beträgt 2 ms, die des Winkelreglers 20 ms. Zu jeder Abtastung werden die Basissysteme angewendet, die Hypersysteme ausgewertet und der Lerninput ggf. mit dem SILKE-Ansatz in den Basissystemen verarbeitet.

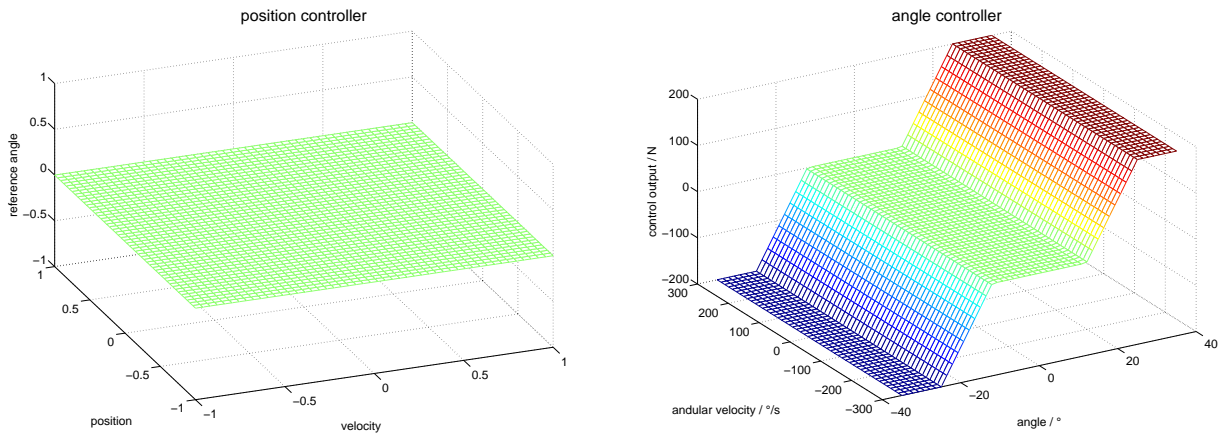


Bild 4: Kennflächen des Positionsreglers (links) und des Winkelreglers (rechts) vor dem Lernen

Das Basissystem des Positionsreglers verarbeitet den Positionsfehler und die Schlittengeschwindigkeit als Eingangsgrößen und gibt Sollwinkel an den Winkelregler vor. Es enthält 5*5 Regeln, die komplett mit 0 vorinitialisiert sind. Die Regelbasis enthält also keinerlei Vorwissen. Dahingegen verknüpft das Basissystem des Winkelreglers den Winkelfehler, die Winkelgeschwindigkeit und den Sollwinkel in einer 9*9*5 Regeln umfassenden Regelbasis und gibt die Kraft als Stellgröße zur Beschleunigung des Schlittens aus. Es enthält grobes Vorwissen, das in frühen Lernphasen ein Umkippen des Pendels verhindert. Die Kennflächen des Initialisierungswissens beider Regler sind in Bild 4 zu sehen.

Das Hyperwissen des Positioniersystems basiert auf den gleichen Eingangsgrößen wie das Basissystem und verändert den vom Basissystem auszugebenden Sollwinkel umso mehr, je stärker die Kombination aus Positionsfehler und Schlittengeschwindigkeit von dem im Hyperwissen enthaltenen als ideal angestrebten Bewegungsmuster abweicht. Dies besagt, dass kleine Positionsfehler auch durch kleine Schlittengeschwindigkeiten ausgeglichen werden sollten, und große durch große. Wird die angestrebte Kombination eingehalten, ist also insbesondere sowohl der Positionsfehler als auch die Schlittengeschwindigkeit null, werden keine Korrekturen vorgegeben. D.h., das Basiswissen im Positionsregler bleibt dann unverändert.

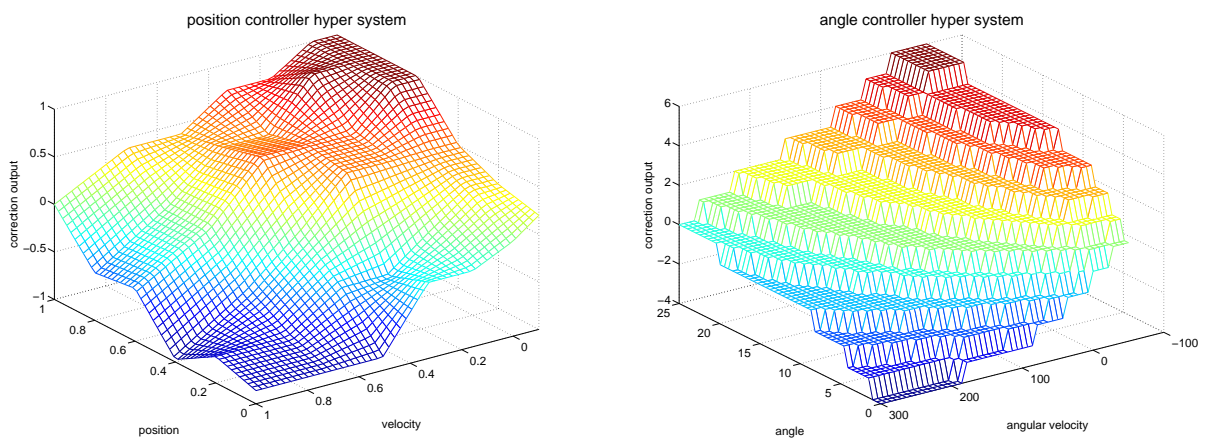


Bild 5: Kennflächen der Hypersysteme des Positionsreglers (links) und des Winkelreglers (rechts)

Das Hypersystem für den Winkelregler nimmt in ähnlicher Weise keine Änderungen am Basissystem vor, wenn kleine Winkelfehler auch durch kleine und große Winkelfehler durch große Winkelgeschwindigkeiten in die richtige Richtung ausgeglichen werden. Insbesondere sollte auch hier bei einem Winkelfehler von null die Winkelgeschwindigkeit null sein. Je stärker das Verhalten der Regelstrecke von dieser Wunschvorstellung abweicht, umso größer sind die Korrekturen, die das Hypersystem vorgibt.

Beide Hypersysteme arbeiten symmetrisch bzgl. der Regeldifferenz. Die Regeln wirken also doppelt. Die Ableitung der Regelgröße ist in Richtung des Sollwertes feiner aufgeteilt, wie die Kennflächen beider Hypersysteme in Bild 5 zeigen.

Sowohl durch die Kaskadenstruktur als auch durch die Closed loop-Kopplung über den geregelten Prozess interagieren beide lernfähigen Regler sehr stark miteinander. Zum Beispiel ändert sich die Kennfläche des unterlagerten Winkelreglers ständig, solange keine Konvergenz erreicht ist. Durch diese Eigendynamik des Lernvorgangs erzielt das Positioniersystem bei gleichen Eingangsgrößen jeweils eine andere Wirkung des Sollwinkels. Das führt zu einem stark dynamischen Lernvorgang, der durch den SILKE-Ansatz sinnvoll beeinflusst wird, wie die nachfolgenden Vergleiche zeigen. Dabei wird einmal ohne den SILKE-Ansatz gearbeitet und einmal der SILKE-Ansatz sowohl für den Positions- als auch Winkelregler angewendet.

In den nachfolgenden Untersuchungen wird ein mittelnder Operator mit 1er-Nachbarschaft als einziges Template verwendet. Es bildet den Mittelwert aller orthogonalen und diagonalen Nachbarn ohne die betrachtete Regel selbst. Damit wird das Ziel verfolgt, eine zu große Steilheit der Kennfläche (insbesondere in der Nulllage) zu vermeiden. Dadurch sollen einerseits zu starke Stelleingriffe vermieden werden. Andererseits, und das ist an dieser Stelle wichtiger, soll so der Lernvorgang stabilisiert und die Konvergenz verbessert werden.

4.2 Untersuchungsergebnisse

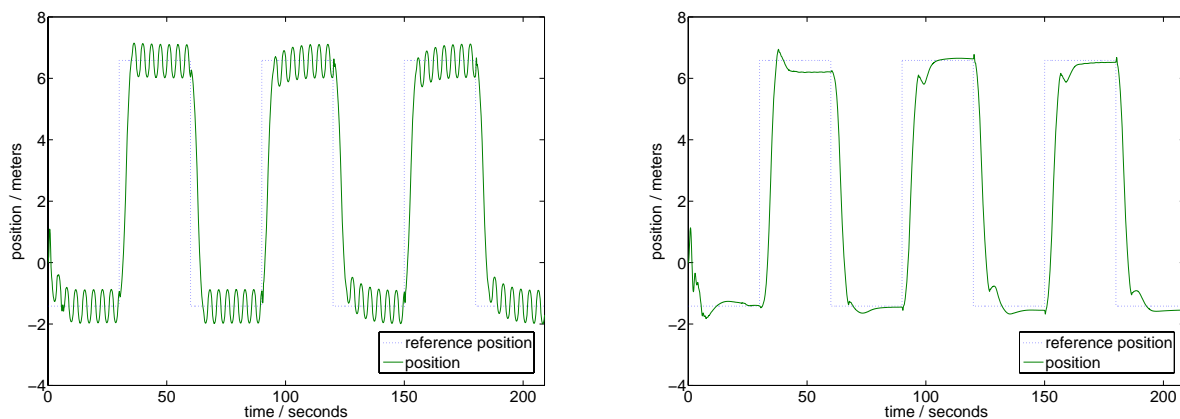


Bild 6: Verlauf der Schlittenposition ohne (links) und mit dem SILKE-Ansatz (rechts)

Ohne den SILKE-Ansatz (Bild 6, links) zeigt die Regelstrecke ein hoch dynamisches Verhalten, bei dem die Regelgröße stark oszilliert, weil das Lernen nicht konvergiert. Mit dem SILKE-Ansatz verbessert sich das Regelverhalten deutlich, wie in Bild 6 (rechts) zu sehen ist. Das Lernen konvergiert schnell und gut. Bereits die erste Sollwertvorgabe wird relativ gut ausgeregelt. Außerdem lernt das System schnell (innerhalb von zwei Sollwertwechseln), das Pendel zunächst in die Gegenrichtung

anzukippen (zu sehen an der kleinen Spitze bei einem Sollgrößenprung), um es in Richtung der neuen Zielposition zu beschleunigen, und in gleicher Weise vor der Zielposition wieder abzubremsen (Gegenschwingung vor Erreichen des Endwertes).

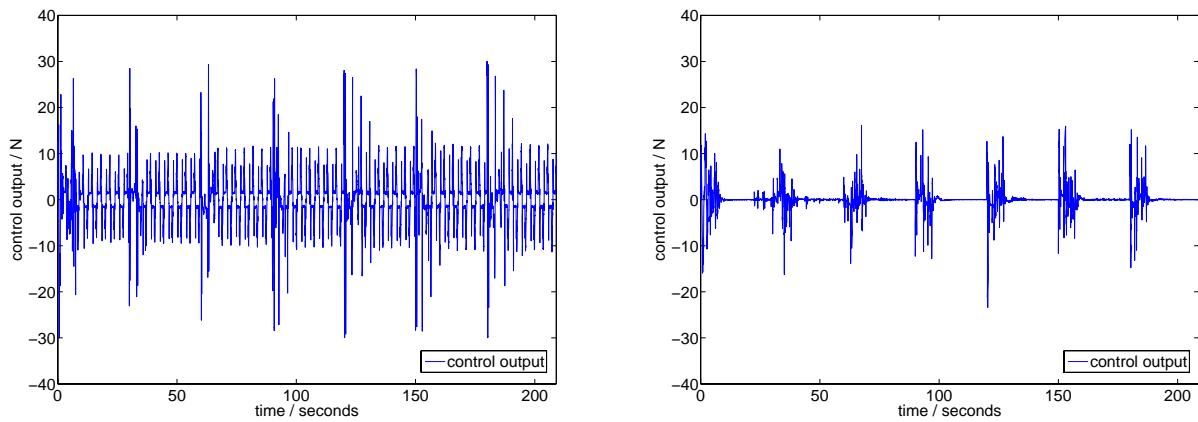


Bild 7: Stellgrößenverlauf ohne (links) und mit dem SILKE-Ansatz (rechts)

Dieses Verhalten spiegelt sich auch im Stellgrößenverlauf wider. Während mit dem SILKE-Ansatz (Bild 7, rechts) nur in den dynamischen Bewegungsphasen größere Stellaktionen auftreten, arbeitet das System ohne den SILKE-Ansatz mit einer ständigen Regelbewegung (Bild 7, links). Es wird insgesamt mehr als das Vierfache der Stellenergie verbraucht und das Stellglied stärker belastet als mit dem SILKE-Ansatz. Den Kennflächen des Positionsreglers (Bild 8, links) und des Winkelreglers (Bild 8, rechts) sieht man dieses deutlich unterschiedliche Verhalten nicht auf den ersten Blick an, weil sie eine Momentaufnahme wiedergeben.

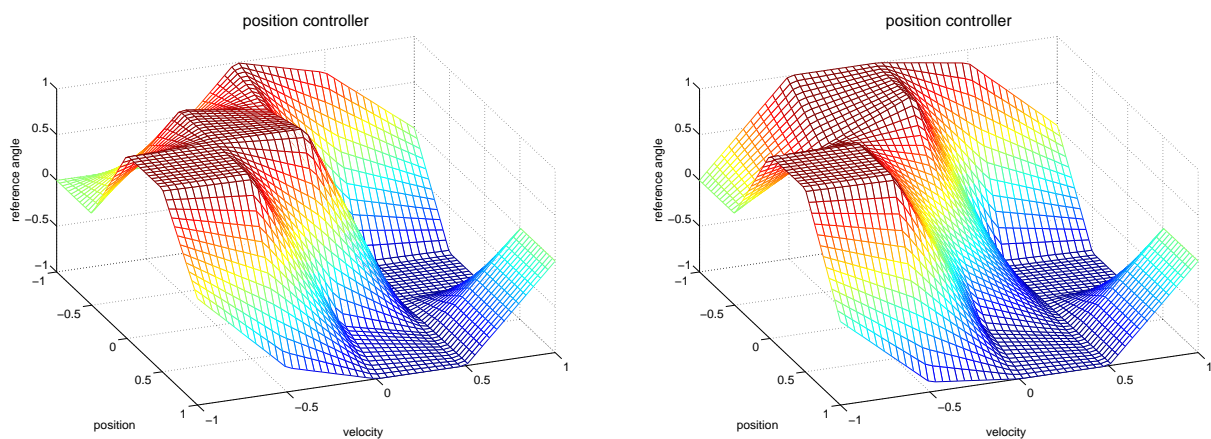


Bild 8: Kennfläche des Positionsreglers ohne (links) und mit dem SILKE-Ansatz (rechts) nach einer Lerndauer von 210 Sekunden

Bei genauerer Betrachtung sieht man in Bild 8 (links), dass bei der Regelung ohne den SILKE-Ansatz im betrachteten Zeitpunkt für einen Positionsfehler von null und eine Schlittengeschwindigkeit von null der maximale Sollwinkel vorgegeben wird. Zu anderen Zeitpunkten ist es wiederholt der minimale Sollwinkel. So stark modifiziert das Lernen dynamisch die Kennfläche bzw. das Verhalten der Positionsregelung von einem Führungsgrößenwechsel zum nächsten. Mit dem SILKE-Ansatz konvergiert das Lernen dahingegen so gut, dass nur noch minimale Änderungen an der Kennfläche

vorgenommen werden und für einen Positionsfehler von null und eine Schlittengeschwindigkeit von null auch ein Sollwinkel von nahezu null ausgegeben wird (Bild 8, rechts). Dadurch steht das Pendel natürlich auch wesentlich stabiler (s. Bild 6).

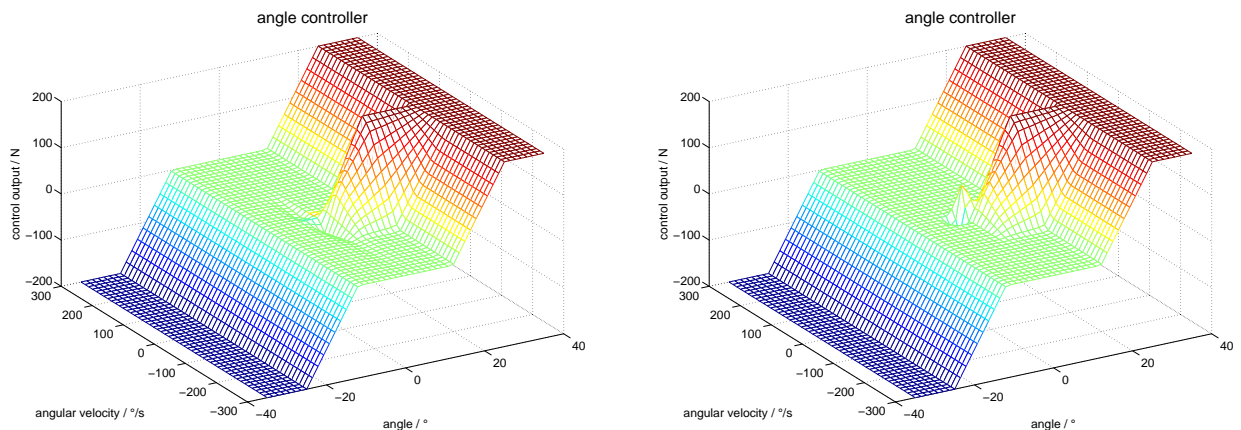


Bild 9: Kennfläche des Winkelreglers ohne (links) und mit dem SILKE-Ansatz (rechts) nach einer Lerndauer von 210 Sekunden (Schnitt für einen Sollwinkel von 0°)

Beim Winkelregler (Bild 9) ist an den Kennflächen zu erkennen, dass in beiden Fällen große Bereiche des Eingangsraums ungelernt bleiben und nur zu Beginn des Lernvorgangs in einem größeren Bereich des Eingangsraums gelernt wurde. Danach arbeitet das System am meisten nur noch in der Nähe des Ursprungs, auch in der hier nicht dargestellten dritten Dimension.

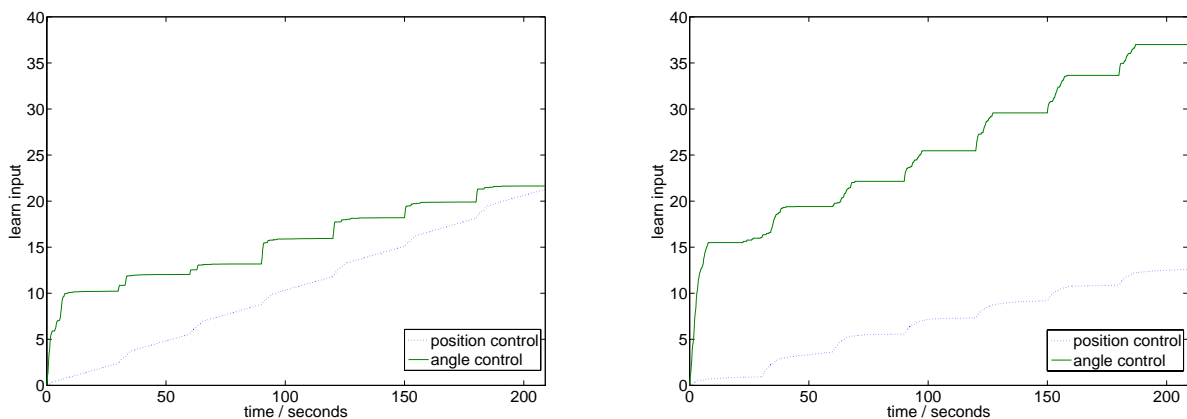


Bild 10: Akkumulierte Lernvorgaben der Hypersysteme ohne (links) und mit dem SILKE-Ansatz (rechts)

Der Hauptunterschied im Regel- und Lernverhalten rührt daher, dass ohne den SILKE-Ansatz wesentlich mehr Dynamik in den Kennflächen ist. Das verdeutlicht Bild 10, in dem die Beträge der Lernvorgaben akkumuliert dargestellt werden. Während der dynamischen Phasen nach einem Sollwertwechsel werden in beiden Fällen in beiden Reglern deutliche Lernvorgaben generiert, was an dem Anstieg der Kurven zu sehen ist. Ohne den SILKE-Ansatz (Bild 10, links) steigt die Kurve der Positionsregelung auch während der statischen Phasen kontinuierlich weiter, weil das Lernen nicht konvergiert, also ständig Lernvorgaben generiert werden. Mit dem SILKE-Ansatz erfolgen in den statischen Phasen für beide Regler keine weiteren Lernvorgaben, was an den Plateaus

im Kurvenverlauf in Bild 10 (rechts) zu sehen ist. Das resultiert auch in einer in der Summe deutlich geringen Lernvorgabe für den Positionsregler.

Im Gegensatz dazu ist die Lernvorgabe in Summe für die Winkelregelung ohne den SILKE-Ansatz deutlich kleiner als mit, weil der Winkelregler die Vorgaben unmittelbar umsetzen kann. Dadurch wird die Regelstrecke deutlich stärker direkt vom Positionsregler beeinflusst als mit dem SILKE-Ansatz. Bei diesem muss das Hypersystem für den Winkelregler wegen der mittelnden Wirkung des Templates deutlich stärkere Vorgaben machen. Das ist in Bild 10 (rechts) daran zu erkennen, dass die Lernvorgaben gleich zu Beginn wesentlich stärker ausfallen und auch im laufenden Betrieb stärker ansteigen als ohne den SILKE-Ansatz.

5 Diskussion

Diese Untersuchungen zeigen, dass mit dem SILKE-Ansatz das dynamische Lernverhalten und damit auch das Systemverhalten während des Lernvorgangs kontrolliert werden kann. Das funktioniert bei dem hier verwendeten mittelnden Operator auch schon in frühen Phasen im unvollständig gelernten System, obwohl die Lernvorgaben durch ungelernete Bereiche, die noch das Initialisierungswissen enthalten, abgeschwächt werden. Das kann zu einer langsameren Konvergenz in der Anfangsphase führen. Sind allerdings die Kennflächen halbwegs ausgeprägt, wird die Konvergenz stabilisiert.

Die ersten Erfahrungen bei der Umsetzung dieses Beispiels zeigen, dass das Engineering lernfähiger Regelungen für die Anwendung des SILKE-Ansatzes nicht geändert werden muss. Insbesondere bei interagierenden lernfähigen Systemen scheint der Entwicklungsaufwand sogar reduziert zu werden, weil das Engineering leichter und schneller vonstatten geht. Denn das Feintuning des Hyperwissens war durch den verwendeten Template-Operator nicht mehr so kritisch. Allerdings sind für eine fundiertere Aussage weitere Untersuchungen nötig.

6 Abschlussbemerkungen

Nach unserem jetzigen Wissenstand wurde bisher noch nicht in der hier beschriebenen Form auf lernfähige Systeme eingewirkt. Wir vermuten daher, eine neue Qualität für lernfähige Systeme und Regelungen mit vielfältigen Anwendungsmöglichkeiten geschaffen zu haben. Diese reichen von der Online-Überwachung (Diagnose), z.B. Alarm auslösen bei charakteristischen Mustern, bis zu hier vorgestellten Korrekturen in den Lernvorgang, um lokale Störungen, also Anomalien, zu beseitigen. Evtl. kann der SILKE-Ansatz auch für weitere Aufgaben eingesetzt werden, z. B. für eine lokale Extrapolation, um ungelernete Bereiche mit lokalem Wissen dynamisch vorzuinitialisieren. Dadurch sind ein beschleunigtes Lernen und eine größere Betriebssicherheit online lernender Systeme zu erwarten.

Der SILKE-Ansatz unterscheidet sich von den in [5, 6] beschriebenen Arten, das Immunsystem technisch nachzubilden. Das Ziel ist auch ein anderes, nämlich nicht das Lernen an sich, sondern die Kontrolle des Lernvorgangs als solchen. Die Motivation dafür wird durch den geplanten Einsatz innerhalb der ORCA-Architektur (Organic Robot Control Architecture, [7]) deutlich. Diese Architektur besteht aus einer Ansammlung interagierender (lernfähiger) Module, die auf verschiedenen Hierarchieebenen angeordnet sein können. Die Fehlertoleranz dieser Architektur soll durch Emergenz und Selbstorganisation erreicht werden, ohne dafür ein Fehlermodell

zu benötigen. Dafür sollen intensiv Lernverfahren eingesetzt werden, woraus sich die Notwendigkeit der garantierten Einhaltung von Sicherheitsbedingungen und einer stärkeren Kontrolle des Lernvorgangs ergibt.

Tabelle 2: Einordnung des SILKE-Ansatzes in natürliche Immunsysteme

Granularität in der Natur	Systemebene	Neuro-Fuzzy-Systeme	SILKE-Ansatz
Zelle	Parameter	Regel	T-Zelle / Template
Organ	Modul	Regelbasis	SILKE-Algorithmus
Körper	Gesamtsystem	Interagierende Module	Künstliches Immunsystem

Vor diesem Hintergrund lässt sich der SILKE-Ansatz wie in Tabelle 2 mit einem organischen Immunsystem vergleichen. In einem Organ macht nicht eine einzelne Zelle die Funktion des Organs aus, sondern das Kollektiv. Einzelne fehlerhafte Zellen werden dadurch toleriert. Analog führt i.d.R. eine einzelne fehlerhafte Regel nicht zum Versagen eines Reglers / Moduls. Wie das Immunsystem bei organischen Systemen auf (Molekül- und) Zellebene arbeitet, erfolgen beim SILKE-Ansatz die Eingriffe auf Regelebene. Aber genauso wie die Organe stark vernetzt sind, interagieren bei komplexen Automatisierungssystemen viele Module stark miteinander. Umso wichtiger ist daher, dass deren Interaktion überwacht und (im Fehlerfall) korrigierend eingegriffen wird. Die Korrekturen sollten so lokal wie möglich erfolgen. Dennoch muss die Vernetzung der Module berücksichtigt werden. Und wie ein organisches System grundsätzlich ohne das Immunsystem funktionsfähig wäre, stellt der SILKE-Ansatz eine Ergänzung für lernfähige Automatisierungssysteme dar.

Hier wurden das Konzept und die ersten Untersuchungen des SILKE-Ansatzes vorgestellt. Naturgemäß bleiben daher noch viele Fragen offen. So sollen weitere Operatoren für Templates und weitere Einsatzmöglichkeiten untersucht werden. Dabei können weitere Anleihen beim natürlichen Immunsystem genommen werden, z.B. um evtl. später auch Templates zu lernen. Dadurch würde der SILKE-Ansatz, der aktuell mit dem angeborenen Immunsystem zu vergleichen ist, zu einem adaptiven Immunsystem, das für spezifischere Reaktionen (für unvorhergesehene Situationen) lernt und ggf. wieder vergisst. Analog zum Immunnetzwerk würde es dann nicht nur ein Template geben, sondern einen ganzen Satz von verschiedenen Templates/T-Zellen, die auf verschiedene Anomalien (Antigene) reagieren. Außerdem ist eine detailliertere Nachbildung der Lymphozyten denkbar, wie die Trennung von Rezeptoren zur Erkennung von Antigenen und dem Reaktionsteil.

Nichtsdestotrotz ist die frühzeitige Übertragung in reale Anwendungen sinnvoll, um die Praxistauglichkeit zu prüfen. Dazu ist die Übertragung auf einen echten, radgetriebenen Pendelaufbau sowie auf einfache, mehrgliedrige Krabbelroboter geplant, um zum einen das Verhalten unter realen Betriebsbedingungen (mit Rauschen, Reibung, Lose, ...) und zum anderen insbesondere auch interagierende, lernfähige Systeme in verkoppelten Systemen unter Echtbedingungen zu untersuchen.

7 Literatur

- [1] Müller-Schloer, Ch.; von der Marlsburg, Ch.; Würtz, R. P.: *Organic Computing*. Informatik Spektrum, 2004, 332-336
- [2] de Castro, L.N.; von Zuben, F.J.: *Artificial Immune Systems: Part III – A Survey of Applications*. Technical Report DCA-RT 02/00, 2000
- [3] Schütt, CH.; Bröker, B.: *Grundwissen Immunologie*. Elsevier GmbH, München, 2006
- [4] D.W. Bradley, A.M. Tyrell: *The Architecture for a Hardware Immune System*. 3rd NATO/DoD Workshop on Evolvable Hardware, Long Beach 2001, 193-201
- [5] Charles L. Karr, K. Nishita, Kenneth S. Graham: *Adaptive Aircraft Flight Control Simulation Based on an Artificial Immune System*. Applied Intelligence 23, 295-308, 2005
- [6] K. Krishnakumar, A. Satyadas, J. Niedhoefer: *An Immune System Framework for Integrating Computational Intelligence Paradigms with Applications to Adaptive Control*. In: M. Palaniswami, Y. Attikiouzel, R. J. Marks II, D. Fogel and T. Fukuda (Eds.): *Computational Intelligence - A Dynamic System Perspective*, IEEE Press, 1995, 32-45
- [7] Brockmann, W.; Maehle, E.; Mösch, F.: *Organic Fault-Tolerant Control Architecture for Robotic Applications*. 4th IARP/IEEE-RAS/EURON Workshop on Dependable Robots in Human Environments, Nagoya, Japan, 16.-18.6.2005

Quasi-Online Newtonverfahren zur Identifikation nichtlinearer dynamischer Systeme mit strukturierten rekurrenten Netzen

Christian Endisch, Dierk Schröder

Lehrstuhl für Elektrische Antriebssysteme – Technische Universität München

Arcisstraße 21, 80333 München

Tel.: (089) 289-28449

Fax: (089) 289-28336

E-Mail: christian.endisch@tum.de

1 Einleitung

Die Identifikation eines nichtlinearen dynamischen Systems kann bei ausreichender Systemkenntnis mit einem speziell für das System entwickelten strukturierten rekurrenten Netz erfolgen. Ein solches Netz wird aus dem nichtlinearen Signalflussplan (SFP) abgeleitet, wobei nur die Struktur des nichtlinearen SFP bekannt sein muss. Die linearen Parameter und die vorhandenen Nichtlinearitäten sollen bei der Identifikation bestimmt werden. Bisher kam bei der Identifikation mit dem strukturierten rekurrenten Netz stets das bekannte Gradientenabstiegsverfahren zum Einsatz [1]. Dieses Verfahren besitzt eine langsame Konvergenzgeschwindigkeit und auch die Wahl der richtigen Lernschrittweiten hat sich als problematisch herausgestellt. Um diese Nachteile zu vermeiden, wird in dieser Veröffentlichung ein Vorgehen beschrieben, mit dem es möglich ist, die schnellen nichtlinearen Optimierungsverfahren 2. Ordnung (bei der Optimierung wird auch die zweite Ableitung berücksichtigt) für strukturierte rekurrente Netze einzusetzen. Der Gradient (Ableitung des Fehlers nach den Parametern des strukturierten rekurrenten Netzes) und die zweite Ableitung werden wie beim Gauß-Newton-Verfahren aus der Jacobimatrix (enthält mehrere Gradientenvektoren zu verschiedenen Zeitpunkten) ermittelt. Die notwendige Jacobimatrix wird schrittweise mit den Ableitungen, welche online berechnet werden, aufgefüllt. Ist die Matrix vollständig besetzt, so wird ein Optimierungsschritt berechnet. Diese Fensterung muss auch beim Ausgangsfehler erfolgen. Es entsteht ein quasi-online Lernverfahren. Die Nachbildung der Nichtlinearitäten beim strukturierten rekurrenten Netz übernehmen GRNN [2]. Der Ausgang eines GRNN (General Regression Neural Network) entsteht durch die Überlagerung gewichteter Aktivierungsfunktionen. Zuerst wird das strukturierte rekurrente Netz kurz vorgestellt. Nach dem bekannten Gradientenabstiegsverfahren soll das Quasi-Online Newtonverfahren beschrieben werden. Abschließend erfolgt die Identifikation eines Laborbeispiels.

2 Strukturierte rekurrente Netze

Strukturierte rekurrente Netze wurden erstmals von Brychcy [3] zur Systemidentifikation eingesetzt. Hintz [1] erweiterte dieses Identifikationsverfahren um die Beobachterkoeffizienten. Dabei nutzte er die Erkenntnisse aus der Theorie der lernfähigen Beobachter [4]. Strukturierte rekurrente Netze ermöglichen die Identifikation nichtlinearer dynamischer Systeme. Die physikalische Interpretierbarkeit des Identifikationsergebnisses steht

dabei im Vordergrund. Voraussetzung für die Anwendung dieses Verfahrens ist, dass die Struktur des Systems in Form einer nichtlinearen Zustandsbeschreibung vorliegt:

$$\dot{\underline{x}} = \mathbf{A} \cdot \underline{x} + \underline{b} \cdot u + \mathbf{K} \cdot \underline{\mathcal{NL}}(u, \underline{x}) \quad (1a)$$

$$y = \underline{c}^T \cdot \underline{x} \quad (1b)$$

u bezeichnet den skalaren Systemeingang, \underline{x} den Zustandsvektor, \mathbf{A} die lineare System-

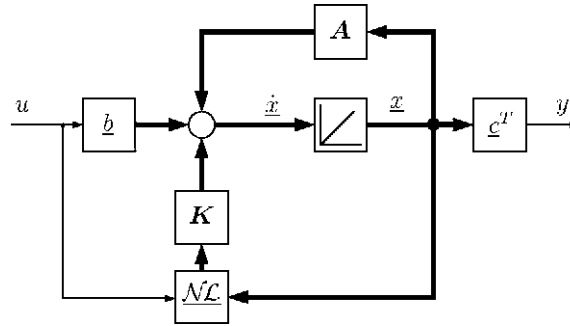


Bild 1: Nichtlineare Zustandsdarstellung des Systems

matrix, \underline{b} den Einkoppelvektor, \mathbf{K} die Kopplungsmatrix der Nichtlinearitäten, $\underline{\mathcal{NL}}$ den Vektor der statischen Nichtlinearitäten und y den Systemausgang. Das strukturierte rekurrenten Netz wird anhand des gegebenen nichtlinearen SFP aufgebaut. Die Nichtlinearitäten werden durch statische Neuronale Netze nachgebildet. Bild 2 zeigt die Transfor-

SFP				
str. rek. Netz				

Bild 2: Umwandlung des Signalflussplanes (SFP) in die diskrete Darstellung des strukturierten rekurrenten Netzes

mation der Elemente des SFP in ein zeitdiskretes rekurrentes Netz mit der Abtastzeit h . Die Summationspunkte im SFP bleiben erhalten und werden zu sogenannten Neuronen des rekurrenten Netzes. Die linearen Parameter im SFP werden zu Gewichten $\hat{\underline{\Psi}}$ zwischen den Neuronen. Die Integratoren werden nach der Euler-vorwärts Approximation (Rechteckapproximation) mit dem Integratoreingang in , dem Integratorausgang out und der Abtastzeit h implementiert ($out_{k+1} = out_k + h \cdot in_k$). Die statischen Nichtlinearitäten werden durch eine entsprechende Anzahl an GRNN ersetzt. Diese bestehen wiederum aus Neuronen und Gewichten $\hat{\underline{\Theta}}$, die adaptiert werden können. Alle zu optimierenden Parameter sind im Parametervektor $\hat{\underline{w}}$ (siehe Gleichung (2)) zusammengefasst. Der Parametervektor setzt sich aus p Elementen zusammen und beinhaltet die linearen Parameter $\hat{\underline{\Psi}}$, die die linearen Systemanteile erfassen und die nichtlinearen Parameter $\hat{\underline{\Theta}}$, welche die

nichtlinearen Systemanteile erfassen.

$$\begin{aligned}\hat{\underline{w}} &= [\hat{w}_1 \quad \dots \quad \hat{w}_p]^T \\ &= [\hat{\underline{\Psi}} \quad \hat{\underline{\Theta}}]^T = [\hat{\Psi}_1 \quad \dots \quad \hat{\Psi}_q \quad \hat{\Theta}_1 \quad \dots \quad \hat{\Theta}_r]^T\end{aligned}\quad (2)$$

Bei der Identifikation wird dieser Parametervektor $\hat{\underline{w}}$ so angepasst, dass der Ausgang des strukturierten rekurrenten Netzes \hat{y} möglichst genau dem Systemausgang y folgt, bzw. dass der Ausgangsfehler $e(\hat{\underline{w}}) = y - \hat{y}$ möglichst klein wird. Um eine stabile Identifikation zu ermöglichen, ist nach Hintz [1] die Einführung von Beobachterrückführungen erforderlich. Das Einschwingverhalten der Modellzustände $\hat{\underline{x}}$ auf die Systemzustände \underline{x} kann über die konstanten Beobachterrückführkoeffizienten $\tilde{\underline{l}}$ vorgegeben werden. In der Dissertation von Hintz ist ebenfalls beschrieben, wie diese Beobachterrückführkoeffizienten auszulegen sind.

Die nichtlineare diskrete Zustandsbeschreibung des strukturierten rekurrenten Netzes erhält man somit in zwei Schritten. Zuerst sollen die Beobachterrückführkoeffizienten $\tilde{\underline{l}}$ berücksichtigt werden. Diese kontinuierliche Beschreibung des strukturierten rekurrenten Netzes folgt aus der nichtlinearen Zustandsbeschreibung der Gleichungen (1a) und (1b). Die geschätzten Werte dieser Beschreibung werden mit einer Tilde gekennzeichnet:

$$\dot{\hat{\underline{x}}} = \left(\tilde{\underline{A}} - \tilde{\underline{l}} \cdot \tilde{\underline{c}}^T \right) \cdot \hat{\underline{x}} + \tilde{\underline{b}} \cdot u + \tilde{\underline{l}} \cdot y + \tilde{\underline{K}} \cdot \tilde{\underline{N}}\mathcal{L}(u, \hat{\underline{x}}) \quad (3a)$$

$$\dot{\hat{y}} = \tilde{\underline{c}}^T \cdot \hat{\underline{x}} \quad (3b)$$

Im zweiten Schritt wird, wie oben beschrieben, in die diskrete Zustandsform nach Euler vorwärts übergegangen. Die geschätzten Werte des diskretisierten strukturierten rekurrenten Netzes sollen mit dem Hut-Symbol markiert werden:

$$\hat{\underline{x}}_{k+1} = \hat{\underline{A}}_{rek} \cdot \hat{\underline{x}}_k + \hat{\underline{b}} \cdot u_k + \hat{\underline{l}} \cdot y_k + \hat{\underline{K}} \cdot \widehat{\underline{N}}\mathcal{L}(u_k, \hat{\underline{x}}_k) \quad (4a)$$

$$\hat{y}_k = \hat{\underline{c}}^T \cdot \hat{\underline{x}}_k \quad (4b)$$

mit $\hat{\underline{A}}_{rek} = h \cdot (\tilde{\underline{A}} - \tilde{\underline{l}} \cdot \tilde{\underline{c}}^T) + \underline{I}$ (\underline{I} bezeichnet die Einheitsmatrix), $\hat{\underline{b}} = h \cdot \tilde{\underline{b}}$, $\hat{\underline{l}} = h \cdot \tilde{\underline{l}}$, $\hat{\underline{K}} = h \cdot \tilde{\underline{K}}$, $\widehat{\underline{N}}\mathcal{L} = \tilde{\underline{N}}\mathcal{L}$ und $\hat{\underline{c}} = \tilde{\underline{c}}$. Diese Umformung ergibt sich für kleine Abtastzeiten h , da dann für die Zustände $\hat{\underline{x}}_k = \underline{x}_k$ gilt und für die Integratoreingänge $\hat{\underline{u}}_k = (\tilde{\underline{A}} - \tilde{\underline{l}} \cdot \tilde{\underline{c}}^T) \cdot \underline{x} + \tilde{\underline{b}} \cdot u + \tilde{\underline{l}} \cdot y + \tilde{\underline{K}} \cdot \tilde{\underline{N}}\mathcal{L}(u, \underline{x})$. Im strukturierten rekurrenten Netz sind in $\tilde{\underline{A}}$, $\tilde{\underline{b}}$, $\tilde{\underline{c}}$, $\tilde{\underline{K}}$ und $\tilde{\underline{N}}\mathcal{L}$ die selben Elemente besetzt wie im realen System in \underline{A} , \underline{b} , \underline{c} , \underline{K} und $\underline{N}\mathcal{L}$. Die Werte der Elemente können aber unterschiedlich sein und entsprechen denen des Parametervektors $\hat{\underline{w}}$. Eine Ausnahme bildet der Auskopplungsvektor \underline{c} , da bei der Berechnung der partiellen Ableitungen (siehe Abschnitt 5) davon ausgegangen wird, dass die Elemente des Auskopplungsvektors bekannt sind, d.h. $\tilde{\underline{c}} = \underline{c}$. Diese Voraussetzung ist bei realen Systemen praktisch immer erfüllt, da der Systemausgang bei realen Anwendungen einem Zustand des Systems entspricht.

3 Lernverfahren 1. Ordnung: Der Gradientenabstieg

Da bei strukturierten rekurrenten Netzen das Lernverfahren direkt in die Identifikationsstruktur des rekurrenten Netzes eingebunden ist, ist eine Online-Optimierung der Parameter notwendig. Das online-fähige Gradientenabstiegsverfahren ist aufgrund seiner einfachen Implementierbarkeit ein häufig eingesetztes Optimierungsverfahren, auch wenn

das Verfahren eine sehr langsame Parameterkonvergenzgeschwindigkeit besitzt. Der Lernalgorithmus minimiert bei der Identifikation den Fehler zwischen Systemausgang y_k und Modellausgang $\hat{y}_k(\hat{w}_k)$, indem die Parameter \hat{w}_k des strukturierten rekurrenten Netzes geeignet angepasst werden. Die Definition des Ausgangsfehlers $e_k(\hat{w}_k)$ erfolgt als Differenz zwischen dem System- und dem Modellausgang.

$$e_k(\hat{w}_k) = y_k - \hat{y}_k(\hat{w}_k) \quad (5)$$

Durch Einführen des quadratischen Fehlermaßes

$$F_k(\hat{w}_k) = \frac{1}{2} (y_k - \hat{y}_k(\hat{w}_k))^2 \quad (6)$$

lautet das zu lösende Optimierungsproblem:

$$\hat{w}_{opt,k} = \min_{\hat{w}_k} F_k(\hat{w}_k) \quad (7)$$

Für die Parameteränderung ergibt sich nach dem Gradientenabstiegsverfahren in diskreter Schreibweise das bekannte Lerngesetz:

$$\begin{aligned} \hat{w}_{k+1} - \hat{w}_k - \eta \cdot \left[\frac{\partial E_k(\hat{w}_k)}{\partial \hat{w}_1} \quad \dots \quad \frac{\partial E_k(\hat{w}_k)}{\partial \hat{w}_p} \right] \\ = \hat{w}_k - \eta \cdot \nabla E_k(\hat{w}_k) \end{aligned} \quad (8)$$

mit der an das Problem angepassten Lernschrittweite $\eta > 0$ und dem Gradienten des Fehlers $\nabla F_k(\hat{w}_k)$. Der Gradient $\nabla F_k(\hat{w}_k)$ in Gleichung (8) kann entsprechend der Definition des Fehlermaßes aus Gleichung (6) wie folgt zerlegt werden:

$$\nabla F_k(\hat{w}_k) = \frac{1}{2} \cdot 2 \cdot (y_k - \hat{y}_k(\hat{w}_k)) \cdot (-\nabla \hat{y}_k(\hat{w}_k)) = -e_k(\hat{w}_k) \cdot \nabla \hat{y}_k(\hat{w}_k) \quad (9)$$

Das Lerngesetz nach dem Gradientenabstiegsverfahren lautet damit:

$$\hat{w}_{k+1} - \hat{w}_k + \underbrace{\eta \cdot e_k(\hat{w}_k) \cdot \nabla \hat{y}_k(\hat{w}_k)}_{\Delta \hat{w}_k} \quad (10)$$

Das in Gleichung (10) angegebene Lerngesetz kann erweitert werden: Für jeden Parameter \hat{w}_i (i bezeichnet ein Element des Gewichtsvektors und keine Zeit) wird eine eigene Lernschrittweite η_i festgelegt. Zusätzlich wird für jeden Parameter \hat{w}_i ein geeigneter Momentumterm mit $0 \leq \alpha_i \leq 1$ eingeführt:

$$\hat{w}_{k+1} = \hat{w}_k + e_k(\hat{w}_k) \cdot \text{diag}(\underline{\eta}) \cdot \nabla \hat{y}_k(\hat{w}_k) + \text{diag}(\underline{\alpha}) \cdot \Delta \hat{w}_{k-1} \quad (11)$$

$\text{diag}(\underline{\eta})$ bzw. $\text{diag}(\underline{\alpha})$ sind Diagonalmatrizen mit den Elementen des Lernschrittweitenvektors $\underline{\eta}$ bzw. des Momentumtermvektors $\underline{\alpha}$ in den Hauptdiagonalen.

4 Lernverfahren 2. Ordnung: Das Quasi-Online Newtonverfahren

Der Große Nachteil des Gradientenverfahrens ist die langsame Konvergenzgeschwindigkeit der Parameter, die stark von den Lernparametern $\underline{\eta}$ und $\underline{\alpha}$, sowie von der Anfangsinitialisierung der Gewichte \hat{w}_0 abhängig ist. Zudem erfordert die Wahl der Lernschrittweiten und der Momentumterme zusätzliches Vorwissen. Aus diesen Gründen wird im Folgenden das Gradientenabstiegsverfahren durch ein nichtlineares lokales Newtonverfahren

2. Ordnung ersetzt. Die Newtonverfahren sind im allgemeinen jedoch nicht onlinefähig. Die Onlinefähigkeit des Lernverfahrens ist aber eine Voraussetzung für die Identifikation des strukturierten rekurrenten Netzes. Deshalb wird in diesem Abschnitt die Vorgehensweise bei der Identifikation mit dem Quasi-Online Newtonverfahren vorgestellt, welches analog zum Gauß-Newton Verfahren den Gradienten und die Hesse-Matrix über die Jacobi-Matrix bestimmt und auch für strukturierte rekurrente Netze eingesetzt werden kann.

4.1 Newton-Algorithmus

Im Unterschied zum Gradientenabstiegsverfahren, das über die Berechnung des Gradienten die Parameteränderung ausführt, benötigen die Newtonverfahren die Hesse-Matrix, also die zweite Ableitung der Fehlerfläche. Beim Newton Algorithmus erfolgt die Definition des Gütefunktionals gemäß dem kumulierten quadratischen Fehlermaßes, bei dem der Ausgangsfehler $e(\hat{\underline{w}}) = y - \hat{y}(\hat{\underline{w}})$ quadratisch über N -Trainingspaare aufsummiert wird.

$$F(\hat{\underline{w}}) = \frac{1}{2} \sum_{j=1}^N e_j^2(\hat{\underline{w}}) \quad (12)$$

Erst nach Auswertung dieser N -Trainingspaare findet eine Gewichtsanzpassung statt, der Gewichtsvektor $\hat{\underline{w}}$ ist also während der Auswertung der Trainingsdaten konstant. Damit lässt sich das zu lösende offline-Optimierungsproblem angeben:

$$\hat{\underline{w}}_{opt} = \min_{\hat{\underline{w}}} E(\hat{\underline{w}}) \quad (13)$$

Das nichtlineare mehrdimensionale Gütefunktional $F(\hat{\underline{w}})$ formt eine Fehlerfläche, die von den Gewichten $\hat{\underline{w}}$ des strukturierten rekurrenten Netzes abhängig ist. Die aktuelle Position auf der Fehlerfläche zu einem bestimmten Optimierungszeitpunkt k soll durch einen entsprechenden Index gekennzeichnet werden. $\hat{\underline{w}}_{[k]}$ ist die Position auf der Fehlerfläche nach dem $[k]$ -ten Optimierungsschritt. Da nicht zu jedem Abtastzeitpunkt ein Optimierungsschritt ausgeführt wird, soll die Optimierungszeit mit einer eckigen Klammer gekennzeichnet werden. Damit sollen Verwechslungen mit der Abtastzeit vermieden werden, die weiterhin mit einem Index ohne Klammer bezeichnet ist. Das Newtonverfahren nähert nun die Umgebung des aktuellen Fehlerflächenpunktes $\hat{\underline{w}}_{[k]}$ durch eine Taylor-Approximation 2. Ordnung an:

$$F(\hat{\underline{w}}) = F(\hat{\underline{w}}_{[k]} | \Delta \hat{\underline{w}}) \approx F(\hat{\underline{w}}_{[k]} | \underline{g}_{[k]}^T \Delta \hat{\underline{w}} | \frac{1}{2} \Delta \hat{\underline{w}}^T \mathbf{H}_{[k]} \Delta \hat{\underline{w}} \quad (14)$$

Dabei bezeichnen $\underline{g}_{[k]}$ den Gradienten der Fehlerfläche am Entwicklungspunkt $\hat{\underline{w}}_{[k]}$

$$\underline{g}_{[k]} = \nabla E(\hat{\underline{w}}_{[k]}) = \left[\frac{\partial E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_1} \quad \frac{\partial E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_2} \quad \dots \quad \frac{\partial E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_p} \right]^T \quad (15)$$

und $\mathbf{H}_{[k]}$ die Hesse-Matrix.

$$\mathbf{H}_{[k]} = \nabla^2 E(\hat{\underline{w}}_{[k]}) = \begin{bmatrix} \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_1^2} & \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_1 \partial \hat{w}_2} & \dots & \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_1 \partial \hat{w}_p} \\ \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_2 \partial \hat{w}_1} & \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_2^2} & \dots & \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_2 \partial \hat{w}_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_p \partial \hat{w}_1} & \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_p \partial \hat{w}_2} & \dots & \frac{\partial^2 E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_p^2} \end{bmatrix} \quad (16)$$

Das Prinzip des Newtonverfahren liegt nun darin, das Minimum dieser quadratischen Fehlerflächen-Approximation zu finden. Dazu wird die Ableitung von Gleichung (14) nach den Gewichtsänderungen $\Delta \hat{w}$ gebildet und zu Null gesetzt:

$$\underline{g}_{|k|} + \mathbf{H}_{|k|} \Delta \hat{w} = 0 \quad (17)$$

Durch Umformen nach $\Delta \hat{w}$ ergibt sich der stationäre Punkt der quadratischen Fehlerflächenapproximation. Der gefundene Vektor zeigt in die so genannte Newton-Richtung.

$$\Delta \hat{w} = -\mathbf{H}_{|k|}^{-1} \underline{g}_{|k|} \quad (18)$$

Aus dem aktuellen Entwicklungspunkt und dem berechneten stationären Punkt ergibt sich der neue Entwicklungspunkt für den nächsten Optimierungsschritt. Die iterative Lösung der Optimierungsaufgabe nach dem Newton Algorithmus lautet somit:

$$\hat{w}_{|k|+1} = \hat{w}_{|k|} + \underbrace{-\mathbf{H}_{|k|}^{-1} \underline{g}_{|k|}}_{\text{Newton-Richtung}} \quad (19)$$

Der gefundene stationäre Punkt muss aber nicht zwingend ein Minimum sein. Die Ableitung ist auch bei einem Sattelpunkt oder bei einem Maximum Null! Damit das Optimierungsverfahren stabil arbeitet, muss die Abstiegsbedingung

$$F(\hat{w}_{|k|+1}) < F(\hat{w}_{|k|}) \quad (20)$$

erfüllt sein. Diese ist erfüllt, wenn die Hesse Matrix $\mathbf{H}_{|k|}$ positiv definit ist, wenn also alle Eigenwerte der Matrix größer Null sind $\lambda_j > 0 \forall j$ (dann ist die quadratische Approximation der Fehlerfläche nach oben geöffnet). Damit ist gewährleistet, dass das Verfahren gegen ein Minimum konvergiert. Da die Berechnung der Hesse-Matrix sehr aufwendig ist, wird bei der Implementierung des Quasi-Online Newtonverfahren die Hesse-Matrix wie beim Gauß-Newton Verfahren [5] über die Jacobimatrix berechnet. Eine genauere Beschreibung dazu folgt im nächsten Abschnitt.

4.2 Der Quasi-Online Newtonalgorithmus

Zunächst soll die Jacobimatrix definiert werden:

$$\mathbf{J}(\hat{w}_{|k|}) = \begin{bmatrix} \frac{\partial c_1(\hat{w}_{|k|})}{\partial \hat{w}_1} & \frac{\partial c_1(\hat{w}_{|k|})}{\partial \hat{w}_2} & \cdots & \frac{\partial c_1(\hat{w}_{|k|})}{\partial \hat{w}_p} \\ \frac{\partial c_2(\hat{w}_{|k|})}{\partial \hat{w}_1} & \frac{\partial c_2(\hat{w}_{|k|})}{\partial \hat{w}_2} & \cdots & \frac{\partial c_2(\hat{w}_{|k|})}{\partial \hat{w}_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_N(\hat{w}_{|k|})}{\partial \hat{w}_1} & \frac{\partial c_N(\hat{w}_{|k|})}{\partial \hat{w}_2} & \cdots & \frac{\partial c_N(\hat{w}_{|k|})}{\partial \hat{w}_p} \end{bmatrix} \quad (21)$$

Die Jacobi-Matrix enthält die partiellen Ableitungen des Fehlers ϵ nach den p Parametern des rekurrenten Netzes zu N verschiedenen Zeitschritten. Da die einzelnen Zeitschritte nicht den Optimierungsschritten entsprechen, sind diese Indizes wieder ohne eckige Klammern bezeichnet.

Mit Gleichung (12) kann das i -te Element des Gradienten berechnet werden:

$$[\nabla F(\hat{w}_{|k|})]_i = \frac{\partial F(\hat{w}_{|k|})}{\partial \hat{w}_i} = \sum_{j=1}^N e_j(\hat{w}_{|k|}) \frac{\partial e_j(\hat{w}_{|k|})}{\partial \hat{w}_i} \quad (22)$$

Das gleiche Ergebnis erhält man, wenn die transponierte Jacobimatrix mit dem Fehlervektor (der Fehlervektor enthält N verschiedene Zeitschritte) multipliziert wird:

$$\mathbf{J}^T(\hat{\underline{w}}_{[k]}) \cdot \underline{e}(\hat{\underline{w}}_{[k]}) = \begin{bmatrix} \frac{\partial E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_1} - \sum_{j=1}^N e_j(\hat{\underline{w}}_{[k]}) \frac{\partial e_j(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_1} \\ \vdots \\ \frac{\partial E(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_p} - \sum_{j=1}^N e_j(\hat{\underline{w}}_{[k]}) \frac{\partial e_j(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_p} \end{bmatrix} \quad (23)$$

In Matrix-Vektor Schreibweise gilt somit für den Gradienten:

$$\nabla F(\hat{\underline{w}}_{[k]}) = \mathbf{J}^T(\hat{\underline{w}}_{[k]}) \cdot \underline{e}(\hat{\underline{w}}_{[k]}) \quad (24)$$

Aus Gleichung (22) berechnet sich das l, i -te Element der Hesse-Matrix nach der Produktregel der Differentiation zu

$$[\nabla^2 E(\hat{\underline{w}}_{[k]})]_{l,i} = \sum_{j=1}^N \left(\frac{\partial e_j(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_l} \frac{\partial e_j(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_i} + e_j(\hat{\underline{w}}_{[k]}) \frac{\partial^2 e_j(\hat{\underline{w}}_{[k]})}{\partial \hat{w}_l \partial \hat{w}_i} \right) \quad (25)$$

Ist der Fehler klein, so ist der rechte Ausdruck in der großen Klammer viel kleiner als der linke und kann vernachlässigt werden. Da der linke Ausdruck dem Matrixprodukt aus transponierter Jacobimatrix und Jacobimatrix entspricht, kann unter dieser Annahme vereinfacht in Matrix-Vektor-Form geschrieben werden:

$$\mathbf{H}_{[k]} \approx \mathbf{J}^T(\hat{\underline{w}}_{[k]}) \mathbf{J}(\hat{\underline{w}}_{[k]}) \quad (26)$$

Setzt man den Gradienten aus Gleichung (24) und die Hesse-Matrix aus Gleichung (26) in die Berechnungsvorschrift des Newton Algorithmus aus Gleichung (19) ein, so erhält man die Berechnungsvorschrift für die Optimierung von $F(\hat{\underline{w}})$ nach dem Gauß-Newton Verfahren

$$\hat{\underline{w}}_{[k+1]} = \hat{\underline{w}}_{[k]} - [\mathbf{J}^T(\hat{\underline{w}}_{[k]}) \mathbf{J}(\hat{\underline{w}}_{[k]})]^{-1} \mathbf{J}^T(\hat{\underline{w}}_{[k]}) \underline{e}(\hat{\underline{w}}_{[k]}) \quad (27)$$

welche nun Ausgangspunkt für den folgenden Quasi-Online Newtonalgorithmus sein soll. Der Große Vorteil dieser Berechnungsvorschrift besteht darin, dass bei der Optimierung keine zweiten Ableitungen berechnet werden müssen. Das Verfahren ist jedoch nicht onlinefähig, da die Jacobimatrix $\mathbf{J}(\hat{\underline{w}}_{[k]})$ die partiellen Ableitungen des Fehlers zu N verschiedenen Zeitschritten verlangt (siehe Gleichung (21)). Somit ist das Optimierungsverfahren in dieser Form nicht zur Identifikation mit strukturierten rekurrenten Netzen geeignet. Der Übergang von einem offline-Lernverfahren zur Onlinefähigkeit wird durch eine geeignete Fensterung über N -Abtastschritte erreicht. Zu jedem Abtastschritt werden die partiellen Ableitungen des strukturierten rekurrenten Netzes (siehe Abschnitt 5) berechnet und in die entsprechende Zeile der Jacobimatrix geschrieben. Jede Zeile der Jacobimatrix enthält die Ableitungen zu einer bestimmten Zeit. Bild 3 zeigt bildhaft diese Vorgehensweise. In der ersten Zeile der Jacobimatrix befinden sich die partiellen Ableitungen des strukturierten rekurrenten Netzes zum ersten Zeitschritt des Fensters. Die letzte Zeile der Jacobimatrix beinhaltet die Ableitungen zum Zeitschritt N des aktuellen Zeitfensters. In gleicher Weise wird der Fehlervektor \underline{e} angelegt. Auch hier befindet sich in der ersten Zeile der älteste Fehlerwert und in der untersten Zeile der aktuellste. Nach N Abtastschritten werden Jacobimatrix und Fehlervektor dem Optimierungsalgorithmus übergeben. Dieser berechnet mit den gesammelten Daten genau einen Optimierungsschritt und führt eine Parameteranpassung durch. Nach erfolgter Optimierung wird das Fenster vollständig durch

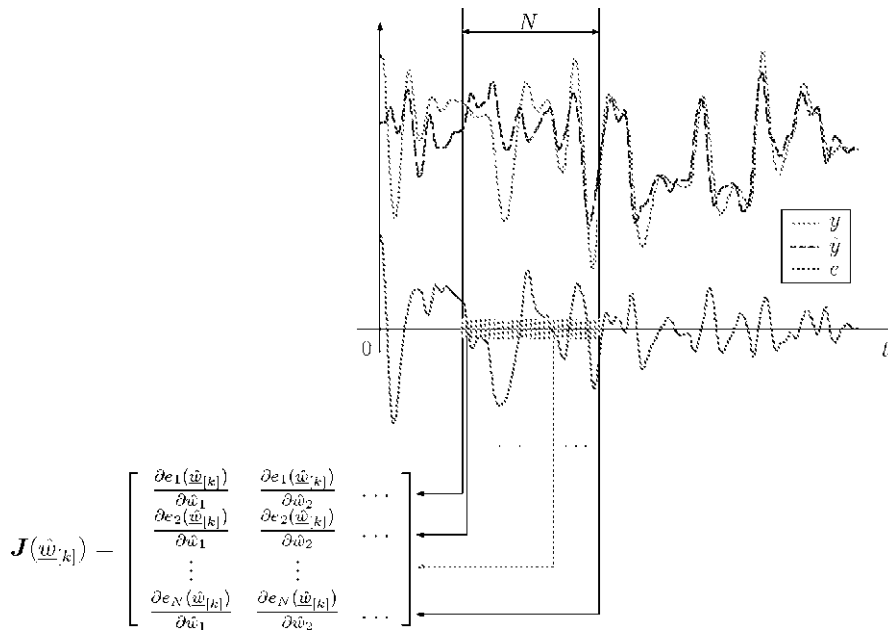


Bild 3: Prinzip des Quasi-Online Newtonverfahrens

N neue Trainingsdaten beschrieben, erst dann findet ein weiterer Optimierungsschritt statt. Bild 3 zeigt einen Ausschnitt des Verlaufs des Systemausgangs y und des Modellausgangs \hat{y} , sowie den zugehörigen Verlauf des Ausgangsfehlers e . Die Grafik veranschaulicht den beschriebenen Vorgang zur Online-Bestimmung der Jacobimatrix. Die Berechnungsvorschrift aus Gleichung (27) wird mit einem Dämpfungsparameter μ erweitert, mit dem die Konvergenzgeschwindigkeit des Lernverfahrens eingestellt werden kann:

$$\hat{\mathbf{w}}_{|k+1|} = \hat{\mathbf{w}}_{|k|} - [\mathbf{J}^T(\hat{\mathbf{w}}_{|k|})\mathbf{J}(\hat{\mathbf{w}}_{|k|}) + \mu\mathbf{I}]^{-1} \mathbf{J}^T(\hat{\mathbf{w}}_{|k|})\mathbf{e}(\hat{\mathbf{w}}_{|k|}) \quad (28)$$

\mathbf{I} ist dabei die Einheitsmatrix. Mit dem Dämpfungsparameter μ kann die Geschwindigkeit des Lernverfahrens an die Dynamik des Beobachters angepasst werden. Es muss gewährleistet sein, dass die Gewichtsveränderungen (verursacht durch den Optimierungsalgorithmus) nicht zu schnell sind, so dass der Beobachter im strukturierten rekurrenten Netz in der Lage ist, die Systemzustände nachzuführen. Der Parameter μ im Lerngesetz des Quasi-Online Newtonverfahrens (28) ermöglicht einen kontinuierlichen Übergang zwischen dem langsamen Gradientenverfahren und dem Gauß-Newton Verfahren, da:

- μ sehr groß \Rightarrow Gradientenabstiegsverfahren mit der Lernschrittweite $\frac{1}{\mu}$

$$\hat{\mathbf{w}}_{k+1} \approx \hat{\mathbf{w}}_k - \frac{1}{\mu} \mathbf{J}^T(\hat{\mathbf{w}}_k)\mathbf{e}(\hat{\mathbf{w}}_k) = \hat{\mathbf{w}}_k - \frac{1}{\mu} \nabla F(\hat{\mathbf{w}})$$

- μ sehr klein \Rightarrow Gauß-Newton Verfahren

Wie oben bereits beschrieben, garantiert das Lerngesetz des Quasi-Online Newtonverfahrens (28) nicht die Konvergenz auf ein Minimum. Ein Optimierungsschritt kann auch zu einem Sattelpunkt oder Maximum führen. Um dies zu vermeiden, wird eine Validierung eingeführt. Parallel zum strukturierten rekurrenten Netz läuft ein Validierungsnetz mit den alten Gewichten. Führen die neuen Gewichte beim strukturierten rekurrenten Netz zu einem größeren Fehler wie beim Validierungsmodell, so wurde bei der Optimierung kein

Minimum gefunden und der letzte Optimierungsschritt wird rückgängig gemacht. In der Praxis stellt man jedoch fest, dass diese Korrektur nur sehr selten vorkommt, und dass die Quasi-Online Newtonoptimierung auch ohne Validierung stabil läuft, was durch den Einfluss der Beobachterrückführungen im strukturierten rekurrenten Netz zu erklären ist. Für eine erfolgreiche Optimierung ist es wichtig, dass in einem Fenster ausreichend Information über das System enthalten ist. Je größer das Fenster N gewählt wird, um so mehr Information über das Systemverhalten wird bei der Ausführung eines Optimierungsschrittes berücksichtigt. Dagegen liegt bei einem großen Fensters viel Zeit zwischen den einzelnen Optimierungsschritten, in denen überhaupt nicht optimiert wird. Bei der Identifikation des Plattenaufbaus zeigte sich, dass $50 < N < 500$ ein vernünftiger Wertebereich ist.

5 Berechnung der partiellen Ableitungen

Beim Quasi-Online Newtonverfahren müssen für die Jacobimatrix (21) die partiellen Ableitungen des Fehlers nach allen Parametern $\partial e_j / \partial \hat{w}_i$ (mit $1 \leq j \leq N$, $1 \leq i \leq p$) berechnet werden. Da mit der Definition des Fehlers $e = y - \hat{y}$ der Zusammenhang $\partial e_j / \partial \hat{w}_i = -\partial \hat{y}_j / \partial \hat{w}_i$ gilt, reduziert sich die Berechnung, wie beim Lerngesetz des Gradientenabstiegsverfahrens (10), auf die Bestimmung der partiellen Ableitungen des Modellausgangs nach den Gewichten unter Beachtung des negativen Vorzeichens. Somit kann im Folgenden einheitlich ein Ausdruck für $\partial \hat{y}_k / \partial \hat{w}$ gesucht werden.

Mit Gleichung (4b) und $\hat{c} = \underline{c}$ (wobei c konstant ist) folgt

$$\frac{\partial \hat{y}_k}{\partial \hat{w}} = \frac{\partial \hat{c}^T}{\partial \hat{w}} \cdot \hat{x} + \hat{c}^T \cdot \frac{\partial \hat{x}}{\partial \hat{w}} = \underline{c}^T \cdot \frac{\partial \hat{x}}{\partial \hat{w}} \quad (29)$$

Die partiellen Ableitungen der Zustände nach den Gewichten erhält man mit der nichtlinearen Zustandsbeschreibung (4a) des strukturierten rekurrenten Netzes:

$$\begin{aligned} \frac{\partial \hat{x}_{k+1}}{\partial \hat{w}} &= \frac{\partial}{\partial \hat{w}} \left(\hat{A}_{rek} \cdot \hat{x}_k \right) + \frac{\partial}{\partial \hat{w}} \left(\hat{b} \cdot u_k \right) + \frac{\partial}{\partial \hat{w}} \left(\hat{l} \cdot y_k \right) + \frac{\partial}{\partial \hat{w}} \left(\hat{K} \cdot \widehat{\mathcal{NL}}(u_k, \hat{x}_k) \right) \\ &= \hat{A}_{rek} \cdot \frac{\partial \hat{x}_k}{\partial \hat{w}} + \frac{\partial \hat{A}_{rek}}{\partial \hat{w}} \cdot \hat{x}_k + \frac{\partial \hat{b}}{\partial \hat{w}} \cdot u_k + \hat{K} \frac{\partial \widehat{\mathcal{NL}}(u_k, \hat{x}_k)}{\partial \hat{w}} + \frac{\partial \hat{K}}{\partial \hat{w}} \widehat{\mathcal{NL}}(u_k, \hat{x}_k) \end{aligned} \quad (30)$$

Wegen der Unabhängigkeit von den wahren Größen u_k und y_k gilt $\partial u_k / \partial \hat{w} = 0$ und $\partial y_k / \partial \hat{w} = 0$. Außerdem sollen die Beobachterkoeffizienten konstant und nicht identifiziert werden, d.h. $\partial \hat{l} / \partial \hat{w} = 0$.

In Gleichung (30) werden die partiellen Ableitungen für den nächsten Abtastschritt $\partial \hat{x}_{k+1} / \partial \hat{w}$ aus den aktuellen partiellen Ableitungen $\partial \hat{x}_k / \partial \hat{w}$ berechnet, wobei für $\partial \hat{x}_0 / \partial \hat{w} = \underline{0}$ gilt, da für die Zeitschritte $k \leq 0$ keine Änderungen der Zustände \hat{x} und der Gewichte \hat{w} erfolgen.

6 Systemidentifikaton

Abschließend soll mit den vorgestellten Lernverfahren und einem geeigneten strukturierten rekurrenten Netz ein dynamisches, nichtlineares mechatronisches System identifiziert

werden. Der zu identifizierende Versuchsaufbau [6] besteht aus einem Gleichstrommotor mit Propeller. Der Propeller erzeugt einen Luftstrom, dieser lenkt eine drehbar gelagerte Plexiglasplatte aus. Zwischen dem Auslenkwinkel der Platte und der Motordrehzahl besteht, wie der Signalflußplan in Bild 4 zeigt, ein nichtlinearer Zusammenhang. Die

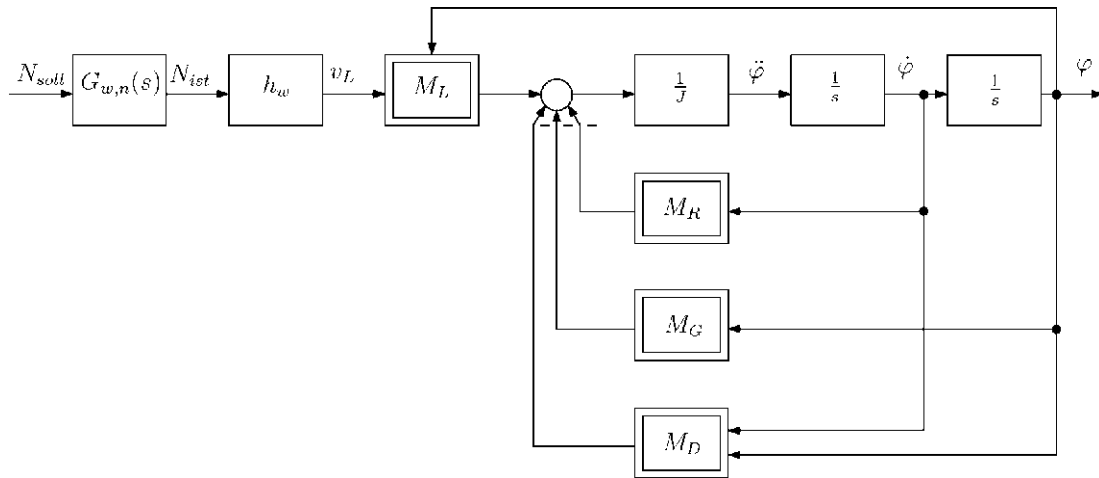


Bild 4: Signalflussplan für das nichtlineare Gesamtmodell des Plattenaufbaus

Führungübertragungsfunktion des geschlossenen Drehzahlregelkreises wird nach dem Betragsoptimum [7] ausgelegt, der Zusammenhang zwischen der Propellerdrehzahl und der Luftströmungsgeschwindigkeit wird linear modelliert: $v_L = h_w \cdot N_{ist}$. Die Bewegungsgleichung des Plattenaufbaus lautet:

$$J\ddot{\varphi} = -M_G - M_D - M_R + M_L$$

Dabei beschreibt J das Massenträgheitsmoment, M_G das Gravitationsmoment, M_D das Dämpfungsmoment, M_R das Reibmoment und M_L das Luftstrahlmoment. Für die einzelnen Momentenanteile der Bewegungsgleichung gilt:

$$M_L = \frac{1}{2}\rho_L b l^2 v_L^2 \cos \varphi \quad M_G = l_s m g \sin \varphi \quad M_D = \frac{1}{8}c_w \rho_L b l^4 \dot{\varphi}^2 \cos \varphi$$

$$M_R = \mu_1 (\tanh(\mu_2 \dot{\varphi}) - \tanh(\mu_3 \dot{\varphi})) + \mu_4 \tanh(\mu_5 \dot{\varphi}) + \mu_6 \dot{\varphi}$$

Für das Reibmoment wird ein nichtlineares, stetig differenzierbares Reibmodell nach [8] herangezogen, um das Gesamtmodell linearisieren zu können (die Linearisierung ist für die Auslegung der Beobachterkoeffizienten erforderlich). Durch die Parameter $\mu_1 \dots \mu_6$ können die Haftreibung, der Stribeck Effekt, die Coulomb Reibung und eine geschwindigkeitsabhängige Gleitreibung berücksichtigt werden.

Die **simulative Identifikation** erfolgt mit dem Gradientenabstiegsverfahren und mit dem in Kapitel 4.2 vorgestellten Quasi-Online Newtonverfahren. Identifiziert wird das Systemverhalten zwischen der Istdrehzahl des Propellers N_{ist} als Systemeingang und dem Plattenwinkel φ als Systemausgang (siehe Abbildung 4). Der Drehzahlregelkreis dient lediglich zur Vorgabe eines für die Identifikation geeigneten Drehzahlverlaufs im Bereich von $N = 100[1/s]$ bis $200[1/s]$. Die Daten des Systems sind in Tabelle 1 zusammengefasst. Das Dämpfungsmoment M_D wird beim Entwurf des strukturierten rekurrenten Netzes vernachlässigt. Diese Vereinfachung ist notwendig, da sonst keine eindeutige Identifikationsstruktur vorliegt. Würde das Dämpfungsmoment berücksichtigt, wäre es nicht möglich, das Reibmoment und das Gravitationsmoment getrennt vom Dämpfungsmoment zu identifizieren, da alle Momente auf den gleichen Summationspunkt wirken. Die

	Parameter	Wert	Einheit
wirksame Propellersteigung	h_w	0.0352	[m]
Luftdichte	ρ_L	1.29	[kg/m ³]
Erdbeschleunigung	g	9.81	[m/s ²]
Luftwiderstandsbeiwert	c_w	1.1	dimensionslos
Gesamtlänge der Platte	l	190	[mm]
Gesamtbreite der Platte	b	173	[mm]
Lage des Schwerpunktes	l_S	120	[mm]
Gesamtmasse der Platte	m	0.056	[kg]
Massenträgheitsmoment	J	0.00101	[kgm ²]
Parameter der Reibkennlinie	μ_1	0.001	[Nm]
	μ_2	1	[Nm/rads ⁻¹]
	μ_3	0.05	[Nm/rads ⁻¹]
	μ_4	0.004	[Nm]
	μ_5	3.7	[Nm/rads ⁻¹]
	μ_6	0.003	[Nm/rads ⁻¹]

Tabelle 1: Daten für die simulative Identifikation

Dämpfung (siehe Abbildung 4) ist wie das Gravitationsmoment vom Plattenwinkel und wie das Reibungsmoment von der Winkelgeschwindigkeit der Platte abhängig. Am realen Versuchsaufbau hat das Dämpfungsmoment M_D einen sehr kleinen Einfluss [9], so dass sich die Modellgüte nur wenig verschlechtert. Das bei der Identifikation verwendete strukturierte rekurrente Netz in diskreter Form ist in Abbildung 5 dargestellt. Es besteht aus zwei eindimensionalen GRNN zur Approximation des Luftstrahlmomentes und der Reibung sowie aus zwei linearen Parametern zur Approximation des Massenträgheitsmomentes der Platte $\hat{\Psi}_1$ und der Gravitationskonstante $\hat{\Psi}_2$. Eine besondere Eigenschaft dieser Identifikationsstruktur stellt das GRNN am Systemeingang dar. Durch das anfangs ungelernete GRNN gelangt zu Beginn der Identifikation „falsche“ Information in das rekurrente Netz und erschwert die Identifikation der weiteren Parameter.

Nach Festlegen der Größen $u = N_{ist}$, $y = \varphi$ und $\underline{x} = [x_1 \ x_2]^T = [\varphi \ \Omega]^T$ können die kontinuierlichen Zustandsgleichungen des Systems angegeben werden:

$$\begin{aligned} \dot{\underline{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \underline{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{J} & -\frac{1}{J} & -\frac{1}{J}l_Smg \end{bmatrix} \cdot \begin{bmatrix} \mathcal{NL}_{M'_L}(u) \cdot \cos x_1 \\ \mathcal{NL}_{M_R}(x_2) \\ \sin x_1 \end{bmatrix} \\ y &= [1 \ 0] \cdot \underline{x} \end{aligned}$$

Entsprechend bezeichnen in der diskreten Zustandsbeschreibung des strukturierten rekurrenten Netzes (Bild 5) die Größen $u_k = N_{ist,k}$, $\hat{y}_k = \hat{\varphi}_k$ und $\hat{\underline{x}}_k = [\hat{x}_{1,k} \ \hat{x}_{2,k}]^T = [\hat{\varphi}_k \ \hat{\Omega}_k]^T$. Zusätzlich werden der Parametervektor $\hat{\underline{w}} = [\hat{\Psi}_1 \ \hat{\Psi}_2 \ \hat{\Theta}_{M'_L}^T \ \hat{\Theta}_{M_R}^T]^T$ (mit seinen linearen Parametern $\hat{\Psi}_1 = \frac{1}{J}$, $\hat{\Psi}_2 = l_Smg$ und seinen beiden Gewichtsvektoren der GRNN $\hat{\Theta}_{M'_L}^T$, $\hat{\Theta}_{M_R}^T$) und der Vektor der Beobachterkoeffizienten $\tilde{\underline{l}} = [\tilde{l}_1 \ \tilde{l}_2]^T$ definiert. Damit ergeben sich die diskreten Zustandsgleichungen des strukturierten rekurrenten Net-

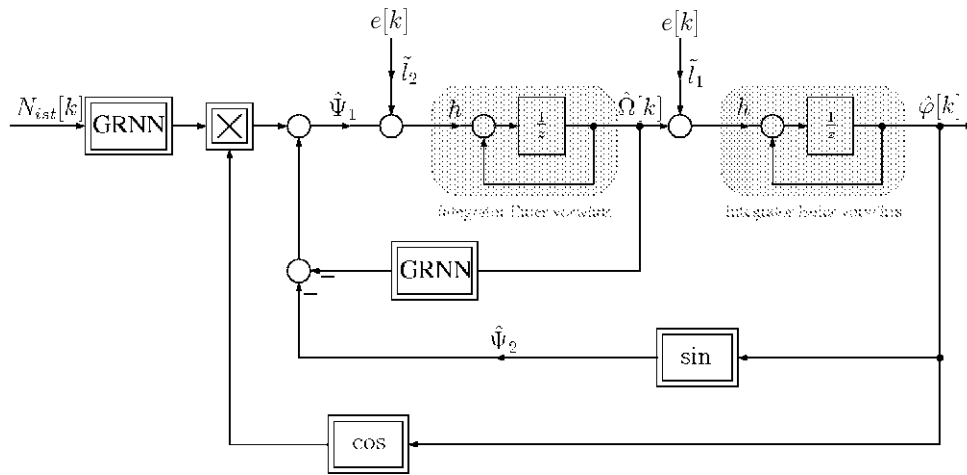


Bild 5: Strukturiertes rekurrentes Netz des Plattenaufbaus

zes zu:

$$\hat{\underline{x}}_{k+1} = \begin{bmatrix} 1 - h\tilde{l}_1 & h \\ -h\tilde{l}_2 & 1 \end{bmatrix} \cdot \hat{\underline{x}}_k + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot u_k + \begin{bmatrix} h\tilde{l}_1 \\ h\tilde{l}_2 \end{bmatrix} \cdot y_k$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ h\hat{\Psi}_1 & -h\hat{\Psi}_1 & -h\hat{\Psi}_1\hat{\Psi}_2 \end{bmatrix} \cdot \begin{bmatrix} \hat{y}_{M_L}(u_k) \cdot \cos(\hat{x}_{1,k}) \\ \hat{y}_{M_R}(\hat{x}_{2,k}) \\ \sin(\hat{x}_{1,k}) \end{bmatrix}$$

$$\hat{y}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \hat{\underline{x}}_k$$

Wie das Bild 5 zeigt, ist der lineare Parameter des Massenträgheitsmomentes $\hat{\Psi}_1$ nicht unabhängig von der Identifikation des Luftstrahlmoments M_L , des Reibmoments M_R und des Gravitationsmoments M_G . Diese drei Momentenanteile wirken auf den gleichen Summationspunkt und werden anschließend mit dem Wert des Massenträgheitsmomentes $\hat{\Psi}_1$ multipliziert. Veränderungen beim Massenträgheitsmoment können der Gravitationskonstante $\hat{\Psi}_2$ und den beiden nichtlinearen Kennlinien zugeschlagen werden. Ein falscher Wert $\hat{\Psi}_1$ führt bei gleichem Ein-Ausgangsverhalten des strukturierten rekurrenten Netzes zu einem kleineren oder größeren Wert bei der Gravitationskonstante $\hat{\Psi}_2$ und zu einem gestreckten oder gestauchten Identifikationsergebnis bei den beiden Kennlinien Luftstrahlmoment M_L und Reibmoment M_R . Um eine unabhängige Identifikation der Parameter und Nichtlinearitäten zu erreichen, wird der lineare Parameter des Massenträgheitsmomentes $\hat{\Psi}_1$ nicht adaptiert. Da das Gewicht und die Maße der Platte bekannt sind, ist dieser Schritt für die Identifikation zulässig. Die Identifikationsstruktur verliert einen Freiheitsgrad und ermöglicht somit die unabhängige Bestimmung der einzelnen Systemparameter. Aus der diskreten Zustandsbeschreibung werden nach den Gleichungen (29) und (30) die für das Lerngesetz und die Parameteradaption notwendigen partiellen Ableitungen $\partial\hat{y}_k/\partial\hat{\underline{w}}$ berechnet. Damit sind zusammen mit dem Gradienten-Lerngesetz nach Gleichung (11) und dem Quasi-Online Newton-Lerngesetz nach Gleichung (28) alle erforderlichen Gleichungen zur Parameteridentifikation des Plattenaufbaus bekannt.

Bei der simulativen Identifikation stimmen die Struktur der Strecke und des rekurrenten Netzes genau überein. Die simulative Identifikation eignet sich zum einen, um einen objektiven Vergleich zwischen den beiden in den Kapiteln 3 und 4.2 vorgestellten Lernverfahren durchzuführen, und zum anderen, um geeignete Lernparameter für die spätere Identifikation an der realen Versuchsanlagen zu bekommen. Die Abtastzeit beträgt $h = 5[m.s]$. Die Anregung des Systems erfolgt mit einem Amplitudenmodulierten Pseudo

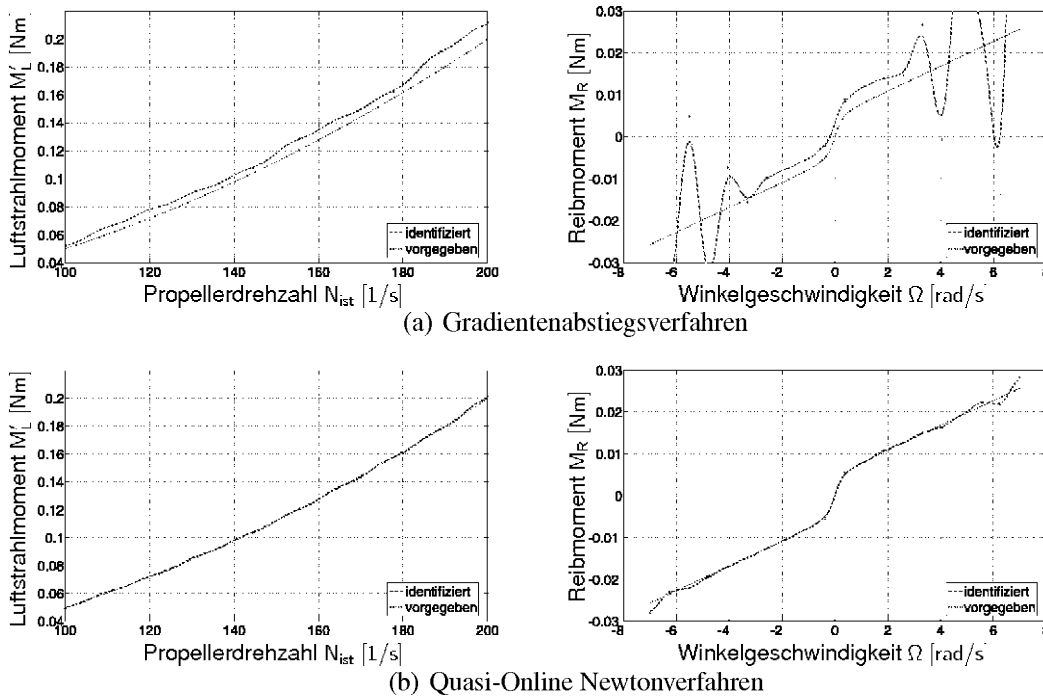


Bild 6: Luftstrahlmoments M_L' und Reibmoments M_R nach einer Identifikation von 60 Sekunden

Rausch Binär Signal (kurzs APRBS). Das APRB-Signal ist ein rechteckförmiges Anregungssignal, das sich dadurch auszeichnet, dass sich die Amplitude und die Pulsbreite des Signals zufällig zwischen vorgebbaren Grenzen ändert. Das rechteckförmige Anregungssignal mit unterschiedlicher Haltezeit und Amplitude soll garantieren, dass alle relevanten Frequenzen und Amplituden des Prozesses ausreichend angeregt werden. Im Falle des Plattenaufbaus wird über den Drehzahlregler ein APRB-Drehzahlssignal mit einer Amplitude von $N_{soll} = 100[1/s]$ bis $200[1/s]$ und einer Haltezeit von $T_h = 200[ms]$ bis $500[ms]$ vorgegeben.

Alle für die Identifikation relevanten Parameter und Ergebnisse sind in Tabelle 2 zusammengefasst. Abbildung 6 zeigt die erzielten Ergebnisse nach einer Identifikationszeit von 60 Sekunden. Während das Gradientenabstiegsverfahren noch keine zufriedenstellenden Resultate liefert, sehen die beiden Kennlinien beim Quasi-Online Newtonverfahren schon ganz gut aus. Der Eingangsbereich der Reibungs nichtlinearität wird durch das Anregungssignal nicht direkt angeregt. Die Reibung ist nicht direkt von der Drehzahl $N_{ist}[1/s]$ sondern von der Winkelgeschwindigkeit der Platte $\Omega[rad/s]$ abhängig. Der Rand des Eingangsbereiches wird selten angeregt und die exakte Identifikation dieser Bereiche dauert dadurch entsprechend lange.

Abbildung 7 zeigt den Verlauf des Ausgangsfehlers $e = \varphi - \hat{\varphi}$ während der Identifikation. In dieser 60-sekündigen Identifikation führt das rechenintensive Quasi-Online Newtonverfahren deutlich weniger Optimierungsschritte aus als das Gradientenabstiegsverfahren. Trotzdem erreicht dieses Verfahren eine viel schnellere Reduzierung des Ausgangsfehlers. Mit diesem Identifikationsbeispiel wird die Überlegenheit des Quasi-Online Newton-Optimierungsverfahrens verdeutlicht und rechtfertigt dessen Einsatz bei der Identifikation mit strukturierten rekurrenten Netzen.

Parameter	$\hat{\Psi}_1 = 1/J$	$\hat{\Psi}_2 = l_{SM}g$	Luftstrahlmoment	Reibmoment
Strecke	0.00101 [kgm ²]	0.0653 [Nm]	Abb. 6	Abb. 6
Initialisierung	0.00101 [kgm ²]	0 [Nm]	zufällig	zufällig
GRNN			$\sigma_{norm} = 0.45$ $r_{M_L'} = 10$	$\sigma_{norm} = 0.45$ $r_{M_R} = 20$
Gradientenabstiegsverfahren				
η	0	$2 \cdot 10^{-4}$	0.01	$2 \cdot 10^{-3}$
α	0	0.95	0.7	0.6
Quasi-Online Newtonverfahren				
$N = 500, \quad \mu = 3$				
Identifikationsergebnisse				
Gradient	/	0.065 [Nm]	Abb. 6(a)	Abb. 6(a)
Newton	/	0.0652 [Nm]	Abb. 6(b)	Abb. 6(b)
Beobachter	$\tilde{l} = [55 \quad 278]^T$			

Tabelle 2: Parameter und Ergebnisse der simulativen Identifikation des Plattenaufbaus

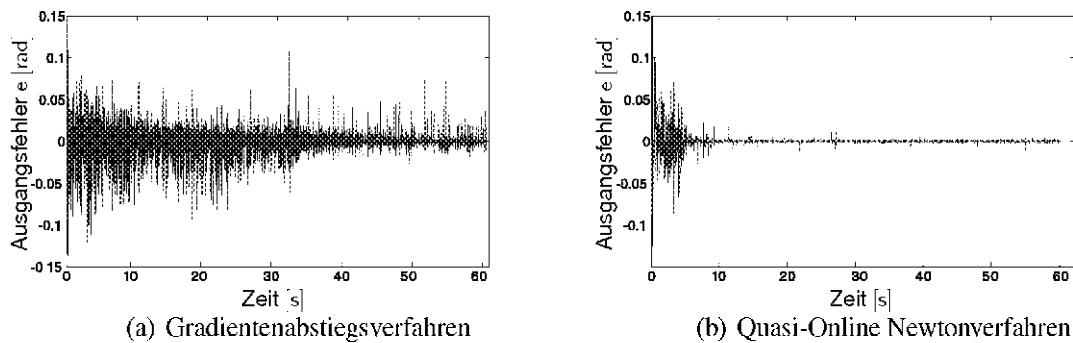


Bild 7: Verlauf des Ausgangsfehlers e während der Identifikation

Die **Identifikation an der realen Anlage** liefert ähnliche Ergebnisse, siehe Abbildung 8. Lediglich das Reibmoment zeigt für negative Winkelgeschwindigkeiten einen stärkeren Abfall. Die Identifikation der Anlage wurde wie oben beschrieben jedoch diesmal 300 Sekunden lang mit dem Quasi-Online Newton-Optimierungsverfahren durchgeführt.

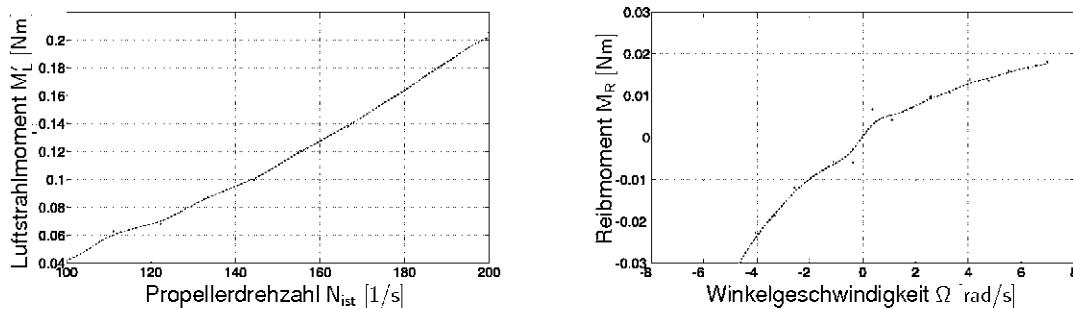


Bild 8: Luftstrahlmoment M_L' und Reibmoment M_R nach einer Identifikation von 300 Sekunden

7 Zusammenfassung und Ausblick

Mit strukturierten rekurrenten Netzen ist die Identifikation nichtlinearer dynamischer Systeme durchführbar, falls die Struktur des Systems ausreichend genau bekannt ist. Bei der Aufstellung des Netzes muss auf die Eindeutigkeit der Identifikationsstruktur geachtet werden. Eventuell müssen Kompromisse, wie bei dem hier gezeigten Laborbeispiel des Plattenaufbaus, eingegangen werden, um die Eindeutigkeit zu erreichen.

Mit dem Quasi-Online Newton-Optimierungsverfahren ist es möglich, die Methoden der nichtlinearen Optimierung 2. Ordnung auch auf strukturierte rekurrente Netze zu übertragen. Dadurch wird eine deutlich höhere Konvergenzgeschwindigkeit erreicht, verglichen mit dem herkömmlichen Gradientenabstieg mit Momentumterm. Die für feedforward Netze realisierbaren Verbesserungen um mehrere Dekaden konnte bei der Optimierung von strukturierten rekurrenten Netzen jedoch nicht erzielt werden. Der Vergleich mit einem Validierungsnetz sorgt für eine stabile Parameteradaptation. In Zukunft soll dieser Vergleich verwendet werden, um den fest eingestellten Wert des Dämpfungsparameters μ anzupassen, so dass kein einziger Parameter beim Optimierungsverfahren vom Benutzer eingestellt werden muss. Wie beim bekannten Levenberg-Marquardt-Algorithmus soll ein Skalierungsverfahren den Wert μ regulieren. Bei einer erreichten Verschlechterung wird der Dämpfungsparameter vergrößert, bei einer Verbesserung wird μ für den nächsten Optimierungsschritt verkleinert.

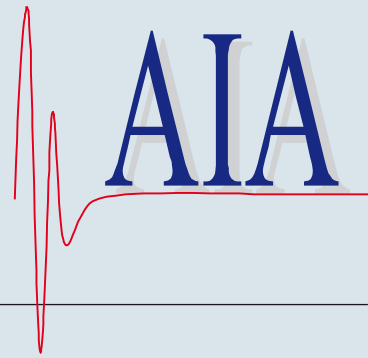
Literatur

- [1] Hintz, Chr.: *Identifikation nichtlinearer mechatronischer Systeme mit strukturierten rekurrenten Netzen*. TU München: Dissertation am Lst. f. El. Antriebssyst. 2003.
- [2] Specht, D. F.: *A general regression neural network*. In: *IEEE Transactions on Neural Networks* 2(6), S. 568–576. 1991.
- [3] Brychcy, T.: *Modellierung dynamischer Systeme mit vorstrukturierten neuronalen Netzen*. Berlin: Akademische Verlagsgesellschaft Aka GmbH. 2000.
- [4] Schröder, D.: *Intelligent Observer and Control Design for Nonlinear Systems*. Berlin: Springer. 2000.
- [5] Hagan, M. T.; Demuth, H. B.; Beale, M. H.: *Neural Network Design*. Boston: PWS Publishing Company. 1996.
- [6] Graf, M. W.: *Newton basierte Lernverfahren für Neuronale Netze zur Identifikation*. TU München: Diplomarbeit am Lst. f. El. Antriebssyst.. 2005.
- [7] Schröder, D.: *Elektrische Antriebe - Regelung von Antriebssystemen*. Berlin: Springer. 2001.
- [8] Makkar, C.; Dixon, W. E.; Sawyer, W. G.; Hu, G.: *A New Continuously Differentiable Friction Model for Control Systems Design*. In: *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, S. 600–605. 2005.
- [9] Rautscher, G.: *Quasi-Online Newtonverfahren zur Identifikation nichtlinearer dynamischer Systeme*. TU München: Diplomarbeit am Lst. f. El. Antriebssyst. 2006.

**Bereits veröffentlicht wurden in der Schriftenreihe des
Instituts für Angewandte Informatik / Automatisierungstechnik im
Universitätsverlag Karlsruhe:**

- Nr. 1: BECK, S.: Ein Konzept zur automatischen Lösung von Entscheidungsproblemen bei Unsicherheit mittels der Theorie der unscharfen Mengen und der Evidenztheorie, 2005
- Nr. 2: MARTIN, J.: Ein Beitrag zur Integration von Sensoren in eine anthropomorphe künstliche Hand mit flexiblen Fluidaktoren, 2004
- Nr. 3: TRACHEL, A.: Neue Verfahren zur Modellierung nichtlinearer thermodynamischer Prozesse in einem Druckbehälter mit siedendem Wasser-Dampf Gemisch bei negativen Drucktransienten, 2005
- Nr. 4: LOOSE, T.: Konzept für eine modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse, 2004
- Nr. 5: MATTHES, J.: Eine neue Methode zur Quellenlokalisierung auf der Basis räumlich verteilter, punktwiser Konzentrationsmessungen, 2004
- Nr. 6: MIKUT, R.; REISCHL, M.: Proceedings - 14. Workshop Fuzzy-Systeme und Computational Intelligence : Dortmund, 10. - 12. November 2004, 2004
- Nr. 7: ZIPSER, S.: Beitrag zur modellbasierten Regelung von Verbrennungsprozessen, 2004
- Nr. 8: STADLER, A.: Ein Beitrag zur Ableitung regelbasierter Modelle aus Zeitreihen, 2005
- Nr. 9: MIKUT, R.; REISCHL, M.: Proceedings - 15. Workshop Computational Intelligence: Dortmund, 16. - 18. November 2005, 2005
- Nr. 10: BÄR, M.: μ FEMOS - Mikro-Fertigungstechniken für hybride mikrooptische Sensoren, 2005
- Nr. 11: SCHAUDEL, F.: Entropie- und Störungssensitivität als neues Kriterium zum Vergleich verschiedener Entscheidungskalküle, 2006
- Nr. 12: SCHABLOWSKI-TRAUTMANN, M.: Konzept zur Analyse der Lokomotion auf dem Laufband bei inkompletter Querschnittlähmung mit Verfahren der nicht-linearen Dynamik, 2006
- Nr. 13: REISCHL, M.: Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen, 2006

Die Schriften sind als PDF frei verfügbar, eine Nachbestellung der Printversion ist möglich.
Nähere Informationen unter www.uvka.de.



Dieser Tagungsband enthält die Beiträge des 16. Workshops „Computational Intelligence“ des Fachausschusses 5.14 der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und der Fachgruppe „Fuzzy-Systeme und Soft-Computing“ der Gesellschaft für Informatik (GI), der vom 29. November bis 1. Dezember 2006 im Haus Bommerholz bei Dortmund stattfindet.

Der GMA-Fachausschuss 5.14 „Computational Intelligence“ entstand 2005 aus den bisherigen Fachausschüssen „Neuronale Netze und Evolutionäre Algorithmen“ (FA 5.21) sowie „Fuzzy Control“ (FA 5.22). Der Workshop steht in der Tradition der bisherigen Fuzzy-Workshops, hat aber seinen Fokus in den letzten Jahren schrittweise erweitert. Die Schwerpunkte sind Methoden, Anwendungen und Tools für

- Fuzzy-Systeme,
- Künstliche Neuronale Netze,
- Evolutionäre Algorithmen und
- Data-Mining-Verfahren

sowie der Methodenvergleich anhand von industriellen und Benchmark-Problemen.

Die Ergebnisse werden von Teilnehmern aus Hochschulen, Forschungseinrichtungen und der Industrie in einer offenen Atmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

ISSN: 1614-5267

ISBN-13: 978-3-86644-057-9

ISBN-10: 3-86644-057-X